Programmer's Guide

HP 8752C Network Analyzer

HP Part No. 08752-90137 Supersedes August 1994
Printed in USA July 1997

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.

1212 Valley House Drive, Rohnert Park, CA 94928-4999, USA

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. See the HP 8752C User's Guide for the "Hewlett-Packard Sales and Service Offices" table.

Safety Notes

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

Caution	Caution denotes a hazard. It calls attention to a procedure that, if not correctly performed or adhered to, would result in damage to or destruction of the instrument. Do not proceed beyond a caution note until the indicated conditions are fully understood and met.
Warning	**Warning denotes a hazard. It calls attention to a procedure which, if not correctly performed or adhered to, could result in injury or loss of life. Do not proceed beyond a warning note until the indicated conditions are fully understood and met.**

General Safety Considerations

Warning	**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.**
Warning	**No operator serviceable parts inside. Refer servicing to qualified personnel. To prevent electrical shock, do not remove covers.**
Caution	Before switching on this instrument, make sure that the line voltage selector switch is set to the voltage of the power supply and the correct fuse is installed.
Warning	**These servicing instructions are for use by qualified personnel only. To avoid electrical shock, do not perform any servicing unless you are qualified to do so.**
Warning	**The opening of covers or removal of parts is likely to expose dangerous voltages. Disconnect the instrument from all voltage sources while it is being opened.**
Warning	**The power cord is connected to internal capacitors that may remain live for 10 seconds after disconnecting the plug from its power supply.**
Warning	**For continued protection against fire hazard replace line fuse only with same type and rating (F 5A/250V). The use of other fuses or material is prohibited.**

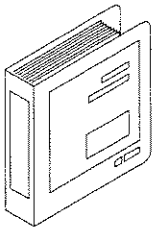
How to Use This Guide

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.

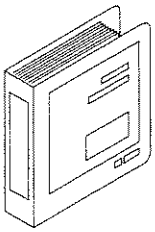
HP 8752C Network Analyzer Documentation Set

The **Installation and Quick Start Guide** familiarizes you with the HP 8752C 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 HP 8752C.



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 selected 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 8752C network analyzer.

Contents

1. **HP-IB Programming and Command Reference**

Where to Look for More Information . . . 1-2
Analyzer Command Syntax . . . 1-3
Code Naming Convention . . . 1-3
Valid Characters . . . 1-4
Units . . . 1-4
Command Formats . . . 1-4
General Structure: . . . 1-4
Syntax Types . . . 1-5
HP-IB Operation . . . 1-6
Device Types . . . 1-6
Talker . . . 1-6
Listener . . . 1-6
Controller . . . 1-6
HP-IB Bus Structure . . . 1-7
Data Bus . . . 1-7
Handshake Lines . . . 1-7
Control Lines . . . 1-7
HP-IB Requirements . . . 1-8
HP-IB Operational Capabilities . . . 1-9
HP-IB Status Indicators . . . 1-10
Bus Device Modes . . . 1-10
System-Controller Mode . . . 1-11
Talker/Listener Mode . . . 1-11
Pass-Control Mode . . . 1-11
Analyzer Bus Modes . . . 1-11
Setting HP-IB Addresses . . . 1-12
Response to HP-IB Meta-Messages (IEEE-488 Universal Commands) . . . 1-12
Abort . . . 1-12
Device Clear . . . 1-12
Local . . . 1-12
Local Lockout . . . 1-13
Parallel Poll . . . 1-13
Pass Control . . . 1-13
Remote . . . 1-13
Serial Poll . . . 1-13
Trigger . . . 1-13
Analyzer Operation . . . 1-14
Operation Complete . . . 1-14
Reading Analyzer Data . . . 1-15
Output Queue . . . 1-15
Command Query . . . 1-15
Identification . . . 1-15
Output Syntax . . . 1-15
Marker data . . . 1-16
Array-Data Formats . . . 1-17

Trace-Data Transfers . . . 1-19
Stimulus-Related Values . . . 1-19
Data-Processing Chain . . . 1-21
Data Arrays . . . 1-21
Fast Data Transfer Commands . . . 1-22
Data Levels . . . 1-22
Learn String and Calibration-Kit String . . . 1-23
Error Reporting . . . 1-24
Status Reporting . . . 1-24
The Status Byte . . . 1-26
The Event-Status Register and Event-Status-Register B . . . 1-26
Error Output . . . 1-27
Calibration . . . 1-28
Disk File Names . . . 1-29
Using Key Codes . . . 1-30
Key Select Codes Arranged by Front-Panel Hardkey . . . 1-31
HP-IB Only Commands . . . 1-49
Alphabetical Mnemonic Listing . . . 1-55

2. HP BASIC Programming Examples

Introduction . . . 2-1
Required Equipment . . . 2-1
Optional Equipment . . . 2-2
System Setup and HP-IB Verification . . . 2-2
HP 8752C Network Analyzer Instrument Control Using BASIC . . . 2-4
Command Structure in BASIC . . . 2-4
Command Query . . . 2-5
Running the Program . . . 2-6
Operation Complete . . . 2-7
Running the Program . . . 2-7
Preparing for Remote (HP-IB) Control . . . 2-7
I/O Paths . . . 2-8
Measurement Process . . . 2-10
Step 1. Setting Up the Instrument . . . 2-10
Step 2. Calibrating the Test Setup . . . 2-10
Step 3. Connecting the Device under Test . . . 2-10
Step 4. Taking the Measurement Data . . . 2-11
Step 5. Post-Processing the Measurement Data . . . 2-11
Step 6. Transferring the Measurement Data . . . 2-11
BASIC Programming Examples . . . 2-12
Program Information . . . 2-13
Analyzer Features Helpful in Developing Programming Routines . . . 2-13
Analyzer-Debug Mode . . . 2-13
User-Controllable Sweep . . . 2-13
Example 1: Measurement Setup . . . 2-14
Example 1A: Setting Parameters . . . 2-14
Running the Program . . . 2-15
Example 1B: Verifying Parameters . . . 2-16
Running the Program . . . 2-17
Example 2: Measurement Calibration . . . 2-18
Calibration kits . . . 2-18
Example 2A: Response Measurement Calibration . . . 2-19
Running the Program . . . 2-20
Example 2B: Reflection 1-Port Measurement Calibration . . . 2-21
Running the Program . . . 2-22

Example 3: Measurement Data Transfer . . . 2-23
Trace-Data Formats and Transfers . . . 2-23
Example 3A: Data Transfer Using Markers . . . 2-24
Running the Program . . . 2-25
Example 3B: Data Transfer Using FORM 4 (ASCII Transfer) . . . 2-26
Running the Program . . . 2-28
Example 3C: Data Transfer Using Floating-Point Numbers . . . 2-29
Running the Program . . . 2-31
Example 3D: Data Transfer Using Frequency-Array Information . . . 2-32
Running the Program . . . 2-34
Example 3E: Data Transfer Using FORM 1, Internal-Binary Format . . . 2-35
Running the Program . . . 2-36
Example 4: Measurement Process Synchronization . . . 2-37
Status Reporting . . . 2-37
Example 4A: Using the Error Queue . . . 2-38
Running the Program . . . 2-39
Example 4B: Generating Interrupts . . . 2-41
Running the Program . . . 2-43
Example 5: Network Analyzer System Setups . . . 2-44
Saving and Recalling Instrument States . . . 2-44
Example 5A: Using the Learn String . . . 2-44
Running the Program . . . 2-45
Example 5B: Reading Calibration Data . . . 2-46
Running the Program . . . 2-48
Example 5C: Saving and Restoring the Analyzer Instrument State . . . 2-49
Running the Program . . . 2-51
Example 6: Limit-Line Testing . . . 2-52
Using List-Frequency Mode . . . 2-52
Example 6A: Setting Up a List-Frequency Sweep . . . 2-52
Running the Program . . . 2-54
Example 6B: Selecting a Single Segment from a Table of Segments . . . 2-55
Running the Program . . . 2-57
Using Limit Lines to Perform PASS/FAIL Tests . . . 2-58
Example 6C: Setting Up Limit Lines . . . 2-58
Running the Program . . . 2-60
Example 6D: Performing PASS/FAIL Tests While Tuning . . . 2-61
Running the Program . . . 2-63
Example 7: Report Generation . . . 2-64
Example 7A1: Operation Using Talker/Listener Mode . . . 2-64
Running the Program . . . 2-65
Example 7A2: Controlling Peripherals Using Pass-Control Mode . . . 2-66
Running the Program . . . 2-68
Example 7B: Plotting to a File and Transferring File Data to a Plotter . . . 2-69
Running the Program . . . 2-71
Utilizing PC-Graphics Applications Using the Plot File . . . 2-71
Example 7C: Reading ASCII Disk Files to the System Controller Disk File . . . 2-72
Running the Program . . . 2-76
Limit Line and Data Point Special Functions . . . 2-77
Overview . . . 2-78
Example Display of Limit Lines . . . 2-80
Limit Segments . . . 2-81
Output Results . . . 2-82
Constants Used Throughout This Document . . . 2-83
Output Limit Test Pass/Fail Status Per Limit Segment . . . 2-84
Output Pass/Fail Status for All Segments . . . 2-85

Example Program of OUTPSEGAF Using BASIC 2-85
Output Minimum and Maximum Point Per Limit Segment 2-87
Output Minimum and Maximum Point For All Segments 2-88
Example Program of OUTPSEGAM Using BASIC 2-89
Output Data Per Point 2-90
Output Data Per Range of Points 2-91
Output Limit Pass/Fail by Channel 2-92

Index

Figures

1-1. HP-IB Bus Structure 1-7
1-2. Analyzer Single Bus Concept 1-10
1-3. FORM 4 (ASCII) Data-Transfer Character String 1-16
1-4. The Data-Processing Chain 1-21
1-5. Status Reporting Structure 1-24
1-6. Key Codes 1-30
2-1. The HP 8752C Network Analyzer System with Controller 2-2
2-2. Status Reporting Structure 2-37
2-3. Limit Segments Versus Limit Lines 2-80

Tables

1-1. Code Naming Convention 1-3
1-2. OPC-compatible Commands 1-14
1-3. Units as a Function of Display Format 1-17
1-4. HP 8752C Network Analyzer Array-Data Formats 1-18
1-5. Status Bit Definitions 1-25
1-6. Calibration Arrays 1-28
1-7. Disk File Names 1-29
1-8. Key Select Codes 1-32
1-9. HP-IB Only Commands 1-49
2-1. Additional BASIC 6.2 Programming Information 2-1
2-2. Additional HP-IB Information 2-1
2-3. HP 8752C Network Analyzer Array-Data Formats 2-26
2-4. Limit Line and Data Point Special Functions Commands 2-78
2-5. Limit Segment Table for Figure 2-4 2-81
2-6. Example Output: OUTPSEGAM (min/max of all segments) 2-82
2-7. Pass/Fail/No_Limit Status Constants 2-83
2-8. Min/Max Test Constants 2-83
2-9. Example Output: OUTPSEGAF (pass/fail for all segments) 2-85
2-10. Example Output: OUTPSEGM (min/max per segment) 2-87
2-11. Example Output: OUTPSEGAM (min/max for all segments) 2-88
2-12. Example Output: OUTPDATP (data per point) 2-90
2-13. Example Output: OUTPDATPR (data per range of points) 2-91

Index

Special characters

$, 1-31

A

A/B, 1-55
AB, 1-55
abort message (IFC), 1-12
abort sequence, 2-7
additional information, 2-1
  BASIC 6.2, 2-1
ADDRCONT[D], 1-55
ADDRDISC[D], 1-55
address
  controller, 1-55
  disk drive, 1-55
  peripheral, 1-55
  plotter, 1-55
  power meter, 1-55
  printer, 1-55
address capability, 1-8
addresses for HP-IB, 1-12
ADDRPERI[D], 1-55
ADDRPLOT[D], 1-55
ADDRPOWM[D], 1-55
ADDRPRIN[D], 1-55
adjust brightness, 1-56
adjust color, 1-58
adjust tint, 1-78
AH1 (full-acceptor handshake), 1-9
ALTAB, 1-55
alternate inputs, 1-55
ANAB, 1-55
ANAI, 1-55
analog bus, 1-55
analog input, 1-55
analyzer array-data formats, 1-18
analyzer bus mode, 1-11
analyzer command syntax, 1-3
analyzer control of peripherals, 1-11
analyzer data reading, 1-15
analyzer-debug mode, 2-13
analyzer features helpful in developing programs, 2-13
analyzer identification, 1-15
analyzer operating modes, 2-3
  pass-control mode, 2-3, 2-66
  system-control mode, 2-3
  talker/listener, 2-3, 2-64
analyzer operation, 1-14
analyzer single bus concept, 1-10
analyzer status reporting structure, 1-24
appendage in syntax, 1-4
AR, 1-55
array-data formats, 1-17, 2-26
  FORM 1, 2-26
  FORM 2, 2-26
  FORM 3, 2-26
  FORM 4, 2-24, 2-26
  FORM 5, 2-26
arrays of data, 1-21
arrays related to frequency, 1-19
ASCII
  save format, 1-74
ASCII disk files, 2-72
  reading, 2-72
ASEG, 1-55
assert sequence, 1-55
ASSS, 1-55
ATN (attention) control line, 1-7
attention (ATN) control line, 1-7
AUTO, 1-55
auto feed
  plotter, 1-71
  printer, 1-72
auto scale, 1-55
averaging, 1-55
  restart, 1-55
averaging factor, 1-55
AVERFACT[D], 1-55
AVERO, 1-55
AVERREST, 1-55
AVG HP-IB commands, 1-32

B

BACI[D], 1-55
background intensity, 1-55
BANDPASS, 1-55
basic talker (T6), 1-9
beep
  emit, 1-60
BEEPDONE, 1-55
beeper on done, 1-55

beeper on warning, 1-55
BEEPFAIL, 1-55
BEEPWARN, 1-55
begin cal sequence, 1-56
bi-directional lines, 1-7
binary
  save format, 1-74
BR, 1-55
bus device modes, 1-10
bus structure, 1-6, 1-7

C

C10 (pass control capabilities), 1-9
C1,C2,C3 (system controller capabilities), 1-9
C1[D], 1-56
C2[D], 1-56
C3[D], 1-56
CAL1, 1-56
CALFCALF[D], 1-56
CALFFREQ[D], 1-56
CALFSENA, 1-56
CALFSENB, 1-56
CAL HP-IB commands, 1-32
calibrating the test setup, 2-10
calibration arrays, 1-28
calibration coefficients, 1-21
calibration command sequence, 1-28
calibration data, 2-46
  inputting, 2-46
  outputting, 2-46
  reading, 2-46
calibration example program, 2-18
calibration kit, 2-2
calibration kit HP-IB commands, 1-33
calibration kits, 1-56, 2-18
calibration kit string and learn string, 1-23
calibration type off, 1-56
CALIRAI, 1-56
CALIRESP, 1-56
CALIS111, 1-56
CALK35MD, 1-56
CALK35MM, 1-56
CALK7MM, 1-56
cal kit done, 1-63
CALKN50, 1-56
CALKN75, 1-56
CALKUSED, 1-56
CALN, 1-56
cal sensor table
  edit, 1-56
cal sequence
  begin, 1-56
  resume, 1-73
CBRI[D], 1-56
CENT[D], 1-56
center, 1-56
chain for data processing, 1-21
CHAN1, 1-56
CHAN2, 1-56
CHANNEL HP-IB commands, 1-34
channels
  coupled, 1-58
characters that are valid, 1-4
CHOPAB, 1-56
citifile
  save format, 1-74
CLAD, 1-56
CLASS11A, 1-57
CLASS11B, 1-57
CLASS11C, 1-57
class done, 1-56
CLEABIT[D], 1-57
CLEA<I>, 1-57
CLEAL, 1-57
CLEARALL, 1-57
clear device, 1-12
CLEAREG<I>, 1-57
clearing any messages waiting to be output, 2-7
clearing syntax errors, 2-7
clearing the input-command buffer, 2-7
clear list, 1-57
clear register, 1-57
clear sequence, 1-57, 2-7
CLEASEQ<I>, 1-57
CLEL, 1-57
CLES, 1-54, 1-57
CLS, 1-57
COAX, 1-57
CO[D], 1-56
code naming conventions, 1-3
code syntax structure, 1-4
COLOCH1D, 1-58
COLOCH1M, 1-58
COLOCH2D, 1-58
COLOCH2M, 1-58
COLOGRAT, 1-58
color
  data channel 1, 1-70
  data channel 2, 1-70
  graticule, 1-70
  memory channel 1, 1-70
  memory channel 2, 1-70
  text, 1-70
  warning, 1-70
COLOR[D], 1-58
colors, 1-70
COLOTEXT, 1-58
COLOWARN, 1-58

? command, 1-15
command formats, 1-4
command query, 1-15
commands
  HP-IB, 1-1
command structure, 2-4
command structure elements, 2-4
  appendage, 2-4
  BASIC command statement, 2-4
  data, 2-4
  terminators, 2-4
  unit, 2-4
command syntax, 1-3
command syntax structure, 1-4
compatible peripherals, 2-2
complete operation, 1-14
complete service request capabilities (SR1), 1-9
computer controllers, 1-6
connecting the device under test, 2-10
connecting the test system, 2-2
CONS, 1-58
CONSTANTS, 2-83
CONT, 1-58
continue sequence, 1-58
controlled sweep, 2-13
controller
  address, 1-55
controller interface function, 1-6
control lines, 1-7
CONV1DS, 1-58
conventions for code naming, 1-3
CONVOFF, 1-58
CONVREF, 1-58
CONVYTRA, 1-58
CONVZTRA, 1-58
copy display, 1-69, 1-71, 1-72
COPYFRFT, 1-58
COPYFRRT, 1-58
COPY HP-IB commands, 1-35
CORI, 1-58
CORR, 1-58
correction, 1-58
  interpolative, 1-58
correction of errors example program, 2-18
COUC, 1-58
COUP, 1-58
coupled channels, 1-58
CRT focus, 1-61
CRT intensity, 1-63
CRT title, 1-79
CW freq, 1-58
CWFREQ[D], 1-58
CW time, 1-58
CWTIME, 1-58

D

[D], 1-31
D1DIVD2, 1-58
data
  include with disk files, 1-61
data-array formats, 1-17
data arrays, 1-21
data bus, 1-7
data channel 1
  color, 1-70
data channel 2
  color, 1-70
data for markers, 1-16
data formats and transfers, 2-23
data levels, 1-22
data only
  include with disk files, 1-61
data-processing chain, 1-21
data rate, 1-8
data reading, 1-15
data taking, 2-11
data transfer, 1-7, 2-11, 2-23
  to a plotter, 2-69
  using floating-point numbers, 2-29
  using FORM 1, 2-35
  using FORM 4, 2-26
  using frequency-array information, 2-32
  using markers, 2-24
data-transfer character definitions, 1-16
Data Transfer Commands
  Fast, 1-22
data transfer for traces, 1-19
data units, 1-4
DATI, 1-58
DC1 (complete device clear), 1-9
DEBU, 1-58
debug, 1-58
debug mode, 2-5, 2-13
decrement loop counter, 1-58
DECRLOOC, 1-58
default calibration kits, 1-56
default colors, 1-58
DEFC, 1-58
definitions of status bit, 1-24
DEFLPRINT, 1-59
DEFLTCPIO, 1-49, 1-59
DEFS[D], 1-59
DELA, 1-59
delay, 1-59, 1-60
  set to mkr, 1-66
delete segment, 1-74
DEL&<I>, 1-59
DELO, 1-59
DELRFIXM, 1-59

delta limits, 1-64
delta reference, 1-59
DEMOAMPL, 1-59
demodulation off, 1-59
DEMOOFF, 1-59
DEMOPHAS, 1-59
DeskJet, 1-72
developing program features, 2-13
device clear, 1-12
device clear (DC1), 1-9
device connection, 2-10
device trigger, 1-13
device types for HP-IB, 1-6
DFLT, 1-59
directory size
  LIF, 1-60
DIRS[D], 1-60
disabling the front panel, 1-13
DISCUNIT[D], 1-60
DISCVOLU[D], 1-60
disk
  load file, 1-65
disk drive
  address, 1-55
disk drive unit, 1-60
disk drive volume, 1-60
disk file names, 1-29
disk files HP-IB commands, 1-43
disk format, 1-62
DISM, 1-60
DISPDATA, 1-60
DISPDATM, 1-60
DISPDDM, 1-60
DISPDMM, 1-60
display A/B, 1-55
display A/R, 1-55
display B/R, 1-55
display data, 1-60
display data — mem, 1-60
display data & mem, 1-60
display data/mem, 1-60
display data to mem, 1-58
display format units, 1-17
DISPLAY HP-IB commands, 1-36
display memory, 1-60
DISPMEMO, 1-60
DIVI, 1-60
does not respond to parallel poll (PPO), 1-9
done
  with class, 1-60
  with isolation, 1-63
DONE, 1-60
done modify sequence, 1-60
DONM, 1-60
DOSEQ<I>, 1-60
do sequence, 1-60
DOS format, 1-62
DOWN, 1-60
DT1 (responds to a group execute trigger), 1-9
DUAC, 1-60
dual channels, 1-60
duplicate sequence, 1-60
DUPLSEQ<X>SEQ<Y>, 1-60

E

E2 (tri-state drivers), 1-9
edit cal sensor table, 1-56
EDITDONE, 1-60
edit limit table, 1-60
EDITLIML, 1-60
EDITLIST, 1-60
edit power loss range, 1-71
edit power loss table, 1-71
edit segment, 1-75
ELED[D], 1-60
EMIB, 1-60
emit beep, 1-60
end or identify, 1-5
end or identify (EOI) control line, 1-7
ENTO, 1-61
ENTRY HP-IB commands, 1-37
entry off, 1-61
EOI, 1-5
EOI (end or identify) control line, 1-7
Epson-P2, 1-72
equipment
  optional, 2-2
  required, 2-1
error-corrected data, 1-21
error-correction example program, 2-18
error correction HP-IB commands, 1-32
error output, 1-27
error queue, 2-38
error reporting, 1-24
ESB?, 1-54, 1-61
ESE[D], 1-54, 1-61
ESNB[D], 1-54, 1-61
ESR?, 1-54, 1-61
event-status register, 1-24, 1-26
event-status-register B, 2-61
event-status registers, 2-37
example
  measurement calibration, 2-18
EXTD, 1-61
extended listener capabilities (LEO), 1-9
external trigger, 1-61
EXTMDATA, 1-61
EXTMDATO, 1-61
EXTMFORM, 1-61

EXTMGRAP, 1-61
EXTMRAW, 1-61
EXTTHIGH, 1-49, 1-61
EXTTLOW, 1-49, 1-61
EXTTOFF, 1-61
EXTTON, 1-61
EXTTPOIN, 1-61

F

Fast Data Transfer Commands, 1-22
features helpful in developing programming routines, 2-13
file names
  disk, 1-29
file titles
  recall, 1-73
firmware revision identification, 1-15
FIXE, 1-61
fixed load, 1-61
fixed marker, 1-59
flat line type, 1-64
FOCU[D], 1-61
FORM1, 1-54, 1-61
FORM1 format, 1-18
FORM2, 1-54, 1-61
FORM2 format, 1-18
FORM3, 1-54, 1-61
FORM3 format, 1-18
FORM4, 1-54, 1-61
form 4 data-transfer character string, 1-16
FORM4 format, 1-18
FORM5, 1-54, 1-61
FORM5 format, 1-18
format
  disk, 1-62
format display units, 1-17
FORMATDOS, 1-62
FORMAT HP-IB commands, 1-37
FORMATLIF, 1-62
formats and transfers of trace-data, 2-23
formats for array-data, 1-17
formats for commands, 1-4
formatted data, 1-21
  include with disk files, 1-61
form feed
  plotter, 1-71
  printer, 1-72
FREO, 1-62
frequency calculation equation, 2-26
frequency notation, 1-62
frequency-related arrays, 1-19
full-acceptor handshake (AH1), 1-9
full-source handshake (SH1), 1-9
FULP, 1-62

G

GATE, 1-62
GATECENT[D], 1-62
gate center time, 1-62
gate on/off, 1-62
gate shape, 1-62
  maximum, 1-62
  minimum, 1-62
  normal, 1-62
  wide, 1-62
GATESPAN[D], 1-62
gate span time, 1-62
GATESTAR[D], 1-62
gate start time, 1-62
GATESTOP[D], 1-62
gate stop time, 1-62
GATSMAXI, 1-62
GATSMINI, 1-62
GATSNORM, 1-62
GATSWIDE, 1-62
general structure of syntax, 1-4
GOSUB, 1-62
gosub sequence, 1-62
graticule
  color, 1-70
group execute trigger response (DT1), 1-9
guidelines for code naming, 1-3

H

halting all modes and functions, 1-12
handshake lines, 1-7
helpful features for developing programs, 2-13
HOLD, 1-62
HP 9000 Series 300 computer, 2-1
HP-IB
  address capability, 1-8
  addresses, 1-12
  bus structure, 1-6, 1-7
  command formats, 1-4
  data rate, 1-8
  device types, 1-6
  message transfer scheme, 1-8
  meta-messages, 1-12
  multiple-controller capability, 1-8
  operation, 1-6
  operational capabilities, 1-9
  requirements, 1-8
  status indicators, 1-10
HP-IB commands, 1-1
HP-IB interconnect cables, 2-1
HP-IB only commands, 1-49

I

<I>, 1-31
identification
  of analyzer, 1-15
  of firmware revision, 1-15
IDN?, 1-15, 1-49, 1-62
IEEE-488 universal commands, 1-12
IEEE standard codes, formats, protocols information, 1-2
IEEE standard digital interface information, 1-2
IF bandwidth, 1-62
IFBIHIGH, 1-62
IFBILOW, 1-62
IFBW[D], 1-62
IFC (abort message), 1-12
IFC (interface clear) control line, 1-7
IFLCEQZESEQ<I>, 1-62
IFLCNEZESEQ<I>, 1-62
IFLTFAILSEQ<I>, 1-62
IFLTPASSSEQ<I>, 1-62
IMAG, 1-62
imaginary, 1-62
increment loop counter, 1-63
INCRLOOC, 1-63
information on programs, 2-13
INID, 1-63
INIE, 1-63
INPUCALC<I>, 1-50
INPUCALC<I>[D], 1-63
INPUCALK[D], 1-50, 1-63
INPUDATA[D], 1-50, 1-63
INPUFORM[D], 1-50, 1-63
INPULEAS[D], 1-50, 1-63
INPURAW<I>, 1-63
INPURAW<I>[D], 1-50
input/output path, 2-8
instrument setup, 2-10
instrument states, 2-44
  recalling, 2-44, 2-49
  saving, 2-44, 2-49
instrument state summary, 1-23
INTE[D], 1-63
intensity
  background, 1-55
interface addresses, 1-12
interface clear (IFC) control line, 1-7
interface functions
  controller, 1-6
  listener, 1-6
  talker, 1-6
interpolative correction, 1-58
interrupts, generating, 2-41
INTM, 1-63
ISOD, 1-63

K

key codes, 1-30
KEY[D], 1-49, 1-63
key select codes, 1-31
KITD, 1-63
kit done, 1-63
kits of calibration standards, 2-18
KOR?, 1-49

L

LABEFWDT[$], 1-64
label cal kit, 1-64
label class, 1-64
label standard, 1-64
LABERESI[$], 1-64
LABERESP[$], 1-64
LABES11A[$], 1-64
LABES11B[$], 1-64
LABES11C[$], 1-64
LABK[$], 1-64
LABS[$], 1-64
LaserJet, 1-72
LE0 (no extended listener capabilities), 1-9
learn string and calibration kit string, 1-23
LEFL, 1-64
LEFU, 1-64
levels of data, 1-22
LIF
  directory size, 1-60
LIF format, 1-62
LIMD[D], 1-64
LIMIAMPO[D], 1-64
LIMILINE, 1-64
LIMIMAOF[D], 1-64
LIMISTIO[D], 1-64
LIMITEST, 1-64
limit line, 1-64
limit line amplitude offset, 1-64
limit line and data point special functions, 2-77
limit lines, 2-58
  setting up, 2-58
limit line stimulus offset, 1-64
limit-line testing, 2-52
  list-frequency table, creating, 2-52
  list-frequency table, selecting a single segment, 2-55
  performing PASS/FAIL tests, 2-58
  using list-frequency mode, 2-52
limit table
  edit, 1-60
limit test, 1-64
limit test beeper, 1-55
limit test fail, 1-62

limit testing HP-IB commands, 1-46
limit test pass, 1-62
limit-test table, 2-58
  creating, 2-58
  transmitting, 2-58
LIML[D], 1-64
LIMM[D], 1-64
LIMS[D], 1-64
LIMTFL, 1-64
LIMTSL, 1-64
LIMTSP, 1-64
LIMU[D], 1-64
linear sweep, 1-64
line feeds, 1-5
lines for control, 1-7
lines for handshaking, 1-7
line type
  data, 1-64
  memory, 1-64
LINFREQ, 1-64
LINM, 1-64
lin mag, 1-64
LINTDATA[D], 1-64
LINTMEMO[D], 1-64
LISFREQ, 1-64
list
  clear, 1-57
listener interface function, 1-6
listen mode (L), 1-10
list-frequency mode, 2-52
list sweep, 1-64
list values, 1-64
  print, 1-72
LISV, 1-64
L (listen mode), 1-10
LOAD<I>, 1-65
LOADSEQ<I>, 1-65
local command (GTL), 1-12
LOCAL HP-IB commands, 1-38
local lockout, 2-5
local lockout command (LLO), 1-13
local mode, 2-5
LOGFREQ, 1-65
LOGM, 1-65
log mag, 1-65
log sweep, 1-65
LOOC[D], 1-65
loop counter
  decrement, 1-58
  increment, 1-63
loop counter value, 1-65
lower limit
  segment, 1-64
low pass frequency, 1-76
low pass impulse, 1-65
low pass step, 1-65
LOWPIMPU, 1-65
LOWPSTEP, 1-65

M

MANTRIG, 1-65
MARKBUCK[D], 1-49
MARKCENT, 1-66
MARKCONT, 1-66
MARKCOUP, 1-66
MARKCW, 1-66
MARKDELA, 1-66
MARKDISC, 1-66
marker bandwidth search, 1-80
marker data, 1-16
MARKER FCTN HP-IB commands, 1-42
MARKER HP-IB commands, 1-41
marker parameters
  print, 1-72
marker positioning, 2-24
  by data point location, 2-24
  by frequency location, 2-24
  by trace-data value, 2-24
marker range, 1-66
markers
  continuous, 1-66
  discrete, 1-66
  displayed, 1-60
markers coupled, 1-66
marker search
  left, 1-75
  maximum, 1-75
  minimum, 1-75
  off, 1-75
  right, 1-75
  target, 1-75
  tracking, 1-79
markers off, 1-66
marker statistics, 1-67
markers uncoupled, 1-66
marker to CW frequency, 1-66
marker to limit offset, 1-64
marker to middle
  segment, 1-66
marker to stimulus
  segment, 1-66
marker width, 1-80
marker zero, 1-66
MARKFAUV[D], 1-66
MARKFSTI[D], 1-66
MARKFVAL[D], 1-66
MARK&<I>[D], 1-66
MARKMIDD, 1-66
MARKMINI, 1-66
MARKOFF, 1-66

MARKREF, 1-66
MARKSPAN, 1-66
MARKSTAR, 1-66
MARKSTIM, 1-66
MARKSTOP, 1-66
MARKUNCO, 1-66
MARKZERO, 1-66
MAXF[D], 1-66
MEASA, 1-66
MEASB, 1-67
MEAS HP-IB commands, 1-38
MEASR, 1-67
MEASTAT, 1-67
measurement calibration, 1-28
measurement calibration example program, 2-18
measurement data post-processing, 2-11
measurement data taking, 2-11
measurement parameters
  required order, 2-14
  setting, 2-14
  verifying, 2-16
measurement process, 2-10
measurement restart, 1-74
measurement setup, 2-14
measurement specifications, 2-28
  group delay, 2-28
  magnitude, 2-28
  phase, 2-28
memory channel 1
  color, 1-70
memory channel 2
  color, 1-70
memory requirements, 2-1
MENU, 1-67
MENUAVG, 1-50, 1-67
MENUCAL, 1-50, 1-67
MENUCOPY, 1-50, 1-67
MENUDISP, 1-50, 1-67
MENUFORM, 1-50, 1-67
MENUMARK, 1-50, 1-67
MENUMEAS, 1-50, 1-67
MENUMRKF, 1-50, 1-67
MENU<ON|OFF>, 1-50
MENURECA, 1-50, 1-67
MENUSAVE, 1-50, 1-67
MENUSCAL, 1-50, 1-67
MENUSEQU, 1-50, 1-67
MENUSTIM, 1-50, 1-67
MENUSYST, 1-50, 1-67
message transfer scheme, 1-8
meta-messages, 1-12
methods of HP-IB operation, 1-6
middle value
  segment, 1-64
MINF[D], 1-67
MINMAX<ON|OFF>, 1-53, 1-67, 2-78
min/max recording , 1-67
modes
  analyzer bus, 1-11
  debug, 2-13
  pass-control, 1-11
  system-controller, 1-10
  talker/listener, 1-11
modes for bus device, 1-10
MODI1, 1-67
modify cal kit, 1-67
modify colors, 1-58
modify sequence, 1-67
multiple-controller capability, 1-8

N

naming conventions, 1-3
NEWSEQ<I>, 1-67
new sequence, 1-67
NEXP, 1-67
next page, 1-67
no extended talker capabilities (TEO), 1-9
NOOP, 1-67
number of HP-IB devices allowed, 1-6
number of listeners allowed, 1-6
NUMG[D], 1-67

O

OFSD[D], 1-67
OFSL[D], 1-67
OFSZ[D], 1-68
OPC, 1-49, 1-68
OPC-compatible commands, 1-14
open capacitance values, 1-56
OPEP, 1-68
operating parameters, 1-68
operational capabilities for HP-IB, 1-9
operation complete, 1-14
operation complete commands, 2-7
operation of analyzer, 1-14
operation of HP-IB, 1-6
OUTPACTI, 1-51
OUTPAMAX, 1-52, 1-68, 2-78
OUTPAMIN, 1-52, 1-68, 2-78
OUTPCALC<I>, 1-51, 1-68
OUTPCALK, 1-51, 1-68
OUTPDAPT, 1-53, 2-78
OUTPDATA, 1-51, 1-68
OUTPDATF, 1-51, 1-68
OUTPDATP, 1-68
OUTPDATR, 1-53, 1-68, 2-78
OUTPERRO, 1-51, 1-68
OUTPFAIP, 1-53, 1-68, 2-78
OUTPFORF, 1-51, 1-68

OUTPFORM, 1-51, 1-68
OUTPICAL<I>, 1-51, 1-69
OUTPIDEN, 1-51, 1-69
OUTPKEY, 1-51, 1-69
OUTPLEAS, 1-51, 1-69
OUTPLIM1, 1-53, 1-69, 2-78
OUTPLIM2, 1-53, 1-69, 2-78
OUTPLIMF, 1-51, 1-69
OUTPLIML, 1-51, 1-69
OUTPLIMM, 1-52, 1-69
OUTPMARK, 1-52, 1-69
OUTPMEMF, 1-52, 1-69
OUTPMEMO, 1-52, 1-69
OUTPMSTA, 1-52, 1-69
OUTPMWID, 1-52, 1-69
OUTPMWIL, 1-52, 1-69
OUTPOPTS, 1-69
OUTPPLOT, 1-69
OUTPPRIN, 1-69
OUTPPRNALL, 1-52, 1-70
OUTPRAF<I>, 1-52, 1-70
OUTPRAW1, 1-70
OUTPRAW<I>, 1-52
OUTPSEGAF, 1-53, 1-70, 2-78
OUTPSEGAM, 1-52, 1-70, 2-78
OUTPSEGF, 1-53, 1-70, 2-78
OUTPSEGM, 1-70
OUTPSEGM[D], 1-52, 2-78
OUTPSEQ<I>, 1-51, 1-70
OUTPSERN, 1-51, 1-70, 2-78
OUTPSTAT, 1-52, 1-54, 1-70
OUTPTITL, 1-52, 1-70
output
  plot string, 1-69
output ch1 status, 1-69
output ch2 status, 1-69
output data by point, 1-68
output data by range, 1-68
output-data command, 1-15
Output Data Per Point, 2-90
Output Data Per Range of Points, 2-91
Output Limit Pass/Fail by Channel, 2-92
output limit test min/max, 1-70
Output Limit Test Pass/Fail Status Per Limit Segment, 2-84
output limit test status, 1-70
output max values, 1-68
Output Minimum and Maximum Point For All Segments, 2-88
Output Minimum and Maximum Point Per Limit Segment, 2-87
output min values, 1-68
output number of failures, 1-68
output of errors, 1-27
Output Pass/Fail Status for All Segments, 2-85
output queue, 1-15
output segment number, 1-70
output syntax, 1-15
outputting trace-related data, 1-16

P

PaintJet, 1-72
parallel poll configure, 1-13
parallel poll non response (PPO), 1-9
PARAOUT[D], 1-70
pass control, 1-80
pass control capabilities (C10), 1-9
pass-control mode, 1-11
pass control mode, 1-13
PASS/FAIL tests, 2-61
PAUS, 1-70
pause, 1-70
pause to select sequence, 1-73
PCB[D], 1-70
PC-graphics applications example program, 2-71
PCOLDATA1, 1-70
PCOLDATA2, 1-70
PCOLGRAT, 1-70
PCOLMEMO1, 1-70
PCOLMEMO2, 1-70
PCOLTEXT, 1-70
PCOLWARN, 1-70
PDATA, 1-70
PENNDATA[D], 1-70
PENNGRAT[D], 1-70
PENNMARK[D], 1-70
PENNMEMO[D], 1-70
PENNTEXT[D], 1-71
pen number
  data, 1-70
  graticule, 1-70
  markers, 1-70
  memory, 1-70
  text, 1-71
peripheral
  address, 1-55
peripheral addresses, 1-12
PGRAT, 1-71
PHAO[D], 1-71
PHAS, 1-71
phase, 1-71
phase offset, 1-71
PLOS, 1-71
PLOT, 1-71
plot data, 1-70
plot file and PC-graphics example program, 2-71

plot graticule, 1-71
plot markers, 1-71
plot memory, 1-71
plot quadrant, 1-64, 1-74
plot scale, 1-74
plot softkeys, 1-72
plot speed, 1-71
plot string
  output, 1-69
plotter
  address, 1-55
  auto feed, 1-71
  form feed, 1-71
plotter default setup, 1-59
plotter port
  disk, 1-71
  HP-IB, 1-71
plotter type, 1-71
plot text, 1-73
plotting
  to a file, 2-69
plotting, remote, 2-64, 2-66
plot to disk
  title, 1-79
PLTPRTDISK, 1-71
PLTPRTHPIB, 1-71
PLTTRAUTF, 1-71
PLTTRFORF, 1-71
PLTTYPHPGL, 1-71
PLTTYPPLTR, 1-71
PMEM, 1-71
PMKR, 1-71
PMTRTTIT, 1-71
POIN[D], 1-71
points
  specify, 1-71
POLA, 1-71
polar, 1-71
polar markers, 1-71
POLMLIN, 1-71
POLMLOG, 1-71
POLMRI, 1-71
PORE, 1-71
PORT1[D], 1-71
PORT2[D], 1-71
port extensions, 1-71
PORTR[D], 1-71
PORTT[D], 1-71
post-processing the measurement data, 2-11
POWE[D], 1-71
power level, 1-71
power loss range
  edit, 1-71
power loss table, 1-73
  edit, 1-71
power meter
  address, 1-55
power meter cal factor, 1-56
power meter into title string, 1-71
power meter type, 1-72
power ranges, 1-72
power slope, 1-76
power sweep, 1-72
power trip, 1-72
POWLFREQ[D], 1-71
POWLLIST, 1-71
POWLLOSS[D], 1-71
POWM, 1-72
POWS, 1-72
POWT, 1-72
POWT<ON|OFF>, 1-50
PPO (does not respond to parallel poll, 1-9
PRAN, 1-72
preparing for remote operation, 2-7
presetting the instrument, 2-7
PRIC, 1-72
PRINALL, 1-72
PRINSEQ<I>, 1-72
PRINTALL, 1-72
print color, 1-72
printer
  address, 1-55
  auto feed, 1-72
  form feed, 1-72
printer default setup, 1-59
printing, remote, 2-64, 2-66
print monochrome, 1-72
print sequence, 1-72
print softkeys, 1-72
PRIS, 1-72
PRNTRAUTF, 1-72
PRNTRFORF, 1-72
PRNTYP540, 1-72
PRNTYPDJ, 1-72
PRNTYPEP, 1-72
PRNTYPLJ, 1-72
PRNTYPPJ, 1-72
PRNTYPTJ, 1-72
processing after taking measurement data, 2-11
processing data chain, 1-21
process of measuring, 2-10
program debugging, 2-13
program development features, 2-13
program example
  measurement calibration, 2-18
program information, 2-13
PSOFT, 1-72
PSOFT<ON|OFF>, 1-49
PTEXT, 1-73

PTOS, 1-73
purge file, 1-73
PURG<I>, 1-73
PWRLOSS, 1-73
PWRR, 1-73

Q

Q<I>, 1-73
query command, 1-15
querying commands, 2-5
query syntax, 1-5
queue for output, 1-15

R

RAID, 1-73
RAIISOL, 1-73
RAIRESP, 1-73
raw data
  include with disk files, 1-61
raw measured data, 1-21
reading analyzer data, 1-15
REAL, 1-73
RECA<I>, 1-73
recall colors, 1-73
recall register, 1-73
recall sequence, 1-65
RECAREG<I>, 1-73
RECO, 1-73
recommended disk drives, 2-2
recommended plotters, 2-2
recommended printers, 2-2
reference line value, 1-73
reference position, 1-73
  set to mkr, 1-66
reflection, 1-57
REFT, 1-73
REFV[D], 1-73
remote enable (REN) control line, 1-7
remote/local capability (RL1), 1-9
remote mode, 1-13, 2-5
remote operation (R), 1-10
REN (remote enable) control line, 1-7
report generation, 2-64
reporting of errors, 1-24
reporting on status, 1-24
RESC, 1-73
RESD, 1-73
reset color, 1-74
RESPDONE, 1-74
response cal done, 1-74
REST, 1-74
restart averaging, 1-55
restore display, 1-73
resume cal sequence, 1-73
RFLP, 1-74
RIGL, 1-74
RIGU, 1-74
RL1 (complete remote/local capability), 1-9
routing debugging, 2-13
R (remote operation), 1-10
RSCO, 1-74
RST, 1-74
rules for code naming, 1-3

S

S11, 1-74
S21, 1-74
SADD, 1-74
SAV1, 1-74
SAVC, 1-74
save cal kit, 1-74
save colors, 1-78
save format, 1-74
SAVE<I>, 1-74
SAVE/RECALL HP-IB commands, 1-43
SAVEREG<I>, 1-74
save register, 1-74
save sequence, 1-78
SAVEUSEK, 1-74
SAVUASCI, 1-74
SAVUBINA, 1-74
SCAL[D], 1-74
scale
  auto, 1-55
SCALE REF HP-IB commands, 1-44
SCAP, 1-74
SDEL, 1-74
SDON, 1-74
SEAL, 1-75
SEAMAX, 1-75
SEAMIN, 1-75
SEAOFF, 1-75
SEAR, 1-75
SEATARG[D], 1-75
SEDI[D], 1-75
segment
  add, 1-74
  delete, 1-74
  edit, 1-75
segment edit done, 1-60
segment select, 1-77
select first point[D], 1-75
select last point[D], 1-75
select point number[D], 1-75
select segment number[D], 1-75
select sequence, 1-73, 1-75
select standard, 1-77
SELL[D], 1-49
SELMAXPT[D], 1-53, 1-75, 2-78
SELMINPT[D], 1-53, 1-75, 2-78

SELPT[D], 1-53, 1-75, 2-78
SELSEG[D], 1-53, 1-75, 2-78
sensor input selection, 1-80
SEQ HP-IB commands, 1-44
SEQ<I>, 1-75
sequence wait, 1-75
SEQWAIT[D], 1-75
serial poll, 1-13, 2-37
service request, 2-41
service request asserted by the analyzer (S), 1-10
service request (SRQ) control line, 1-7
set bandwidth, 1-62
SETBIT[D], 1-76
SETF, 1-76
setting addresses, 2-2
setting HP-IB addresses, 1-12
setting the control mode, 2-2
setting up the instrument, 2-10
setting up the system, 2-2
SETZ[D], 1-76
SH1 (full-source handshake), 1-9
SHOM, 1-76
show menus, 1-76
SING, 1-76
single bus concept, 1-10
single point type, 1-64
SLID, 1-76
sliding load, 1-76
  done, 1-76
  set, 1-76
SLIL, 1-76
SLIS, 1-76
SLOPE[D], 1-76
sloping line type, 1-64
SLOPO, 1-76
SMIC, 1-76
SMIMGB, 1-76
SMIMLIN, 1-76
SMIMLOG, 1-76
SMIMRI, 1-76
SMIMRX, 1-76
Smith chart, 1-76
Smith markers, 1-76
SMOOAPER[D], 1-76
SMOOO, 1-76
smoothing, 1-76
smoothing aperture, 1-76
SOFR, 1-49, 1-76
SOFT<I>, 1-76
SOFT[I], 1-54
SOUP, 1-76
source power on/off, 1-76
SPAN[D], 1-76
SPECFWDT, 1-77
specify class, 1-77
specify gate menu, 1-77
specify points, 1-71
SPECRESI[I], 1-77
SPECRESP[I], 1-77
SPECS11A[I], 1-77
SPECS11B[I], 1-77
SPECS11C[I], 1-77
SPEG, 1-77
SPLD, 1-77
split display, 1-77
SR1 (complete service request capabilities), 1-9
SRE[D], 1-54
SRQ (service request) control line, 1-7
SSEG[D], 1-77
S (service request asserted by the analyzer), 1-10
STANA, 1-77
STANB, 1-77
STANC, 1-77
STAND, 1-77
standard defined, 1-77
standard definition, 1-59
standard labelling, 1-64
standard offsets, 1-67
standard type, 1-77
STANE, 1-77
STANF, 1-77
STANG, 1-77
STAR[D], 1-77
statistics
  marker, 1-67
status bit definitions, 1-24
status byte, 1-24, 1-26, 2-37
STATUS CONSTANTS, 2-83
status indicators, 1-10
status reporting, 1-24, 2-37
STB?, 1-77
STDD, 1-77
STDTARBI, 1-77
STDTDELA, 1-77
STDTLOAD, 1-77
STDTOPEN, 1-77
STDTSHOR, 1-78
step 1 of a measurement, 2-10
step 2 of a measurement, 2-10
step 3 of a measurement, 2-10
step 4 of a measurement, 2-11
step 5 of a measurement, 2-11
step 6 of a measurement, 2-11
step down, 1-60
step up, 1-80
STIMULUS HP-IB commands, 1-45
stimulus menu HP-IB commands, 1-40

stimulus value
  segment, 1-64
STOP[D], 1-78
storage
  disk, 1-61
  internal memory, 1-63
store to disk, 1-78
STOR<I>, 1-78
STORSEQ<I>, 1-78
STPSIZE[D], 1-78
string for calibration kit, 1-23
structure of command syntax, 1-4
structure of HP-IB bus, 1-7
structure of status reporting, 1-24
SVCO, 1-78
SWEA, 1-78
sweep user-controlled, 2-13
SWET[D], 1-78
SWR, 1-78
synchronization, 2-37
syntax for commands, 1-3
syntax for output, 1-15
syntax structure, 1-4
syntax types, 1-5
system controller capabilities (C1,C2,C3), 1-9
system-controller mode, 1-10, 1-11
SYSTEM HP-IB commands, 1-46
system setups, 2-44
  reading calibration data, 2-46
  using the learn string, 2-44

T

T6 (basic talker), 1-9
take-control command, 1-13
taking the measurement data, 2-11
talker interface function, 1-6
talker/listener, 1-78
talker/listener mode, 1-11
TALKLIST, 1-78
talk mode (T), 1-10
TE0 (no extended talker capabilities), 1-9
TERI[D], 1-78
terminal impedance, 1-78
terminators, 1-5
TESS?, 1-49, 1-78
test port return cables, 2-2
test setup calibration, 2-10
text
  color, 1-70
ThinkJet, 1-72
TIMDTRAN, 1-78
time domain bandpass, 1-55
time domain gate, 1-62
time domain HP-IB commands, 1-47
time specify, 1-78
TINT, 1-78
TITF<I>, 1-79
TITL[$], 1-79
title
  CRT, 1-79
  plot to disk, 1-79
title disk file, 1-79
title register, 1-79
title sequence, 1-79
title string to trace memory, 1-79
title to peripheral, 1-79
title to printer, 1-79
TITP[$], 1-79
TITREG<I>, 1-79
TITR<I>, 1-79
TITSEQ<I>, 1-79
TITSQ, 1-79
TITTMEM, 1-79
TITTPERI, 1-79
TITTPRIN, 1-79
trace-data formats and transfers, 2-23
trace-data transfers, 1-19
trace memory, 1-21
trace-related data, 1-16
TRACK, 1-79
transfer of data, 1-7
transferring the measurement data, 2-11
transfers and formats of trace-data, 2-23
transfers of trace-data, 1-19
transform, 1-78
TRAP, 1-79
TRIG, 1-80
trigger
  continuous, 1-58
  external, 1-61
  hold, 1-62
  number of groups, 1-67
  single, 1-76
trigger device, 1-13
tri-state drivers (E2), 1-9
troubleshooting, 2-3, 2-5
TST?, 1-80
T (talk mode), 1-10
TTLHPULS, 1-80
TTLLPULS, 1-80
types of syntax, 1-5

U

units, 1-4
units as a function of display format, 1-17
universal commands, 1-12
UP, 1-80
upper limit
  segment, 1-64

USEPASC, 1-80
user-controllable sweep, 2-13
user-defined cal kits, 1-56
user-defined kit
  save, 1-74
user graphics
  include with disk files, 1-61
USESENSA, 1-80
USESENSB, 1-80
use sensor A, 1-80
use sensor B, 1-80

V

valid characters, 1-4
velocity factor, 1-80
VELOFACT[D], 1-80
verifying HP-IB operation, 2-2

W

WAIT, 1-80
waiting-for-group-execute-trigger, 1-13
waiting-for-reverse-get bit, 1-13
warning
  color, 1-70
warning beeper, 1-55
WAVD, 1-80
WAVE, 1-80
WIDT, 1-80
WIDV[D], 1-80
WINDMAXI, 1-80
WINDMINI, 1-80
WINDNORM, 1-80
window
  maximum, 1-80
  minimum, 1-80
  normal, 1-80
  shape, 1-80
  value, 1-80
WINDOW[D], 1-80
WINDUSEM, 1-80
WRSK<I>[$], 1-54, 1-80

Z

Z0, 1-76

Contents

1. HP-IB Programming and Command Reference

Where to Look for More Information . . . 1-2
Analyzer Command Syntax . . . 1-3
Code Naming Convention . . . 1-3
Valid Characters . . . 1-4
Units . . . 1-4
Command Formats . . . 1-4
General Structure: . . . 1-4
Syntax Types . . . 1-5
HP-IB Operation . . . 1-6
Device Types . . . 1-6
Talker . . . 1-6
Listener . . . 1-6
Controller . . . 1-6
HP-IB Bus Structure . . . 1-7
Data Bus . . . 1-7
Handshake Lines . . . 1-7
Control Lines . . . 1-7
HP-IB Requirements . . . 1-8
HP-IB Operational Capabilities . . . 1-9
HP-IB Status Indicators . . . 1-10
Bus Device Modes . . . 1-10
System-Controller Mode . . . 1-11
Talker/Listener Mode . . . 1-11
Pass-Control Mode . . . 1-11
Analyzer Bus Modes . . . 1-11
Setting HP-IB Addresses . . . 1-12
Response to HP-IB Meta-Messages (IEEE-488 Universal Commands) . . . 1-12
Abort . . . 1-12
Device Clear . . . 1-12
Local . . . 1-12
Local Lockout . . . 1-13
Parallel Poll . . . 1-13
Pass Control . . . 1-13
Remote . . . 1-13
Serial Poll . . . 1-13
Trigger . . . 1-13
Analyzer Operation . . . 1-14
Operation Complete . . . 1-14
Reading Analyzer Data . . . 1-15
Output Queue . . . 1-15
Command Query . . . 1-15
Identification . . . 1-15
Output Syntax . . . 1-15
Marker data . . . 1-16
Array-Data Formats . . . 1-17

Trace-Data Transfers 1-19
Stimulus-Related Values 1-19
Data-Processing Chain 1-21
Data Arrays 1-21
Fast Data Transfer Commands 1-22
Data Levels 1-22
Learn String and Calibration-Kit String 1-23
Error Reporting 1-24
Status Reporting 1-24
The Status Byte 1-26
The Event-Status Register and Event-Status-Register B 1-26
Error Output 1-27
Calibration 1-28
Disk File Names 1-29
Using Key Codes 1-30
Key Select Codes Arranged by Front-Panel Hardkey 1-31
HP-IB Only Commands 1-49
Alphabetical Mnemonic Listing 1-55

Index

Figures

1-1. HP-IB Bus Structure 1-7
1-2. Analyzer Single Bus Concept 1-10
1-3. FORM 4 (ASCII) Data-Transfer Character String 1-16
1-4. The Data-Processing Chain 1-21
1-5. Status Reporting Structure 1-24
1-6. Key Codes 1-30

Tables

1-1. Code Naming Convention 1-3
1-2. OPC-compatible Commands 1-14
1-3. Units as a Function of Display Format 1-17
1-4. HP 8752C Network Analyzer Array-Data Formats 1-18
1-5. Status Bit Definitions 1-25
1-6. Calibration Arrays 1-28
1-7. Disk File Names 1-29
1-8. Key Select Codes 1-32
1-9. HP-IB Only Commands 1-49

1

HP-IB Programming and Command Reference

This chapter is a reference for operation of the 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:

- Analyzer Command Syntax
  - Code Naming Convention
  - Valid Characters
  - Units
  - Command Formats
- HP-IB Operation
  - Device Types
  - HP-IB Bus Structure
  - HP-IB Requirements
  - HP-IB Operational Capabilities
  - Bus Device Modes
  - Setting HP-IB Addresses
  - Response to HP-IB Meta-Messages (IEEE-488 Universal Commands)
- Analyzer Operation-Complete Commands
- Reading Analyzer Data
  - Output Queue
  - Command Query
  - Identification
  - Output Syntax
  - Marker Data
  - Array-Data Formats
  - Trace Data Transfers
  - Stimulus-Related Values
- Data Processing Chain
  - Data Arrays
  - Fast Data Transfer Commands
  - Data Levels
  - Learn String and Calibration-Kit String

- Error Reporting
  - Status Reporting
  - The Status Byte
  - The Event-Status Register and Event-Status Register B
  - Error Output
- Calibration
- Disk File Names
- Using Key Codes
- Key Select Codes Arranged by Front-Panel Hardkey
- HP-IB Only Commands
- Alphabetical Mnemonic Listing

For information about manual operation of the analyzer, refer to the *HP 8752C Network Analyzer User's Guide.*

Where to Look for More Information

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

- *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.
- *IEEE Standard Digital Interface for Programmable Instrumentation ANSI/IEEE std 488.1-1987* contains detailed information on IEEE-488 operation. Published by the Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, New York 10017.
- Chapter 2, "HP BASIC Programming Examples," includes programming examples in HP BASIC.

Analyzer Command Syntax

Code Naming Convention

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

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 category, as in markers or plotting pen numbers, the command is increased to 8 letters. The first 4 letters are the category label and the last 4 letters are the function specifier. As an example, category pen numbers are represented by the command PENN, which is used in combination with several functions such as PENNDATA, PENNMEMO.

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

- make commands more meaningful and easier to remember
- maintain compatibility with other products (including the HP 8510)

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

Table 1-1. Code Naming Convention

Convention	Key Title	For HP-IB Code Use	Example
One Word	Power	First Four Letters	POWE
	Start		STAR
Two Words	Electrical Delay	First Three Letters of First Word, First Letter of Second Word	ELED
	Search Right		SEAR
Two Words in a Group	Marker →Center	Four Letters of Both	MARKCENT
	Gate →Span		GATESPAN
Three Words	Cal Kit N 50 Ω	First Three Letters of First Word, First Letter of Second Word, First Four Letters of Third Word	CALKN50
	Pen Num Data		PENNDATA

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.

Valid Characters

The analyzer accepts the following ASCII characters:

- letters
- numbers
- decimal points
- +/–
- semicolons (;)
- quotation marks (")
- carriage returns (CR)
- 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. If the analyzer does not recognize a character as appropriate, it generates a syntax error message and recovers at the next terminator.

Units

The analyzer can input and output data in basic units such as Hz, dB, seconds, etc.

S	Seconds	HZ	Hertz
V	Volts	DB	dB or dBm

Input data is assumed to be in basic units (see above) unless one of the following units is used (upper and lower case are equivalent):

MS	Milliseconds	KHZ	Kilohertz
US	Microseconds	MHZ	Megahertz
NS	Nanoseconds	GHZ	Gigahertz
PS	Picoseconds	FS	Femtoseconds

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 (these codes are described in the "Alphabetical Mnemonic Listing" later in this chapter.)

[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 capability 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 enclosed by double quotation marks.
For example:
A title string using RMB BASIC would look like:
`OUTPUT 716;"TITL""""Unit1""";"`
where the first two "" are an escape so that RMB BASIC will interpret the third " properly.

[unit] The units of the operand, if applicable. If no units are specified, the analyzer assumes the basic units as described previously. 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.

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.

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 `CORRON;` and `CORROFF;` (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.

Note In the following cases: CLEAREG[D], RECAREG[D], SAVEREG[D], and EG[D], [D] must be 2 characters. For example, `CLEAREG01;` will execute, while `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 `TITR1 "STATE1"` (title register 1 STATE1), `TITR2 "TEST2"` (title register 2 TEST2).

QUERY SYNTAX: [code][?]

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

HP-IB Operation

The Hewlett-Packard Interface Bus (HP-IB) is Hewlett-Packard's hardware, software, documentation, and support for IEEE 488.2 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

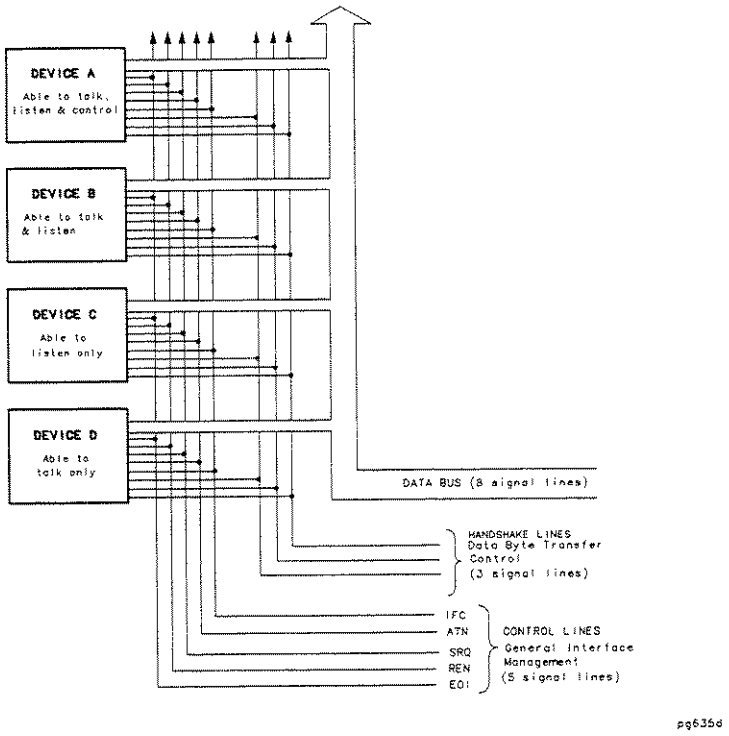


Figure 1-1. HP-IB Bus Structure

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 mode 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 asynchronous 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 (such as this instrument), 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.

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.

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

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

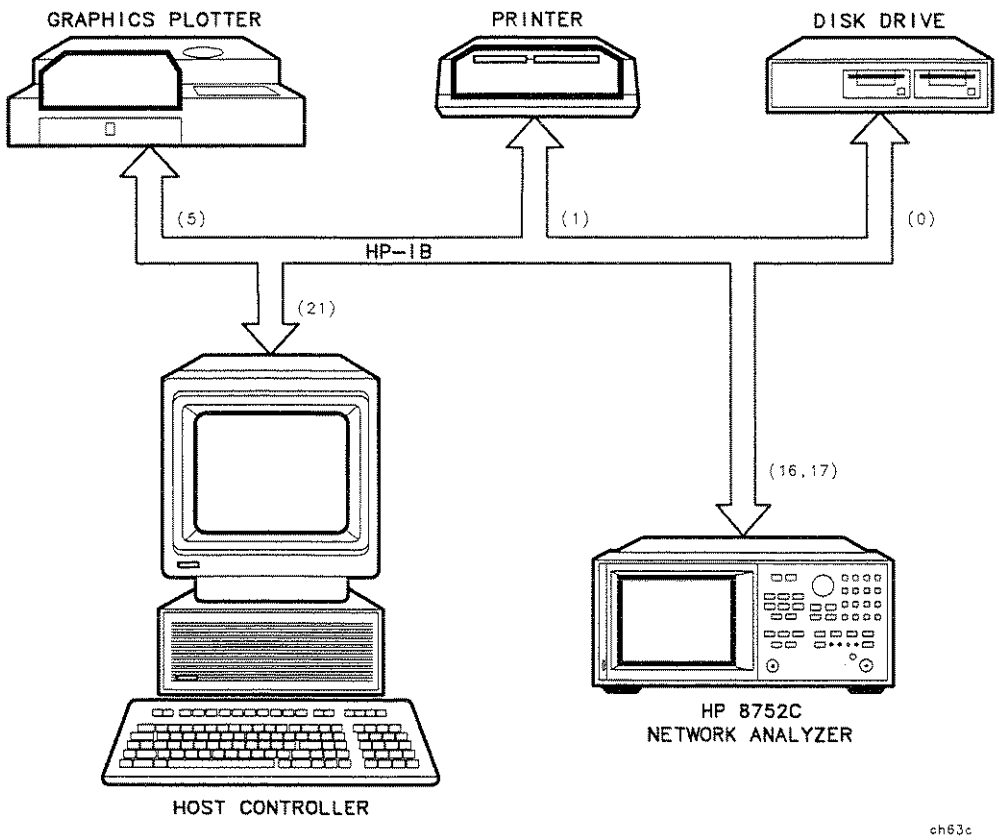


Figure 1-2. Analyzer Single Bus Concept

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:

```
"ANOTHER SYSTEM CONTROLLER ON HP-IB BUS"
```

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 8752C 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 `OUTPPLOT;` or `OUTPPRIN;` commands. The analyzer will wait until it is addressed 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.

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. 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.

Operation Complete

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. 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 querying the analyzer causes the bus to halt until the instrument preset is complete and the analyzer outputs a one (1).

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 should be programmed to read the number one (1) response from the analyzer indicating completion of the single sweep. At this point a valid trace exists and the trace data could be read into the computer.

Table 1-2. OPC-compatible Commands

CHAN1	EXTTPOIN	RST
CHAN2	GATEO<ON|OFF>	SAV1
CLASS11A[1]	ISOD	SAVC
CLASS11B[1]	MANTRIG	SAVE<1 to 5>
CLASS11C[1]	NOOP	SAVEREG<01 to 31>
CLEA<1 to 5>	NUMG	SING
CLEARALL	PRES	SLIS
CLEAREG<01 to 31)	RAID	STAN<A to G>
DATI	RECA<1 to 5>	TIMDTRAN<ON|OFF>
EXTTOFF	RECAREG<01 to 31>	WAIT
EXTTON	RESPDONE	

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

Reading Analyzer Data

Output Queue

Whenever an 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 query or data request from the analyzer.

Command Query

All instrument functions can be queried to find the current ON/OFF state or value. For instrument state commands, append the question mark character (?) to the command to query 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-query 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 to the controller.

ON/OFF commands can also be queried. 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, querying 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 queried.

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.

Identification

The analyzer's response to `IDN?;` is "HEWLETT PACKARD,8752C,0,X.XX" where X.XX is the firmware revision of the instrument.

The analyzer also has the capability to output its serial number with the command `OUTPSERN;`, and to output its installed options with the command `OUTPOPTS;`.

Output Syntax

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

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

Marker-output commands and queried 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:

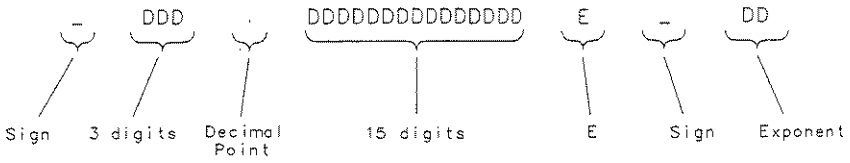


Figure 1-3. FORM 4 (ASCII) Data-Transfer Character String

Sign	'-' for negative, blank for positive.
3 digits	Digits to the left of the decimal point.
Decimal point	Standard decimal point.
15 digits	Digits to the right of the decimal point.
E	Exponent notation.
Sign	'-' for negative, '+' for positive.
Exponent	Two digits for the exponent.

When multiple numbers are sent, the numbers are separated by commas. When number pairs are sent, the numbers are separated by a comma and terminated with a line feed (LF).

Marker data

The network analyzer offers several options for outputting trace-related data. 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. Specific data points can be read using the OUTPDATP or OUTPDATR commands. These commands allow a much faster data transfer than when using markers to output specific data points. For more information on these commands, see "Limit Line and Data Point Special Functions," located in Chapter 2.

A marker must first be assigned to the desired frequency before it can be used to read the trace data. 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 by using analyzer commands. These commands cause the analyzer to transmit three numbers: marker value 1, marker value 2, and marker stimulus value. For example, in log-magnitude display mode we get the log magnitude at the marker (value 1), zero (value 2), and the marker frequency. See Table 1-3 for a complete listing of all the possibilities for values 1 and 2. The four possibilities for the marker stimulus value are:

- frequency
- time (as in time domain, Option 010 Only)
- CW time
- power (in power sweep mode)

Table 1-3. Units as a Function of Display Format

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

*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 query, or output marker values.

A transmitted array will be output 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:

FORM1 — 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.

FORM2 — 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.

FORM3 — 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.

FORM4 — ASCII floating-point format. The data is transmitted as ASCII numbers, as described previously in "Output Syntax".

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

FORM5 — 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 1-4 offers a comparative overview of the five array-data formats.

Table 1-4. HP 8752C Network Analyzer Array-Data Formats

Format type	Type of Data	Bytes per Data Value	Bytes per point 2 data values	(201 pts) Bytes per trace	Total Bytes with header
FORM 1	Internal Binary	3	6	1206	1210
FORM 2	IEEE 32-bit Floating-Point	4	8	1608	1612
FORM 3	IEEE 64-bit Floating-Point	8	16	3216	3220
FORM 4	ASCII Numbers	24 (Typical)	50 (Typical)	10,050 (Typical)	10,050* (Typical)
FORM 5	PC-DOS 32-bit Floating-Point	4	8	1608	1612

*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. Programming examples of data transfers using different formats are discussed in "Example 3: Measurement Data Transfer," located in Chapter 2. 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.

For information on transferring trace-data by selected points, see "Limit Line and Data Point Special Functions," located in Chapter 2.

Note "Example 7C: Reading ASCII Disk Files to the System Controller Disk File," located in Chapter 2, explains how to access disk files from a computer.

Stimulus-Related Values

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

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 querying 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:

$$F = \text{Start frequency} + (n-1) \times \text{Span}/(\text{Points}-1)$$

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 Transfer)," located in Chapter 2.

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, an effective way of determining the frequencies of the current sweep is to use the OUTPLIML command. Although this command is normally used for limit lines, it can also be used to identify all of the frequency points in a sweep. Limit lines do not need to be on in order to read the frequencies directly out of the instrument with the OUTPLIML command. Refer to "Example 3D: Data Transfer Using Frequency-Array Information," located in Chapter 2.

Note Another method of identifying all of the frequency points in a sweep is to use the marker commands MARKBUCKx and OUTPMARK in a FOR NEXT programming loop that corresponds to the number of points in the sweep. MARKBUCKx places a marker at a point in the sweep, where x is the number of the point in a sweep, and OUTPMARK outputs the stimulus value as part of the marker data.

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 1-4 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.

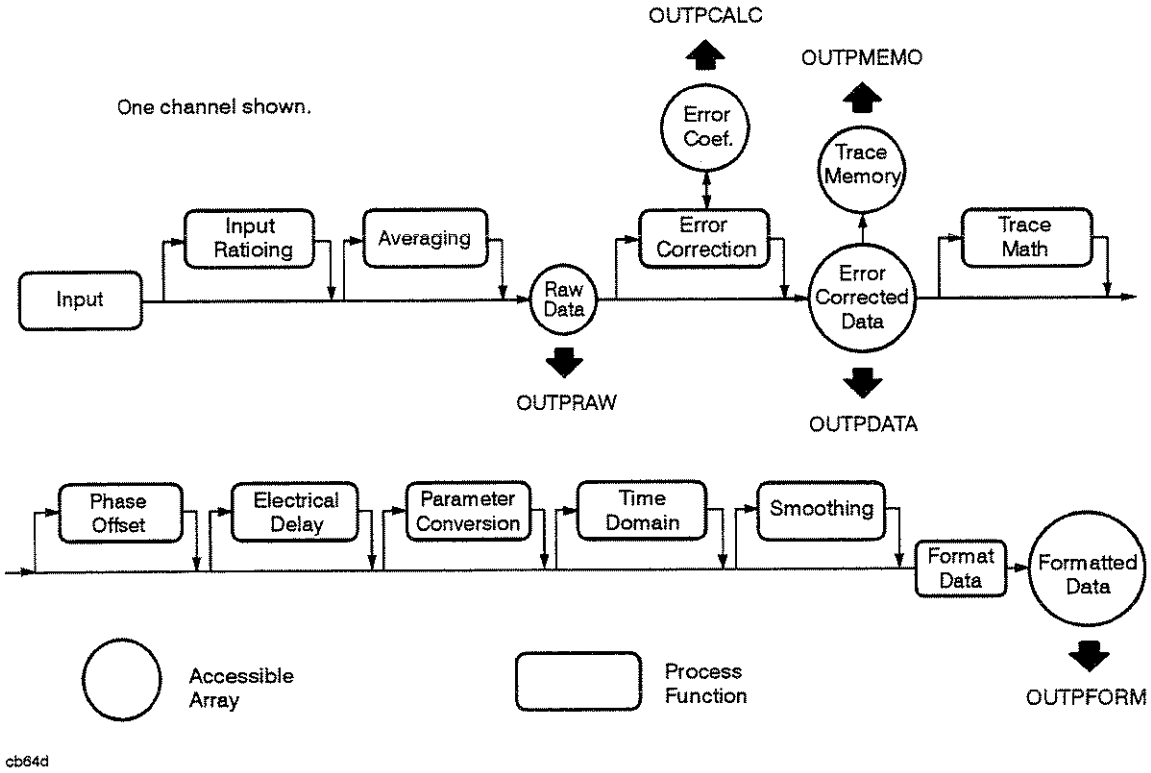


Figure 1-4. The Data-Processing Chain

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.

Fast Data Transfer Commands

The HP 8752C has four distinct fast data transfer commands. These commands circumvent the internal "byte handler" routine and output trace dumps as block data. In other words, the analyzer outputs the entire array without allowing any process swapping to occur. FORM4, ASCII data transfer times are not affected by these routines. However, there are speed improvements with binary data formats. The following is a description of the four fast data transfer commands:

- OUTPDATF outputs the error corrected data from the active channel in the current output format. This data may be input to the analyzer using the INPUDATA command.
- OUTPFORF outputs the formatted display trace array from the active channel in the current output format. Only the first number in each of the OUTPFORM data pairs is actually transferred for the display formats `LOG MAG`, `PHASE`, group `DELAY`, `LIN MAG`, `SWR`, `REAL` and `IMAG`inary. Because the data array does not contain the second value for these display formats, the INPUFORM command may not be used to re-input the data back into the analyzer. The second value may not be significant in some display formats (see Table 1-4), thus reducing the number of bytes transferred.
- OUTPMEMF outputs the memory trace from the active channel. The data is in real/imaginary pairs, and, as such, may be input back into the memory trace using INPUDATA or INPUFORM followed by the DATI command.
- OUTPRAF<I> outputs the raw measurement data trace. The data may be input back into the memory trace using the INPURAW<I> command.

Data Levels

Different levels of data can be read out of the instrument. Refer to the data-processing chain in Figure 1-4. 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.
Error coefficients	The results of a measurement calibration are arrays containing error coefficients. These error coefficients are then used in the error-correction routines. Each array corresponds to a specific error term in the error model. The *HP 8752C Network Analyzer User's Guide* 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 3 arrays. The data is stored as 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 `OUTPMEMO;`

	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 1-3 for the various units defined as a function of display format.

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. The array has the same header as in FORM 1.

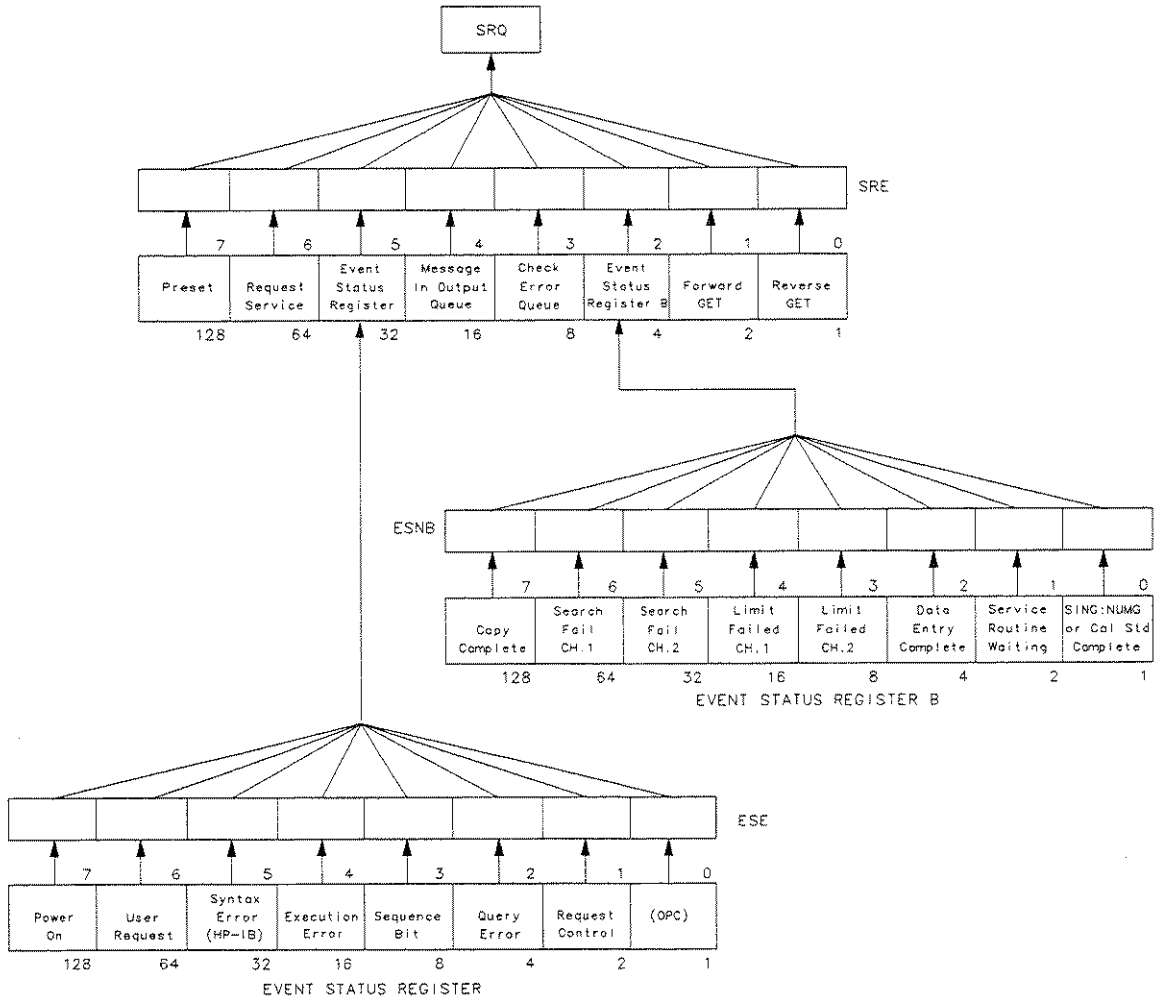
The calibration kit is a set of key characteristics of the calibration standards used to increase the calibration accuracy. There are default kits for several different connector types. There is also space for a user-defined calibration kit. The command OUTPCALK outputs the currently active calibration 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 1-5.



pg6151d

Figure 1-5. Status Reporting Structure

Table 1-5. Status Bit Definitions

Status Byte		
Bit	**Name**	**Definition**
0		This bit is not used with the HP 8752C network analyzer.
1		This bit is not used with the HP 8752C network analyzer.
2	Check event-status-register B	One of the enabled bits in event status register B has been set.
3	Check error queue	An error has occurred and the message has been placed in the error queue, but has not been read yet.
4	Message in output queue	A command has prepared information to be output, but it has not been read yet.
5	Check event-status register	One of the enabled bits in the event-status register has been set.
6	Request service	One of the enabled status-byte bits is causing an SRQ.
7	Preset	An instrument preset has been executed.
Event-Status Register		
Bit	**Name**	**Definition**
0	Operation complete	A command for which OPC has been enabled has completed operation.
1	Request control	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.
2	Query error	The analyzer has been addressed to talk but there is nothing in the output queue to transmit.
3	Sequence Bit	A sequence has executed the assert SRQ command.
4	Execution error	A command was received that could not be executed.
5	Syntax error	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.
6	User request	The operator has pressed a front-panel key or turned the RPG.
7	Power on	A power-on sequence has occurred since the last read of the register.
Event-Status-Register B		
Bit	**Name**	**Definition**
0	Single sweep, number of groups, or calibration step complete	A single sweep, group, or calibration step has been completed since the last read of the register.
1	Service routine waiting or done	An internal service routine has completed operation, or is waiting for an operator response.
2	Data entry complete	A terminator key has been pressed or a value entered over HP-IB since the last read of the register.
3	Limit failed, Channel 2	Limit test failed on Channel 2.
4	Limit failed, Channel 1	Limit test failed on Channel 1.
5	Search failed, Channel 2	A marker search was executed on Channel 2, but the target value was not found.
6	Search failed, Channel 1	A marker search was executed on Channel 1, but the target value was not found.
7	Copy Complete	A copy has been completed since the last read of the register.

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 `SREnn;` 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 `ESNBnn;`, both of which work in the same manner as `SREnn`. The units variable *nn* 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.

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 reflection 1-port (`CALIS111;` over HP-IB).
3. Call each class used by the calibration type, such as `FORWARD: OPEN` (`CLASS11A;` over HP-IB).
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 calibration kits can have a different number of standards in a given class, any automated calibration sequence is valid only for a specific calibration kit.

Table 1-6. Calibration Arrays

Array	Response	Response and Isolation	Reflection 1-port
1	ER or ET	E_X (E_D)[1]	E_D
2		E_T (E_R)	E_S
3			E_R

1 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

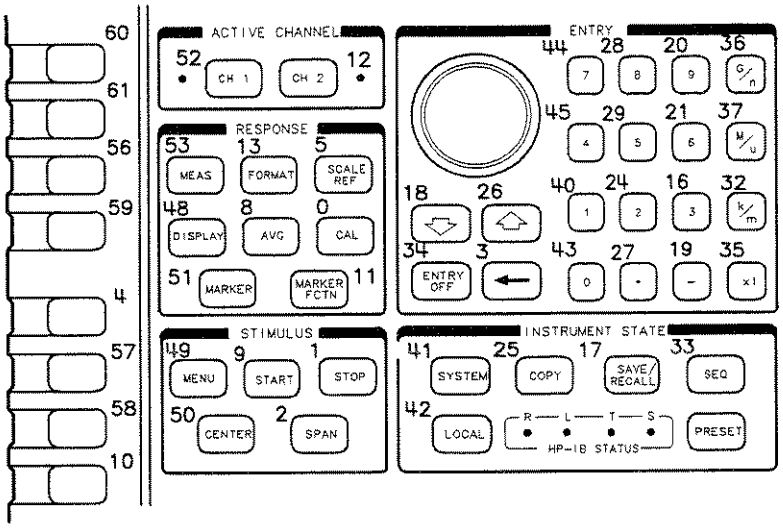
Disk File Names

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 1-7. Disk File Names

Char 1	Meaning	Char 2	Meaning
I	Instrument state		
G	Graphics	1	Display graphics
D	Error-corrected data	1	Channel 1
		2	Channel 2
R	Raw data	1 to 4	Channel 1, raw arrays 1 to 4
		5 to 8	Channel 2, raw arrays 1 to 4
F	Formatted data	1	Channel 1
		2	Channel 2
M	Memory trace	1	Channel 1
		2	Channel 2
P	Instrument state appendix		
C	Cal kit	K	
1	Cal data, channel 1		
		O	Stimulus state
		1 to 9	Coefficients 1 to 9
		A	Coefficient 10
		B	Coefficient 11
		C	Coefficient 12
2	Cal data, channel 2	0 to C	Same as channel 1
F	Full page (HP-GL plot)	P	
L	Left (HP-GL plot)	L	Lower
		U	Upper
R	Right (HP-GL plot)	L	Lower
		U	Upper

Using Key Codes



ch61c

Figure 1-6. 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. If the knob count was negative, no modification is needed. If the knob count was positive, however, bit 14 will not be set. In this case, the number must be decoded by clearing the most significant byte, as by AND'ing the integer with 255.

Key Select Codes Arranged by Front-Panel Hardkey

The HP-IB mnemonics in this table are functionally arranged by their front-panel key equivalent in the order shown:

Keys
AVG
CAL-Error correction, calibration
CAL-Calibration kits
CHANNEL
COPY
DISPLAY
ENTRY
FORMAT
LOCAL
MEAS
MENU (stimulus)
MARKER
MARKER FCTN
SAVE/RECALL-Internal registers
SAVE/RECALL-Disk files
SCALE REF
SEQ-Sequencing
STIMULUS
SYSTEM
SYSTEM-Limit testing
SYSTEM-Transform

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."
?	Query 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 1-8. Key Select Codes

Function	Action	Mnemonic	S	?	O	Range
AVG						
Averaging	Restart	AVERREST	1			
	Factor	AVERFACT[D]	3	D		0 to 999
	On/off	AVER0<ON|OFF>	2	1,0		
Smoothing	Set aperture	SMOOAPER[D]	3	D		0.05 to 20%
	On/off	SMOOO<ON|OFF>	2	1,0		
IF bandwidth	Set bandwidth	IFBW[D]	3	D		10, 30, 100, 300, 1000, 3000, 3700 Hz
CAL-error correction, calibration						
Correction	On/off	CORR<ON|OFF>	2	1,0		
Interpolative correction	On/off	CORI<ON|OFF>	2	1,0		
Cal sequence	Resume	RESC	1			
Port extensions	Reflection	PORTR[D]	3	D		±10 s
		PORT1[D]	3	D		±10 s
	Transmission	PORTT[D]	3	D		±10 s
		PORT2[D]	3	D		±10 s
	Off	PORE<ON|OFF>	2	1,0		
Velocity factor	Set value	VELOFACT[D]	3	D		0 to 10
ZO	Set Value	SETZ[D]	3	D		0.1 to 500Ω
Begin cal sequence	Response	CALIRESP	1	0,1		
	Response and Isol	CALIRAI	1	0,1		
	Reflection 1-port	CALIS111	1	0,1		
	None	CALN	1	0,1		
Select response & isol. class	Response	RAIRESP	1			
	Isolation	RAIISOL	1			
Select reflection class	RFL: (open)	CLASS11A	1		OPC††	
	RFL: (short)	CLASS11B	1		OPC††	
	RFL: (load)	CLASS11C	1		OPC††	

†† 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.

Table 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
CAL-error correction, calibration (continued)						
Select standard in class	Standard A	STANA	1		OPC	
	Standard B	STANB	1		OPC	
	Standard C	STANC	1		OPC	
	Standard D	STAND	1		OPC	
	Standard E	STANE	1		OPC	
	Standard F	STANF	1		OPC	
	Standard G	STANG	1		OPC	
Sliding load	Set	SLIS	1		OPC	
	Done	SLID	1			
Done with:	Class	DONE	1			
	Isolation	ISOD	1		OPC	
Save cal	Response	RESPDONE	1		OPC	
	Resp and isol	RAID	1		OPC	
	1-port cal	SAV1	1		OPC	
CAL-calibration kits						
Select default kits	7 mm	CALK7MM	1	1,0		
	3.5 mmC	CALK35MM	1	1,0		
	3.5 mmD	CALK35MD	1	1,0		
	Type N, 50 ohm	CALKN50	1	1,0		
	Type N, 75 ohm	CALKN75	1	1,0		
	User-defined	CALKUSED	1	1,0		
Modify kit	Modify current	MODI1	1			
Define std. number (begin std. definition)		DEFS[D]	3			
Define std. type	Open	STDTOPEN	1	1,0		
	Short	STDTSHOR	1	1,0		
	Load	STDTLOAD	1	1,0		
	Delay/thru	STDTDELA	1	1,0		
	Arbitrary imped.	STDTARBI	1	1,0		
Define std. parameters	Open cap. C0	C0[D]	3	D		$\pm$10k (10^{-15} F)
	Open cap. C1	C1[D]	3	D		$\pm$10k (10^{-27} F/Hz)
	Open cap. C2	C2[D]	3	D		$\pm$10k (10^{-36} F/Hz^2)
	Open cap. C3	C3[D]	3	D		$\pm$10k (10^{-45} F/Hz^3)
	Fixed load	FIXE	1			
	Sliding load	SLIL	1			
	Terminal imped.	TERI[D]	3	D		0 to 1 kΩ

Table 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
CAL-calibration kits (continued)						
Define std. offsets	Delay	OFSD[D]	3	D		±1 s
	Loss	OFSL[D]	3	D		0 to 1000 TΩ/s
	Z0	OFSZ[D]	3	D		0.1 to 500Ω
	Min. frequency	MINF[D]	3	D		0 to 1000 GHz
	Max. frequency	MAXF[D]	3	D		0 to 1000 GHz
	Coaxial	COAX	1	0,1		
	Waveguide	WAVE	1	0,1		
Std. done	Standard defined	STDD	1			
Label std		LABS[$]	3			10 char.
Specify class	Response	SPECRESP[I,I..]	3			Std numbers
	Resp & Isol	SPECRESI[I,I..]	3			Std numbers
	RFL: (open)	SPECS11A[I,I..]	3			Std numbers
	RFL: (short)	SPECS11B[I,I..]	3			Std numbers
	RFL: (load)	SPECS11C[I,I..]	3			Std numbers
	Forward Trans	SPECFWDT[I,I..]	3			Std numbers
Class done		CLAD	1			
Label class	Response	LABERESP[$]	3			10 char.
	Resp. & isolation	LABERESI[$]	3			10 char.
	RFL: (open)	LABES11A[$]	3			10 char.
	RFL: (short)	LABES11B[$]	3			10 char.
	RFL: (load)	LABES11C[$]	3			10 char.
	Forward Trans	LABEFWDT[$]	3			10 char.
Label kit		LABK[$]	3			10 char.
Kit done		KITD	1			
Save kit	Into user kit	SAVEUSEK	1			
CHANNEL						
Channel	CH 1 active	CHAN1	1		OPC	
	CH 2 active	CHAN2	1		OPC	

Table 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
COPY						
Copy display	To printer§	PRINALL	1			
	To plotter§	PLOT	1			
Copy display talker/listener	To plotter	OUTPPLOT	1			
	To printer	OUTPPRIN	1			
Printer	Auto feed	PRNTRAUTF<ON|OFF>	2	1,0		
Printer	Form feed	PRNTRFORF	1			
Printer setup	Default	DEFLPRINT	1			
Plotter	Auto feed	PLTTRAUTF<ON|OFF>	2	1,0		
Plotter	Form feed	PLTTRFORF	1			
Plotter setup	Default	DFLT	1			
List values		LISV	1			
Operating parameters		OPEP	1			
Print list values or operating and marker parameters	Raster display dump to HP-IB§	PRINTALL	1			
Next page		NEXP	1			
Restore display		RESD	1			
Select print color	Monochrome	PRIS	1			
	Color	PRIC	1			
Print feature color	Data channel 1	PCOLDATA1<color>	2			Colors‡
	Data channel 2	PCOLDATA2<color>	2			Colors‡
	Memory channel 1	PCOLMEMO1<color>	2			Colors‡
	Memory channel 2	PCOLMEMO2<color>	2			Colors‡
	Graticule	PCOLGRAT<color>	2			Colors‡
	Text	PCOLTEXT<color>	2			Colors‡
	Warning	PCOLWARN<color>	2			Colors‡

§ Requires pass control mode.

‡ Colors: white|cyan|magenta|blue |yellow|green|red|black.

Table 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
COPY (continued)						
Quadrant	Left lower	LEFL	1	0,1		
	Left upper	LEFU	1	0,1		
	Right lower	RIGL	1	0,1		
	Right upper	RIGU	1	0,1		
	Full page	FULP	1	0,1		
Features to be plotted	Data	PDATA<ON|OFF>	2	1,0		
	Memory	PMEM<ON|OFF>	2	1,0		
	Graticule	PGRAT<ON|OFF>	2	1,0		
	Text	PTEXT<ON|OFF>	2	1,0		
	Marker	PMKR<ON|OFF>	2	1,0		
Pen number	Data	PENNDATA[D]	3	D		0,1,2 . . . 10
	Memory	PENNMEMO[D]	3	D		0,1,2 . . . 10
	Graticule	PENNGRAT[D]	3	D		0,1,2 . . . 10
	Text	PENNTEXT[D]	3	D		0,1,2 . . . 10
	Marker	PENNMARK[D]	3	D		0,1,2 . . . 10
Line type	Data	LINTDATA[D]	3	D		0,1,2 . . . 10
	Memory	LINTMEMO[D]	3	D		0,1,2 . . . 10
Plot scale	Full page	SCAPFULL	1			
	Graticule to p1,p2	SCAPGRAT	1			
Plot speed	Slow	PLOSSLOW	1			
	Fast	PLOSFAST	1			
DISPLAY						
Channels	Dual on/off	DUAC<ON|OFF>	2	1,0		
	Split on/off	SPLD<ON|OFF>	2	1,0		
	D2/D1 to D2	D1DIVD2<0N|OFF>	2	1,0		
Display	Data	DISPDATA	1	0,1		
	Memory only	DISPMEMO	1	0,1		
	Data and mem	DISPDATM	1	0,1		
	Data/mem	DISPDDM	1	0,1		
	Data — mem	DISPDMM	1	0,1		
	Data to mem	DATI	1	0,1	OPC	
Beeper	On done	BEEPDONE<ON|OFF>	2	1,0		
	On warning message	BEEPWARN<ON|OFF>	2	1,0		
CRT	Intensity	INTE[D]	3	D		0 to 100
	Focus	FOCU[D]	3	D		0 to 100
	Title	TITL[$]	4	$		48 char.

Table 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
DISPLAY (continued)						
Frequency notation	Blank	FREO	1			
Adjust display	Background intensity	BACI[D]	3	D		0 to 100
	Save colors	SVCO	1			
	Recall colors	RECO	1			
	Default colors	DEFC	1			
Modify colors	Ch 1 data/lim ln	COLOCH1D	1			
	Ch 1 memory	COLOCH1M	1			
	Ch 2 data/lim ln	COLOCH2D	1			
	Ch 2 memory	COLOCH2M	1			
	Graticule	COLOGRAT	1			
	Text	COLOTEXT	1			
	Warning	COLOWARN	1			
Adjust color	Brightness	CBRI[D]	3	D		0 to 100
	Color	COLOR[D]	3	D		0 to 100
	Tint	TINT[D]	3	D		0 to 100
	Reset	RSCO	1			
ENTRY						
Step keys	Up	UP	1			
	Down	DOWN	1			
Entry off		ENTO	1			
FORMAT						
Format	Log mag	LOGM	1	0,1		
	Phase	PHAS	1	0,1		
	Delay	DELA	1	0,1		
	Smith chart	SMIC	1	0,1		
	Polar	POLA	1	0,1		
	Lin mag	LINM	1	0,1		
	Real	REAL	1	0,1		
	Imaginary	IMAG	1	0,1		
	SWR	SWR	1	0,1		

Table 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
		LOCAL				
HP-IB modes	Talker/listener	TALKLIST	1	0,1		
	Use pass control	USEPASC	1	0,1		
Debug	Display commands	DEBU<ON|OFF>	2	1,0		
Disk drive	Unit	DISCUNIT[D]	3	D		0 to 30
	Volume	DISCVOLU[D]	3	D		0 to 30
HP-IB addresses	Plotter	ADDRPLOT[D]	3	D		0 to 30
	Printer	ADDRPRIN[D]	3	D		0 to 30
	Disk drive	ADDRDISC[D]	3	D		0 to 30
	Controller	ADDRCONT[D]	3	D		0 to 30
Power meter	Address	ADDRPOWM[D]	3	D		0 to 30
	Type	POWM<ON|OFF>	2	1,0		On=436A, Off=438A/437B
Plotter type	Plotter	PLTTYPPLTR	1			
	HPGL printer	PLTTYPHPGL	1			
Printer type	ThinkJet	PRNTYPTJ	1			
	DeskJet	PRNTYP540	1			
	DeskJet	PRNTYPDJ	1			
	LaserJet	PRNTYPLJ	1			
	PaintJet	PRNTYPPJ	1			
	Epson-P2	PRNTYPEP	1			
Plotter port	HP-IB	PLTPRTHPIB	1			
	Disk	PLTPRTDISK	1			
		MEAS				
Reflection	Measure	RFLP	1	0,1		
		S11	1	0,1		
Transmission	Measure	TRAP	1	0,1		
		S21	1	0,1		
Input ports	A/R	AR	1	0,1		
	B/R	BR	1	0,1		
	A/B	AB	1	0,1		
	A	MEASA	1	0,1		
	B	MEASB	1	0,1		
	R	MEASR	1	0,1		

Table 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
MEAS (continued)						
Conversion to alternate parameters	Off	CONVOFF	1	0,1		
	Z:reflection	CONVZREF	1	0,1		
	Z:transmission	CONVZTRA	1	0,1		
	Y:reflection	CONVYREF	1	0,1		
	Y:transmission	CONVYTRA	1	0,1		
	1/S	CONV1DS	1	0,1		
Analog input	Auxiliary input	ANAI[D]	1*	0,1		

* Syntax type 1 when ANABOFF. Syntax type 3, and range = 1 to 31, when ANABON.

Table 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
MENU (stimulus)						
Power	Level	POWE[D]	3	D		−22 to +15 dBm‡
	Always couple power	COUP<ON|OFF>	2	1,0		
	Range 0	PRAN0	1			
	Range 1	PRAN1	1			
	Range 2	PRAN2	1			
	Range 3	PRAN3	1			
	Range 4	PRAN4	1			
	Range 5	PRAN5	1			
	Range 6	PRAN6	1			
	Range 7	PRAN7	1			
	Power range auto/manual	PWRR<PAUTO|PMAN>	2			
	Source power on/off	SOUP<ON|OFF>	2	1,0		
Time	Specify	SWET[D]	3	D		0.01 to 86,400 s
	Selects fastest sweep time	SWEA	1			
Measurement	Restart	REST	1			
Trigger	Hold	HOLD	1	0,1		
	Single	SING	1		OPC	
	Number of groups	NUMG[D]	3	D	OPC	1 to 999
	Continuous	CONT	1	0,1		
	External trigger off	EXTTOFF	1	0,1	OPC	
	External trigger on sweep	EXTTON	1	0,1	OPC	
	External trigger on point	EXTTPOIN	1	0,1	OPC	
	Manual trigger on point	MANTRIG	1	0,1	OPC	OPC
Points	Specify	POIN[D]	3	D		3, 11, 26, 51, 101 201, 401, 801, 1601
Coupled channels	On/off	COUC<ON|OFF>	2	1,0		
CW freq	Set value	CWFREQ[D]	3	D		Stimulus range†
Power slope	Value	SLOPE[D]	3	D		−2 to +2 dB/GHz
	On/off	SLOPO<ON|OFF>	2	1,0		

‡ −85 to +20 dBm for Option 004.

† For frequency sweeps: 300 kHz to 1.3 GHz. (300 kHz to 3 GHz for Option 003, and 30 kHz to 6 GHz for Option 006). For power sweeps: −15 to 20 dBm in range 0, 25 dB maximum in other ranges. 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 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
MENU (stimulus) (continued)						
Sweep type	Linear	LINFREQ	1	0,1		
	Log	LOGFREQ	1	0,1		
	List	LISFREQ	1	0,1		
	Select a segment	SSEG[D]	3	0,1		1 to 30
	Select all segments	ASEG	1	0,1		
	Power	POWS	1	0,1		
	CW time	CWTIME	1	0,1		
Edit list	Begin	EDITLIST	1			
	Add segment	SADD	1			
	Edit segment N	SEDI[D]	3	D		1 to 30
	Done with segment	SDON	1			
	Delete segment	SDEL	1			
	Done	EDITDONE	1			
	Clear list	CLEL	1			
Edit segment	Start	STAR[D]	3	D		Stimulus range†
	Stop	STOP[D]	3	D		Stimulus range†
	Center	CENT[D]	3	D		Stimulus range†
	Span	SPAN[D]	3	D		Stimulus range†
	Points	POIN[D]	3	D		1 to 1632
	Stepsize	STPSIZE[D]	3	D		Stimulus range†
	CW	CWFREQ[D]	3	D		Stimulus range†
MARKER						
Select active	1 to 5	MARK<I>[D]	3	D		Stimulus range†
	All off	MARKOFF	1	0,1		
Marker zero	Zero offsets	MARKZERO	1			
Delta reference	1 to 5	DELR<I>	2	0,1		1 to 5
	Fixed marker	DELRFIXM	1	0,1		
	Mode off	DELO	1	0,1		
Fixed mkr position	Stimulus	MARKFSTI[D]	3	D		Stimulus range†
	Value	MARKFVAL[D]	3	D		Amplitude range#
	Aux value	MARKFAUV[D]	3	D		Amplitude range#

† For frequency sweeps: 300 kHz to 1.3 GHz. (300 kHz to 3 GHz for Option 003, and 30 kHz to 6 GHz for Option 006). For power sweeps: −15 to 20 dBm in range 0, 25 dB maximum in other ranges. For CW time: 0 to 24 hours. For frequency sweep, transform on: ± 1/frequency step. For CW time sweep, transform on: ±1/time step.

For log mag: ± 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 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
MARKER FCTN						
Marker placement	Continuous	MARKCONT	1	0,1		
	Discrete	MARKDISC	1	0,1		
Coupled	Couple channels	MARKCOUP	1	0,1		
	Uncouple	MARKUNCO	1	0,1		
Displayed	On/off	DISM<ON|OFF>	2	1,0		
Polar markers	Log	POLMLOG	1	0,1		
	Linear	POLMLIN	1	0,1		
	Re/Im	POLMRI	1	0,1		
Smith markers	Linear	SMIMLIN	1	0,1		
	Log	SMIMLOG	1	0,1		
	Re/Im	SMIMRI	1	0,1		
	R+jX	SMIMRX	1	0,1		
	G+jB	SMIMGB	1	0,1		
Set function to marker value	Start	MARKSTAR	1			
	Stop	MARKSTOP	1			
	Center	MARKCENT	1			
	Span	MARKSPAN	1			
	Reference	MARKREF	1			
	Delay	MARKDELA	1			
Search	Off	SEAOFF	1	0,1		
	Maximum	SEAMAX	1	0,1		
	Minimum	SEAMIN	1	0,1		
	Target	SEATARG[D]	3	D		Amplitude range#
	Search left	SEAL	1			
	Search right	SEAR	1			
Width	Value	WIDV[D]	3	D		Amplitude range#
Search	Search on/off	WIDT<ON|OFF>	2	1,0		
Tracking search	On/off	TRACK<ON|OFF>	2	1,0		
Statistics	On/off	MEASTAT<ON|OFF>	2	1,0		

For log mag: ± 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 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
SAVE/RECALL-internal registers						
Save	Selected reg	SAVE<I>	2		OPC	1 to 5
	Selected reg	SAVEREG<I>	2		OPC	01 TO 32
Clear	Selected reg	CLEA<I>	2		OPC	1 to 5
	Selected reg	CLEAREG<I>	2		OPC	01 to 32
	All regs	CLEARALL	1		OPC	
Recall	Selected reg	RECA<I>	2		OPC	1 to 5
	Selected reg	RECAREG<I>	2		OPC	01 to 32
Title	Internal reg	TITR<I>[$]	4			1 to 5, 10 char.
	Internal reg	TITREG<I>[$]	4			01 to 32, 10 char.
	Plot	TITP[$]	4			01 to 32, 10 char.
SAVE/RECALL-disk files						
Purge	Selected file§	PURG<I>	2			1 to 5
Store	To disk§	STOR<I>	2			1 to 5
Title	Disk file	TITF<I>[$]	4			1 to 5, 10 char.
Include with disk files	Data	EXTMDATA<ON|OFF>	2	1,0		
	Raw data	EXTMRAW<ON|OFF>	2	1,0		
	Formatted data	EXTMFORM<ON|OFF>	2	1,0		
	User graphics	EXTMGRAP<ON|OFF>	2	1,0		
	Data only	EXTMDATO<ON|OFF>	2	1,0		
Save format	ASCII/citifile	SAVUASCI	1			
	Binary	SAVUBINA	1			
Load	From disk§	LOAD<I>	2			1 to 5
	File titles§	REFT	1			
Initialize	External disk§	INID	1			
	External disk§	INIE	1			
	LIF Directory size	DIRS[D]	3	D		8 to 8192
Select storage	Internal memory	INTM	1			
	External Disk	EXTD	1			
Disk format	DOS	FORMATDOS	1			
	LIF	FORMATLIF	1			

§ Requires pass control mode.

Table 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
SCALE REF						
Scale	Auto	AUTO	1			
	Value	SCAL[D]	3	D		Amplitude range#
Reference	Position	REFP[D]	3	D		0<D<10
	Value	REFV[D]	3	D		Amplitude range#
	Set to mkr	MARKREF	1			
Delay	Set delay	ELED[D]	3	D		+1.0 s
	Set to mkr	MARKDELA	1			
	Waveguide delay	WAVD	1			
Phase	Offset	PHAO[D]	3	D		+360 deg
SEQ-sequencing						
Sequencing menu	Continue sequence	CONS	1			
	Do sequence	DOSEQ<I>	2			1 to 6
	Gosub sequence	GOSUB	1			
	New/modify sequence	NEWSEQ<I>	2			1 to 6
	Pause to select seq.	PTOS	1			
	Done modify	DONM	1			
	Select sequence	SEQ<I>	2	I		1 to 6
		Q<I>	2	I		1 to 6
TTL I/O	TTL low - end sweep high	TTLHPULS	1			
	TTL high - end sweep low	TTLLPULS	1			
	Programs all TTL output bits	PARAOUT[D]	3			0 to 15
	Set specified bit on TTL	SETBIT[D]	3			0 to 3
	Clear specified bit on TTL	CLEABIT[D]	2			0 to 3
	Input TTL bit high - do SEQ<I>	IFBIHIGH	1			
	Input TTL bit low - do SEQ<I>	IFBILOW	1			
Save/recall sequences	Store to disk§	STORSEQ<I>	2			1 to 6
	Recall from disk§	LOADSEQ<I>	2			1 to 6

For log mag: ± 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.

§ Requires pass control when using the HP-IB port.

Table 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
SEQ-sequencing (continued)						
Special functions	Peripheral address	ADDRPERI[D]	3	D		
	Title to peripheral	TITTPERI	1			
	Wait D seconds	SEQWAIT[D]	3	D		0.1 to 3000 s
	Pause	PAUS	1			
	Marker to CW freq.	MARKCW	1			
	Emit beep	EMIB	1			
	Title to printer	TITTPRIN	1			
	Title to power meter/HP-IB	TITTPMTR	1			
	Show menus	SHOM	1			
	Assert seq. status bit	ASSS	1			
	Read pwr mtr/HP-IB into title string	PMTRTTIT	1			
	Send number into trace memory	TITTMEM	1			
	Duplicate seq. X to seq. Y	DUPLSEQ<X>SEQ<Y>	2			X, Y = 1 to 6
	Print sequence I	PRINSEQ<I>	2			1 to 6
	Begin title sequence	TITSQ	1			
	Title sequence I	TITSEQ<I>	2			1 to 6
	Clear sequence I	CLEASEQ<I>	2			1 to 6
Decision making	If limit test pass then do sequence	IFLTPASSSEQ<I>	2			1 to 6
	If limit test fail then do sequence	IFLTFAILSEQ<I>	2			1 to 6
Loop counter	Set value	LOOC[D]	3			0 to 32,760
	Increment by 1	INCRLOOC				
	Decrement by 1	DECRLOOC				
	If counter equals 0 then do sequence	IFLCEQZESEQ<I>	2			1 to 6
	If counter not eq. 0 then do sequence	IFLCNEZESEQ<I>	2			1 to 6
STIMULUS						
Stimulus	Center	CENT[D]	3	D		Stimulus range†
	Span	SPAN[D]	3	D		Stimulus range†
	Start	STAR[D]	3	D		Stimulus range†
	Stop	STOP[D]	3	D		Stimulus range†

† For frequency sweeps: 300 kHz to 1.3 GHz. (300 kHz to 3 GHz for Option 003, and 30 kHz to 6 GHz for Option 006). For power sweeps: −15 to 20 dBm in range 0, 25 dB maximum in other ranges. 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 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
SYSTEM						
Service	Analog bus	ANAB<ON|OFF>	2	1,0		
Edit power loss table	On/off	PWRLOSS<ON|OFF>	2	1,0		
	Edit list	POWLLIST	1			
	Use sensor A or B	USES<ENSA|ENSB>	2			Sensor B - HP 438A only
	Add segment	SADD	1			
	Edit segment N	SEDI[D]	3	D		1 to 12
	Done with segment	SDON	1			
	Delete segment	SDEL	1			
	Done	EDITDONE	1			
	Clear list	CLEL	1			
Edit power loss segment	Frequency	POWLFREQ[D]	3	D		Stimulus range†
	Value	POWLLOSS[D]	3	D		−9900 to 9900 dB
Edit cal sensor table	Edit sensor menu A	CALFSENA	1			
	Edit sensor menu B	CALFSENB	1			HP 438A only
	Add segment	SADD	1			
	Edit segment N	SEDI[D]	3	D		1 to 12
	Done with segment	SDON	1			
	Delete segment	SDEL	1			
	Done	EDITDONE	1			
	Clear list	CLEL	1			
Edit cal sensor segment	Frequency	CALFFREQ[D]	3	D		Stimulus range†
	Cal factor	CALFCALF[D]	3	D		0 to 200%
SYSTEM-limit testing						
Limit line	On/off	LIMILINE<ON|OFF>	2	1,0		
Limit test	On/off	LIMITEST<ON|OFF>	2	1,0		
	Beeper	BEEPFAIL<ON|OFF>	2	1,0		
Limit offset	Stimulus	LIMISTIO[D]	3	D		Stimulus range†
	Amplitude	LIMIAMPO[D]	3	D		Amplitude range#
	Marker to offset	LIMIMAOF	1			

† For frequency sweeps: 300 kHz to 1.3 GHz. (300 kHz to 3 GHz for Option 003, and 30 kHz to 6 GHz for Option 006). For power sweeps: −15 to 20 dBm in range 0, 25 dB maximum in other ranges. For CW time: 0 to 24 hours. For frequency sweep, transform on: ± 1/frequency step. For CW time sweep, transform on: ±1/time step.

For log mag: ± 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 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
SYSTEM-limit testing (continued)						
Edit table	Begin edit	EDITLIML	1			
	Add segment	SADD	1			
	Edit segment D	SEDI[D]	3	D		1 to 18
	Segment done	SDON	1			
	Delete segment	SDEL	1			
	Done with edit	EDITDONE	1			
	Clear list	CLEAL	1			
Edit segment	Stimulus value	LIMS[D]	3	D		Stimulus range†
	Marker to stimulus	MARKSTIM	1			
	Upper limit	LIMU[D]	3	D		Amplitude range#
	Lower limit	LIML[D]	3	D		Amplitude range#
	Delta limits	LIMD[D]	3	D		Amplitude range#
	Middle value	LIMM[D]	3	D		Amplitude range#
	Marker to middle	MARKMIDD	1			
	Flat line type	LIMTFL	1	0,1		
	Sloping line type	LIMTSL	1	0,1		
	Single point type	LIMTSP	1	0,1		
SYSTEM-transform						
Transform	On/off	TIMDTRAN<ON|OFF>	2		OPC	
Set freq	Low pass	SETF	1			
Mode	Low pass impulse	LOWPIMPU	1	0,1		
	Low pass step	LOWPSTEP	1	0,1		
	Bandpass	BANDPASS	1	0,1		
	Specify gate menu	SPEG	1			
Window	Maximum	WINDMAXI	1			
	Normal	WINDNORM	1			
	Minimum	WINDMINI	1			
	Any value	WINDOW[D]	3	D	State dependent	
Window shape	Use trace memory	WINDUSEM<ON|OFF>	2	1,0		
Demodulation	Off	DEMOOFF	1	0,1		
	Amplitude	DEMOAMPL	1	0,1		
	Phase	DEMOPHAS	1	0,1		

† For frequency sweeps: 300 kHz to 1.3 GHz. (300 kHz to 3 GHz for Option 003, and 30 kHz to 6 GHz for Option 006). For power sweeps: −15 to 20 dBm in range 0, 25 dB maximum in other ranges. For CW time: 0 to 24 hours. For frequency sweep, transform on: ± 1/frequency step. For CW time sweep, transform on: ±1/time step.

For log mag: ± 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 1-8. Key Select Codes (continued)

Function	Action	Mnemonic	S	?	O	Range
SYSTEM-transform (continued)						
Gate	On/off	GATEO<ON|OFF>	2	1,0	OPC	
	Start	GATESTAR[D]	3	D		Stimulus range†
	Stop	GATESTOP[D]	3	D		Stimulus range†
	Center	GATECENT[D]	3	D		Stimulus range†
	Span	GATESPAN[D]	3	D		Stimulus range†
Gate shape	Maximum	GATSMAXI	1	0,1		
	Wide	GATSWIDE	1	0,1		
	Normal	GATSNORM	1	0,1		
	Minimum	GATSMINI	1	0,1		

† For frequency sweeps: 300 kHz to 1.3 GHz. (300 kHz to 3 GHz for Option 003, and 30 kHz to 6 GHz for Option 006). For power sweeps: −15 to 20 dBm in range 0, 25 dB maximum in other ranges. For CW time: 0 to 24 hours. For frequency sweep, transform on: ± 1/frequency step. For CW time sweep, transform on: ±1/time step.

HP-IB Only Commands

Table 1-9. HP-IB Only Commands

Action	Mnemonic	Syntax	?	Description
		MISCELLANEOUS		
Identity	IDN?	1		Outputs the identification string: "HEWLETT PACKARD, 8752C,0,X.XX", where X.XX is the firmware revision of the instrument.
Key	KEY[D]	1	D	Imitates pressing a key. The data transmitted is the key code, as defined in Figure 1-6. Range for D=1 to 61.
Key code	KOR?	1		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.
Move marker	MARKBUCK[D]	2	D	Moves the marker to the selected point on the trace. On a 201 point sweep, D can range from 0 to 200.
On completion	OPC	1		Reports completion of the last OPC-compatible command received since `OPC;` or `OPC?;` was received.
Plot/print softkeys	PSOFT<ON|OFF>	2	1,0	Includes the menu keys when printing or plotting the screen.
Copy default	DEFLTCPIO	1		Sets up a default state for copy.
Revision	SOFR	1		Displays the software revision on the analyzer.
Learn string	SELL[D]	2	D	Selects the learn string revision to input to and output from the analyzer. The valid parameters are: 0: Defaults to current revision. 101: Revision 8752A 1.01 103: Revision 8752A/B 1.03 510: Revision 8752C 5.10 520: Revision 8752C 5.20 526: Revision 8752C 5.26 530: Revision 8752C 5.30 534: Revision 8752C 5.34 538: Revision 8752C 5.38 540: Revision 8752C 5.40 548: Revision 8752C 5.48 612: Revision 8752C 6.12
Test set	TESS?	1		Always returns a one on the HP 8752C.
External trigger	EXTTHIGH	1		Sets the trigger polarity high.
	EXTTLOW	1		Sets the trigger polarity low.

Table 1-9. HP-IB Only Commands (continued)

Action	Mnemonic	Syntax	?	Description
INPUT				
Data	INPUDATA[D]	3	D	Accepts error-corrected data.
Formatted	INPUFORM[D]	3	D	Accepts formatted data.
Uncorrected	INPURAW1[D]	3	D	Accepts raw data.
Error coefficient	INPUCALC<01|02|03>	2		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.
Cal kit	INPUCALK[D]	3	D	Accepts a cal kit.
Learn string	INPULEAS[D]	3	D	Accepts the learn string. Preceded by SELL if learn string is not current revision.
Power trip	POWT<ON|OFF>	2	1,0	Turning power trip off clears a power trip after an overload condition is detected at one of the input ports.
MENUS				
Averaging	MENUAVG	1		
Calibration	MENUCAL	1		
Copy	MENUCOPY	1		
Display	MENUDISP	1		
Format	MENUFORM	1		
Marker	MENUMARK	1		
Meas	MENUMEAS	1		
Marker function	MENUMRKF	1		
Off	MENU<ON|OFF>	2		
Save Recall	MENURECA	1		
Save Recall	MENUSAVE	1		
Scale	MENUSCAL	1		
Sequence	MENUSEQU	1		
Stimulus	MENUSTIM	1		
System	MENUSYST	1		

Table 1-9. HP-IB Only Commands (continued)

Action	Mnemonic	Syntax	?	Description
OUTPUT				
Active function	OUTPACTI	1		Outputs value of function in active entry area in ASCII format.
Serial number	OUTPSERN	1		Outputs the serial number of the analyzer.
Error coefficient	OUTPCALC<01|02|03>	2		Outputs the selected error coefficient array from the active channel. Each array is the same as a data array. See Table 1-6, for the contents of the arrays.
Interp. cal.	OUTPICAL<01|02|03>	2		Outputs the selected interpolated cal coefficient array.
Cal kit	OUTPCALK	1		Outputs the active cal kit, a less than 1000 byte string in form 1.
Data	OUTPDATA	1		Outputs the error corrected data from the active channel in real/imaginary pairs. See Figure 1-4.
	OUTPDATF	1		Fast data transfer command for OUTPDATA.
Error	OUTPERRO	1		Outputs the oldest error in the error queue. The error number is transmitted, then the error message, in ASCII format.
Formatted	OUTPFORM	1		Outputs the formatted trace data from the active channel in current display units. See Table 1-3 for data transmitted.
	OUTPFORF	1		Fast data transfer command for OUTPFORM. Only the first number of the OUTPFORM data pairs is transferred. See Table 1-3.
Identity	OUTPIDEN	1		Outputs identification string, same as IDN?.
Key code	OUTPKEY	1		Outputs the code of the last key pressed, in ASCII format. See Figure 1-6 for key codes. A −1 is transmitted for a knob turn.
Learn string	OUTPLEAS	1		Outputs the learn string, a less than 3,000 byte string in form 1.
Sequencing	OUTPSEQ<I>	2		Outputs sequence I (I=1 to 6) listing over HP-IB.
Limit failures	OUTPLIMF	1		Outputs the limit results as described under OUTPLIML for only those stimulus points that failed.
Limit list	OUTPLIML	1		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.

Table 1-9. HP-IB Only Commands (continued)

Action	Mnemonic	Syntax	?	Description
OUTPUT (continued)				
Limit marker	OUTPLIMM	1		Outputs the limit test results as described for OUTPLIML at the marker.
Marker	OUTPMARK	1		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 1-3 for the marker values.
Memory	OUTPMEMO	1		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.
	OUTPMEMF	1		Fast data transfer command for OUTPMEMO.
Marker statistics	OUTPMSTA	1		Outputs marker statistics: mean, standard deviation, and peak to peak deviation. ASCII format.
Bandwidth	OUTPMWID	1		Outputs results of bandwidth search: bandwidth, center, and Q. ASCII format.
Bandwidth + loss	OUTPMWIL	1		Same operation as OUTPMWID plus the loss value.
Options	OUTPOPTS	1		Outputs an ASCII string of the options installed.
Print list values	OUTPPRNALL	1		Prints all list values or operating and marker parameters in text mode to HP-IB. Requires pass control mode.
Raw data	OUTPRAW1	1		Outputs uncorrected data array for the active channel.
	OUTPRAF<I>	1		Fast data transfer command for OUTPRAW<I>.
Status byte	OUTPSTAT	1		Outputs the status byte. ASCII format.
Display title	OUTPTITL	1		Outputs the display title. ASCII format.
Max values	OUTPAMAX*	1		Outputs max values for all limit line segments.
Min values	OUTPAMIN*	1		Outputs min values for all limit line segments.
Min/max values	OUTPSEGAM*	1		Outputs limit test min/max all segs. Outputs the segment number, max stimulus, max value, min stimulus, min value for all active segments.†
Min/max value	OUTPSEGM*	1		Outputs limit test min/max for a specified segment. See SELSEG[D].†

* Refer to the "Limit Line and Data Point Special Functions" section in Chapter 2.

† For the definition of a limit segment, see "Example Display of Limit Lines" in the Chapter 2 section titled "Limit Line and Data Point Special Functions."

Table 1-9. HP-IB Only Commands (continued)

Action	Mnemonic	Syntax	?	Description
OUTPUT (continued)				
Data: point	OUTPDATP	1		Outputs trace data indexed by point. (see SELPT[D])
Data: range	OUTPDATR	1		Outputs trace data for range of points. (see SELMINPT[D], SELMAXPT[D])
Limit test: ch1	OUTPLIM1*	1		Outputs status§ of limit test for channel 1.
Limit test: ch2	OUTPLIM2*	1		Outputs status§ of limit test for channel 2.
Limit test status	OUTPSEGAF*	1		Outputs the segment number and its limit test status§ for all active segments.†
Limit test status	OUTPSEGF*	1		Outputs the limit test status§ for a specified segment. See SELSEG[D].†
Fail report	OUTPFAIP*	1		This command is similar to OUTPLIMF except that it reports the number of failures first, followed by the stimulus and trace values for each failed point in the test.
LIMIT LINE AND DATA POINT TEST				
Min/max recording	MINMAX<ON|OFF>*	2	1,0	Enables/disables min/max recording per segment. Min and max values are recorded per limit segment.
Segment	SELSEG[D]*	3	D	Selects segment number for the OUTPSEGF and OUTPSEGM commands to report on. D can range from 1 to 18.†
Last point	SELMAXPT[D]	3	D	Selects the last point number in the range of points that the OUTPDATR command will report. D can range from 0 to the number of points minus 1.
First point	SELMINPT[D]	3	D	Selects the first point number in the range of points that the OUTPDATR command will report. D can range from 0 to the number of points minus 1.
Specify point	SELPT[D]	3	D	Selects point number that the OUTPDATR command will report. D can range from 0 to the number of points minus 1.

* Refer to the "Limit Line and Data Point Special Functions" section in Chapter 2.

§ Values returned for limit test status are: 1 (PASS), 0 (FAIL), −1 (NO_LIMIT)

† For the definition of a limit segment, see "Example Display of Limit Lines" in the Chapter 2 section titled "Limit Line and Data Point Special Functions."

Table 1-9. HP-IB Only Commands (continued)

Action	Mnemonic	Syntax	?	Description
OUTPUT FORMATS				
	FORM1	1		HP 8752 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).
SOFTKEYS				
Press	SOFT[I]	2		Activates softkey I, I=1 to 8.
Label	WRSK<1 TO 8>[$]	4		Writes label (10 char) to indicated softkey.
STATUS REPORTING				
Clear	CLES	1		Clears the status byte.
Query	ESB?	1		Returns event-status register B.
	ESR?	1		Returns the event-status register.
	OUTPSTAT	1		Returns the status byte.
Enable	ESE[D]	3	D	Enables event-status register. (0<D<255)
	ESNB[D]	3	D	Enables event-status register B. (0<D<255)
	SRE[D]	3	D	Enables SRQ. (0<D<255)

Alphabetical Mnemonic Listing

Mnemonic	Description
AB	Measure and display A/B on the active channel.
ADDRCONT[D]	Controller HP-IB address: the address where control is returned after a pass control.
ADDRDISC[D]	Disk HP-IB address.
ADDRPERI[D]	Peripheral HP-IB address.
ADDRPLOT[D]	Plotter HP-IB address.
ADDRPOWM[D]	Power meter HP-IB address.
ADDRPRIN[D]	Printer HP-IB address.
ALTAB	Places the analyzer in the alternate inputs measurement mode, where inputs A and B are measured on alternate sweeps. As opposed to `CHOPAB;`.
ANAB<ON|OFF>	Enables the analog bus for service use.
ANAI	Measure and display the data at the auxiliary input (ANALOG IN).
AR	Measure and display A/R on the active channel.
ASEG	Use all segments for list frequency sweep.
ASSS	Asserts the sequence status bit.
AUTO	Auto scale the active channel.
AVERFACT[D]	Set the averaging factor on the active channel.
AVERO<ON|OFF>	Turns averaging ON and OFF on the active channel.
AVERREST	Restarts the averaging on the active channel.
BACI[D]	Sets the background intensity of the display.
BANDPASS	Select the time domain bandpass mode.

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.

CALFSENA Edit the sensor A calibration factor table.

CALFSENB Edit the sensor B calibration factor table.

These commands begin a calibration sequence:

CALIRAI Response and isolation.

CALIRESP Response.

CALIS111 Reflection 1-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|BICIDIEIFIG>. If there is only one standard in the class, these commands are OPC-compatible.

CLASS11A Reflection 1-port, opens.

CLASS11B Reflection 1-port, shorts.

CLASS11C Reflection 1-port, loads.

These commands (OPC- compatible) 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 (0 to 3) on the test set I/O interconnect.

CLEAL Clears the limit line list. Should be preceded by EDITLIML.

CLEL Clears 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:

CLEASEQ1 Sequence 1.

CLEASEQ2 Sequence 2.

CLEASEQ3 Sequence 3.

CLEASEQ4 Sequence 4.

CLEASEQ5 Sequence 5.

CLEASEQ6 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.
COPYFRRT	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.
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.

D1DIVD2<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.
DEBU<ON|OFF>	Turns the HP-IB debug mode ON and OFF. When ON, the HP 8752C scrolls incoming HP-IB commands across the display.
DECRLOOC	Decrements the sequencing loop counter by 1. NEWSEQ<I> 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:	HP-IB
Plotter HP-IB Address:	5
Printer Type:	THINKJET
Printer HP-IB Address:	1

DEFLPRINT Sets the printer to the following default setup conditions:

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

DELR5 Marker 5.

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.

Plot Data	On	Pen Number	
Plot Mem	On	Data	2
Plot Grat	On	Memory	5
Plot Text	On	Graticule	1
Plot Mkr	On	Text	7
Auto-feed	On	Marker	7
Scale Plot	Full	Line Type	
Plot Speed	Fast	Data	7
		Memory	7

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

DISCUNIT[D] Specifies which disk in a multiple-disk 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 DISPDDM.

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

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.

EDITLIML Begin editing limit table.

EDITLIST Begin editing list frequency table.

ELED[D] Sets the electrical delay offset.

EMIB Sends 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 STORSEQ<I>, TITF<I>, and PURG<I>.

These commands include the indicated information when a register is stored on disk. See Figure 1-4 for data types:

EXTMDATA<ON|OFF> Error corrected data.

EXTMDATO<ON|OFF> Data array only.

EXTMFORM<ON|OFF> Formatted trace data.

EXTMGRAP<ON|OFF> User graphics.

EXTMRAW<ON|OFF> Raw data arrays.

EXTTHIGH Sets the external trigger line high.

EXTTLOW Sets the external trigger line 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 8752C 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.

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 control the time domain gate (available only with Option 010, time domain):

GATECENT[D] Center time.

GATEO<ON|OFF> Gate ON/OFF. OPC-compatible.

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<I>.

HOLD Puts the sweep trigger into hold.

IDN? Outputs the identification string: "HEWLETT PACKARD,8752C,0,X.XX", where X.XX is the firmware revision of the instrument.

IFBIHIGH Tests the input test set I/O interconnect bit. Invokes the sequence specified by SEQ<I> if the TTL input bit is high.

IFBILOW Tests the input test set I/O interconnect bit. Invokes the sequence specified by SEQ<I> if the TTL input bit is low.

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>.

IFLCNEZESEQ<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 external disk. All information on the disk will be destroyed. Requires pass control.

INIE Same as INID.

These commands input individual calibration coefficient arrays. Before sending the array, issue a `CALIXXXX;` command, where XXX specifies the calibration type of the data. Then input the cal arrays. Lastly store the data with `SAVC;`. The instrument goes into hold, displaying uncorrected data: `SING;` completes the process by displaying error corrected data. See Table 1-6, for the contents of the different arrays.

INPUCALC01[D] Array 1.

INPUCALC02[D] Array 2.

INPUCALC03[D] Array 3.

INPUCALK[D] Inputs a cal kit read out with `OUTCALK;` . 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 the current format. The instrument stops sweeping and displays the data.

INPULEAS[D] Inputs a learn string read out by `OUTPLEAS;`.

INPURAW1[D] Inputs a raw data array using the current format. See OUTPRAW for the meaning of the array. The instrument stops sweeping, error corrects the data, then formats and displays the data.

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.

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 1-6 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. Otherwise, it has to be converted to a knob count by clearing the upper 8 bits if bit 14 is not set. The resulting integer is the knob count, either positive or negative, depending on the direction of turn. There are approximately 120 counts per knob turn.

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

LABEFWDT[$] Forward transmission.

LABERESI[$] Response, response and isolation.

LABERESP[$] Response.

LABES11A[$] Reflection (opens).

LABES11B[$] Reflection (shorts).

LABES11C[$] Reflection (loads).

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.

LIMIAMPO[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.

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

These commands edit a limit test segment. The limit table editing is begun with `EDITLIML;`, and a segment is brought up for editing with either `SADD;` or `SEDI N;`. 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.

LIMTFL Make the segment a flat line.

LIMTSL Make the segment a sloping line.

LIMTSP 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.

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<I> must precede to ensure that a sequence is currently being created or modified.

LOWPIMPU 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:

MARK1[D]	Marker 1.
MARK2[D]	Marker 2.
MARK3[D]	Marker 3.
MARK4[D]	Marker 4.
MARK5[D]	Marker 5.
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 1-3 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. Use with caution, as this may give unusual results when setting up an instrument state. Recommend setting up states using MENU<ON> (default) and, when setup is complete, using MENU<OFF>.

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

MENUAVG AVG

MENUCAL CAL

MENUCOPY COPY

MENUDISP DISPLAY

MENUFORM FORMAT

MENUMARK MARKER

MENUMEAS MEAS

MENUMRKF MARKER FCTN

MENURECA SAVE RECALL

MENUSAVE SAVE RECALL

MENUSEQU SEQ

MENUSCAL SCALE

MENUSTIM STIMULUS MENU

MENUSYST SYSTEM

MINF[D] Sets the minimum valid frequency of a standard being defined during a cal kit modification.

MINMAX<ON|OFF> Enables/disables min/max recording per segment. Min and max values are recorded per limit segment. Limit testing need not be active.

MINU Displays data minus memory, the same as DISPDMM.

MODI1 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.

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.

OPC Operation complete. Reports the completion of the next command received by setting bit 0 in the event-status register, or by replying to a query if `OPC?;` is issued. See "Command Query" 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 `PLOT;`, 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.

OUTPAMAX Outputs the max values for all limit line segments.

OUTPAMIN Outputs the min values for all limit line segments.

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 1-6, 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.

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 1-4.

OUTPDATF Fast data transfer command for OUTPDATA.

OUTPDATP Outputs the trace data indexed by point (see SELPT[D]).

OUTPDATR Outputs the trace data for range of points (see SELMINPT[D], SELMAXPT[D]).

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.

OUTPFAIP This command is similar to OUTPLIMF except that it reports the number of failures first, followed by the stimulus and trace values for each failed point in the test.

OUTPFORF Fast data transfer command for OUTPFORM.

OUTPFORM Outputs the formatted display data array from the active channel in the current format. See Table 1-3 for the contents of the array positions as a function of display format.

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

OUTPICAL01 Array 1.

OUTPICAL02 Array 2.

OUTPICAL03 Array 3.

OUTPIDEN Outputs the identification string for the analyzer: "HEWLETT PACKARD,8752C,0,X.XX" where X.XX is the firmware revision.

OUTPKEY 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 1-6 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.

OUTPLIM1 Outputs the status of the limit test for channel 1.

OUTPLIM2 Outputs the status of the limit test for channel 2.

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 1-3 for the meaning of the response values as a function of display format.

OUTPMEMF Fast data transfer command for OUTPMEMO.

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 bandwidths 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 Performs the same operation as OUTPMWID plus appends the loss value as well.

OUTPOPTS Outputs an ASCII string of the options installed.

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.

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.

OUTPRAF<I> Fast data transfer command for OUTPRAW<I>.

OUTPRAW1 Outputs the raw measurement data. See Figure 1-4 for the meaning of the data.

OUTPSEGAF Outputs the segment number and it's limit test status for all active segments.

OUTPSEGAM Outputs the limit test min/max for all segments. Outputs the segment number, max stimulus, max value, min stimulus, min value for all active segments.

OUTPSEGF Outputs the limit test status for a specified segment. See SELSEG[D].

OUTPSEGM Outputs limit test min/max for a specified segment. See SELSEG[D].

OUTPSEQ<I> Outputs a sequence listing over HP-IB.

OUTPSERN Outputs the serial number of the analyzer.

OUTPSTAT Outputs the status byte.

OUTPTITL Outputs the display title.

PARAOUT[D] Programs all TTL output bits (0 to 15) 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.
PHAO[D]	Sets the phase offset.
PHAS	Selects the phase display format.
PLOS<SLOW|FAST>	Selects the pen speed for plotting.
PLOT	Initiates a plot. Requires pass control mode.
PLTPRTDISK	Sets the plotter port to disk.
PLTPRTHPIB	Sets the plotter port to HP-IB.
PLTTRAUTF<ON|OFF>	Turns ON and OFF the plotter auto feed.
PLTTRFORF	Sends a form feed to the plotter.
PLTTYPHPGL	Selects HPGL-compatible printer as the plotter type.
PLTTYPPLTR	Selects plotter as the plotter type.
PMEM<ON|OFF>	Selects whether memory is plotted.
PMKR<ON|OFF>	Selects whether markers are plotted.
PMTRTTIT	Reads power meter/HP-IB value into title string. NEWSEQ<I> must precede to ensure that a sequence is currently being created or modified.
POIN[D]	Sets the number of points in the sweep.
POLA	Selects the polar display format.

These commands select the marker readout format for polar display:

POLMLIN	Linear markers.
POLMLOG	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.

PORT1[D]	Reflection. Same as PORTR[D].
PORT2[D]	Transmission. Same as PORTT[D].
PORTR[D]	Reflection.
PORTT[D]	Transmission.
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 use with a power meter or with some source tests.

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. This is an HP-IB only command.
PRAN0	Selects power range 0 when in manual power range. Used with PWRR and PMAN. (Option 004 only.)
PRAN1	Selects power range 1 when in manual power range. Used with PWRR and PMAN. (Option 004 only.)
PRAN2	Selects power range 2 when in manual power range. Used with PWRR and PMAN. (Option 004 only.)
PRAN3	Selects power range 3 when in manual power range. Used with PWRR and PMAN. (Option 004 only.)
PRAN4	Selects power range 4 when in manual power range. Used with PWRR and PMAN. (Option 004 only.)
PRAN5	Selects power range 5 when in manual power range. Used with PWRR and PMAN. (Option 004 only.)
PRAN6	Selects power range 6 when in manual power range. Used with PWRR and PMAN. (Option 004 only.)
PRAN7	Selects power range 7 when in manual power range. Used with PWRR and PMAN. (Option 004 only.)
PRES	Presets the analyzer to the factory preset state. OPC-compatible.
PRIC	Selects color print.
PRINALL	Copies the display, in raster graphics mode, to a printer. Requires pass control mode.
PRINSEQ<I>	Begins printing the sequence selected.
PRINTALL	Prints all list values or operating and marker parameters in ASCII text mode. Requires pass control mode.
PRIS	Selects standard (monochrome) print.
PRNTRAUTF<ON|OFF>	Turns ON and OFF the printer auto feed.
PRNTRFORF	Sends a form feed to the printer.
PRNTYP540	Select the DeskJet 540C printer as the printer type.
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.
PRNTYPPJ	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;`, `PRINALL;`, `OUTPPLOT;` or `OUTPPRIN;`.

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.

PWRLOSS<ON|OFF> Selects whether or not to use the power loss table for use in service tests.

PWRR<PAUTO|PMAN> Select whether the source power range is in auto or manual mode. (Option 004 only.)

Q<I> Same as SEQ<I>.

RAID Completes the response and isolation cal sequence. OPC-compatible.

RAIISOL Calls the isolation class for the response and isolation calibration.

RAIRESP Calls the response class for the response and isolation calibration.

REAL Selects the real display format.

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<I> Recalls save/recall registers 01 through 31. RECAREG01 through RECAREG05 are the same as RECA1 through RECA5. OPC-compatible.

RECO Recalls previously saved display colors.

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.

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.

RFLP Selects the reflection port to measure the reflection response of the device under test.

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 2 commands select the S-parameter for the active channel:

S11 Same as RFLP.

S21 Same as TRAP.

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

SAV1 Completes the 1-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. OPC-compatible.

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

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

SAVUBINA 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.

SEDI[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.

101: Revision 8752A 1.01

103: Revision 8752A/B 1.03

510: Revision 8752C 5.10

520: Revision 8752C 5.20

526: Revision 8752C 5.26

530: Revision 8752C 5.30

534: Revision 8752C 5.34

538: Revision 8752C 5.38

540: Revision 8752C 5.40

548: Revision 8752C 5.48

612: Revision 8752C 6.12

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

SELMAXPT[D] Selects the last point number in the range of points that the OUTPDATR command will report. D can range from 0 to the number of points minus 1.

SELMINPT[D] Selects the first point number in the range of points that the OUTPDATR command will report. D can range from 0 to the number of points minus 1.

SELPT[D] Selects the point number that the OUTPDATR command will report. D can range from 0 to the number of points minus 1.

SELSEG[D] Selects the segment number to report on for the OUTPSEGF and OUTPSEGM commands. D can range from 1 to 18.

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 3) on the TTL output.
SETF	Sets the frequency for low pass transform, Option 010.
SETZ	Sets 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. OPC-compatible.
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. When the class is full, send `CLAD;` to terminate the sequence.

SPECFWDT Forward transmission.

SPECRESP Response.

SPECRESI Resp & Isol, response.

SPECS11A RFL: (opens).

SPECS11B RFL: (shorts).

SPECS11C RFL: (loads).

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, reflection 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:

STDTARBI Arbitrary impedance.

STDTDELA Delay/thru.

STDTLOAD Load.

STDTOPEN Open.

STDTSHOR 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.

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.

SWEA Automatically selects the fastest sweep time based on the current analyzer settings for number of points, IF bandwidth, sweep mode, averaging condition and frequency span.

SWET[D] Sets the sweep time.

SWR Selects the SWR display format.

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? Query test set. Always returns a one on the HP 8752C.

TIMDTRAN<ON|OFF> Turns the Option 010 (time domain) transform ON and OFF. OPC-compatible.

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

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 50 characters (alphanumeric and mathematical symbols) are allowed.

TITP[$] Titles a plot when the plotter type is disk. Should be followed by a string containing at least one and up to ten alphanumeric characters. When a PLOT command is issued, then the filename will be appended with the suffix associated with the plot layout:

FP = full page
LU = left upper quadrant
LL = left lower quadrant
RU = right upper quadrant
RL - right lower quadrant

These commands title the indicated internal register:

TITR1[$] Register 1.

TITR2[$] Register 2.

TITR3[$] Register 3.

TITR4[$] Register 4.

TITR5[$] Register 5.

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

TITSEQ<I> Selects the sequence to be titled.

TITSQ Provides access to the sequence title functions.

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

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

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

TITTPRIN Sends the title string to the printer address. NEWSEQ<I> must precede to ensure that a sequence is currently being created or modified.

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

TRAP Selects the transmission port to measure the transmission response of the device under test.

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

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

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

TTLHPULS TTL normally low, high pulse at end of sweep.

TTLLPULS TTL normally high, low pulse at end of sweep.

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.

WAIT Waits for a clean sweep. OPC-compatible.

WAVD Selects waveguide electrical delay. (See also COAD.)

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

$, 1-31

A

A/B, 1-55
AB, 1-55
abort message (IFC), 1-12
ADDRCONT[D], 1-55
ADDRDISC[D], 1-55
address
  controller, 1-55
  disk drive, 1-55
  peripheral, 1-55
  plotter, 1-55
  power meter, 1-55
  printer, 1-55
address capability, 1-8
addresses for HP-IB, 1-12
ADDRPERI[D], 1-55
ADDRPLOT[D], 1-55
ADDRPOWM[D], 1-55
ADDRPRIN[D], 1-55
adjust brightness, 1-56
adjust color, 1-58
adjust tint, 1-78
AH1 (full-acceptor handshake), 1-9
ALTAB, 1-55
alternate inputs, 1-55
ANAB, 1-55
ANAI, 1-55
analog bus, 1-55
analog input, 1-55
analyzer array-data formats, 1-18
analyzer bus mode, 1-11
analyzer command syntax, 1-3
analyzer control of peripherals, 1-11
analyzer data reading, 1-15
analyzer identification, 1-15
analyzer operation, 1-14
analyzer single bus concept, 1-10
analyzer status reporting structure, 1-24
appendage in syntax, 1-4
AR, 1-55
array-data formats, 1-17
arrays of data, 1-21
arrays related to frequency, 1-19
ASCII
  save format, 1-74
ASEG, 1-55
assert sequence, 1-55
ASSS, 1-55
ATN (attention) control line, 1-7
attention (ATN) control line, 1-7
AUTO, 1-55
auto feed
  plotter, 1-71
  printer, 1-72
auto scale, 1-55
averaging, 1-55
  restart, 1-55
averaging factor, 1-55
AVERFACT[D], 1-55
AVERO, 1-55
AVERREST, 1-55
AVG HP-IB commands, 1-32

B

BACI[D], 1-55
background intensity, 1-55
BANDPASS, 1-55
basic talker (T6), 1-9
beep
  emit, 1-60
BEEPDONE, 1-55
beeper on done, 1-55
beeper on warning, 1-55
BEEPFAIL, 1-55
BEEPWARN, 1-55
begin cal sequence, 1-56
bi-directional lines, 1-7
binary
  save format, 1-74
BR, 1-55
bus device modes, 1-10
bus structure, 1-6, 1-7

C

C10 (pass control capabilities), 1-9
C1,C2,C3 (system controller capabilities), 1-9
C1[D], 1-56
C2[D], 1-56

C3[D], 1-56
CAL1, 1-56
CALFCALF[D], 1-56
CALFFREQ[D], 1-56
CALFSENA, 1-56
CALFSENB, 1-56
CAL HP-IB commands, 1-32
calibration arrays, 1-28
calibration coefficients, 1-21
calibration command sequence, 1-28
calibration kit HP-IB commands, 1-33
calibration kits, 1-56
calibration kit string and learn string, 1-23
calibration type off, 1-56
CALIRAI, 1-56
CALIRESP, 1-56
CALIS111, 1-56
CALK35MD, 1-56
CALK35MM, 1-56
CALK7MM, 1-56
cal kit done, 1-63
CALKN50, 1-56
CALKN75, 1-56
CALKUSED, 1-56
CALN, 1-56
cal sensor table
  edit, 1-56
cal sequence
  begin, 1-56
  resume, 1-73
CBRI[D], 1-56
CENT[D], 1-56
center, 1-56
chain for data processing, 1-21
CHAN1, 1-56
CHAN2, 1-56
CHANNEL HP-IB commands, 1-34
channels
  coupled, 1-58
characters that are valid, 1-4
CHOPAB, 1-56
citifile
  save format, 1-74
CLAD, 1-56
CLASS11A, 1-57
CLASS11B, 1-57
CLASS11C, 1-57
class done, 1-56
CLEABIT[D], 1-57
CLEA<I>, 1-57
CLEAL, 1-57
CLEARALL, 1-57
clear device, 1-12
CLEAREG<I>, 1-57
clear list, 1-57
clear register, 1-57
clear sequence, 1-57
CLEASEQ<I>, 1-57
CLEL, 1-57
CLES, 1-54, 1-57
CLS, 1-57
COAX, 1-57
CO[D], 1-56
code naming conventions, 1-3
code syntax structure, 1-4
COLOCH1D, 1-58
COLOCH1M, 1-58
COLOCH2D, 1-58
COLOCH2M, 1-58
COLOGRAT, 1-58
color
  data channel 1, 1-70
  data channel 2, 1-70
  graticule, 1-70
  memory channel 1, 1-70
  memory channel 2, 1-70
  text, 1-70
  warning, 1-70
COLOR[D], 1-58
colors, 1-70
COLOTEXT, 1-58
COLOWARN, 1-58
? command, 1-15
command formats, 1-4
command query, 1-15
commands
  HP-IB, 1-1
command syntax, 1-3
command syntax structure, 1-4
complete operation, 1-14
complete service request capabilities (SR1), 1-9
computer controllers, 1-6
CONS, 1-58
CONT, 1-58
continue sequence, 1-58
controller
  address, 1-55
controller interface function, 1-6
control lines, 1-7
CONV1DS, 1-58
conventions for code naming, 1-3
CONVOFF, 1-58
CONVREF, 1-58
CONVYTRA, 1-58
CONVZTRA, 1-58
copy display, 1-69, 1-71, 1-72
COPYFRFT, 1-58
COPYFRRT, 1-58
COPY HP-IB commands, 1-35

CORI, 1-58
CORR, 1-58
correction, 1-58
  interpolative, 1-58
COUC, 1-58
COUP, 1-58
coupled channels, 1-58
CRT focus, 1-61
CRT intensity, 1-63
CRT title, 1-79
CW freq, 1-58
CWFREQ[D], 1-58
CW time, 1-58
CWTIME, 1-58

D

[D], 1-31
D1DIVD2, 1-58
data
  include with disk files, 1-61
data-array formats, 1-17
data arrays, 1-21
data bus, 1-7
data channel 1
  color, 1-70
data channel 2
  color, 1-70
data for markers, 1-16
data levels, 1-22
data only
  include with disk files, 1-61
data-processing chain, 1-21
data rate, 1-8
data reading, 1-15
data transfer, 1-7
data-transfer character definitions, 1-16
Data Transfer Commands
  Fast, 1-22
data transfer for traces, 1-19
data units, 1-4
DATI, 1-58
DC1 (complete device clear), 1-9
DEBU, 1-58
debug, 1-58
decrement loop counter, 1-58
DECRLOOC, 1-58
default calibration kits, 1-56
default colors, 1-58
DEFC, 1-58
definitions of status bit, 1-24
DEFLPRINT, 1-59
DEFLTCPIO, 1-49, 1-59
DEFS[D], 1-59
DELA, 1-59
delay, 1-59, 1-60
  set to mkr, 1-66
delete segment, 1-74
DEL&<I>, 1-59
DELO, 1-59
DELRFIXM, 1-59
delta limits, 1-64
delta reference, 1-59
DEMOAMPL, 1-59
demodulation off, 1-59
DEMOOFF, 1-59
DEMOPHAS, 1-59
DeskJet, 1-72
device clear, 1-12
device clear (DC1), 1-9
device trigger, 1-13
device types for HP-IB, 1-6
DFLT, 1-59
directory size
  LIF, 1-60
DIRS[D], 1-60
disabling the front panel, 1-13
DISCUNIT[D], 1-60
DISCVOLU[D], 1-60
disk
  load file, 1-65
disk drive
  address, 1-55
disk drive unit, 1-60
disk drive volume, 1-60
disk file names, 1-29
disk files HP-IB commands, 1-43
disk format, 1-62
DISM, 1-60
DISPDATA, 1-60
DISPDATM, 1-60
DISPDDM, 1-60
DISPDMM, 1-60
display A/B, 1-55
display A/R, 1-55
display B/R, 1-55
display data, 1-60
display data — mem, 1-60
display data & mem, 1-60
display data/mem, 1-60
display data to mem, 1-58
display format units, 1-17
DISPLAY HP-IB commands, 1-36
display memory, 1-60
DISPMEMO, 1-60
DIVI, 1-60
does not respond to parallel poll (PPO), 1-9
done
  with class, 1-60
  with isolation, 1-63
DONE, 1-60

done modify sequence, 1-60
DONM, 1-60
DOSEQ<I>, 1-60
do sequence, 1-60
DOS format, 1-62
DOWN, 1-60
DT1 (responds to a group execute trigger), 1-9
DUAC, 1-60
dual channels, 1-60
duplicate sequence, 1-60
DUPLSEQ<X>SEQ<Y>, 1-60

E

E2 (tri-state drivers), 1-9
edit cal sensor table, 1-56
EDITDONE, 1-60
edit limit table, 1-60
EDITLIML, 1-60
EDITLIST, 1-60
edit power loss range, 1-71
edit power loss table, 1-71
edit segment, 1-75
ELED[D], 1-60
EMIB, 1-60
emit beep, 1-60
end or identify, 1-5
end or identify (EOI) control line, 1-7
ENTO, 1-61
ENTRY HP-IB commands, 1-37
entry off, 1-61
EOI, 1-5
EOI (end or identify) control line, 1-7
Epson-P2, 1-72
error-corrected data, 1-21
error correction HP-IB commands, 1-32
error output, 1-27
error reporting, 1-24
ESB?, 1-54, 1-61
ESE[D], 1-54, 1-61
ESNB[D], 1-54, 1-61
ESR?, 1-54, 1-61
event-status register, 1-24, 1-26
EXTD, 1-61
extended listener capabilities (LEO), 1-9
external trigger, 1-61
EXTMDATA, 1-61
EXTMDATO, 1-61
EXTMFORM, 1-61
EXTMGRAP, 1-61
EXTMRAW, 1-61
EXTTHIGH, 1-49, 1-61
EXTTLOW, 1-49, 1-61
EXTTOFF, 1-61
EXTTON, 1-61
EXTTPOIN, 1-61

F

Fast Data Transfer Commands, 1-22
file names
  disk, 1-29
file titles
  recall, 1-73
firmware revision identification, 1-15
FIXE, 1-61
fixed load, 1-61
fixed marker, 1-59
flat line type, 1-64
FOCU[D], 1-61
FORM1, 1-54, 1-61
FORM1 format, 1-18
FORM2, 1-54, 1-61
FORM2 format, 1-18
FORM3, 1-54, 1-61
FORM3 format, 1-18
FORM4, 1-54, 1-61
form 4 data-transfer character string, 1-16
FORM4 format, 1-18
FORM5, 1-54, 1-61
FORM5 format, 1-18
format
  disk, 1-62
format display units, 1-17
FORMATDOS, 1-62
FORMAT HP-IB commands, 1-37
FORMATLIF, 1-62
formats for array-data, 1-17
formats for commands, 1-4
formatted data, 1-21
  include with disk files, 1-61
form feed
  plotter, 1-71
  printer, 1-72
FREO, 1-62
frequency notation, 1-62
frequency-related arrays, 1-19
full-acceptor handshake (AH1), 1-9
full-source handshake (SH1), 1-9
FULP, 1-62

G

GATE, 1-62
GATECENT[D], 1-62
gate center time, 1-62
gate on/off, 1-62
gate shape, 1-62
  maximum, 1-62
  minimum, 1-62
  normal, 1-62
  wide, 1-62

GATESPAN[D], 1-62
gate span time, 1-62
GATESTAR[D], 1-62
gate start time, 1-62
GATESTOP[D], 1-62
gate stop time, 1-62
GATSMAXI, 1-62
GATSMINI, 1-62
GATSNORM, 1-62
GATSWIDE, 1-62
general structure of syntax, 1-4
GOSUB, 1-62
gosub sequence, 1-62
graticule
  color, 1-70
group execute trigger response (DT1), 1-9
guidelines for code naming, 1-3

H

halting all modes and functions, 1-12
handshake lines, 1-7
HOLD, 1-62
HP-IB
  address capability, 1-8
  addresses, 1-12
  bus structure, 1-6, 1-7
  command formats, 1-4
  data rate, 1-8
  device types, 1-6
  message transfer scheme, 1-8
  meta-messages, 1-12
  multiple-controller capability, 1-8
  operation, 1-6
  operational capabilities, 1-9
  requirements, 1-8
  status indicators, 1-10
HP-IB commands, 1-1
HP-IB only commands, 1-49

I

<I>, 1-31
identification
  of analyzer, 1-15
  of firmware revision, 1-15
IDN?, 1-15, 1-49, 1-62
IEEE-488 universal commands, 1-12
IEEE standard codes, formats, protocols information, 1-2
IEEE standard digital interface information, 1-2
IF bandwidth, 1-62
IFBIHIGH, 1-62
IFBILOW, 1-62
IFBW[D], 1-62
IFC (abort message), 1-12
IFC (interface clear) control line, 1-7
IFLCEQZESEQ<I>, 1-62
IFLCNEZESEQ<I>, 1-62
IFLTFAILSEQ<I>, 1-62
IFLTPASSSEQ<I>, 1-62
IMAG, 1-62
imaginary, 1-62
increment loop counter, 1-63
INCRLOOC, 1-63
INID, 1-63
INIE, 1-63
INPUCALC<I>, 1-50
INPUCALC<I>[D], 1-63
INPUCALK[D], 1-50, 1-63
INPUDATA[D], 1-50, 1-63
INPUFORM[D], 1-50, 1-63
INPULEAS[D], 1-50, 1-63
INPURAW<I>, 1-63
INPURAW<I>[D], 1-50
instrument state summary, 1-23
INTE[D], 1-63
intensity
  background, 1-55
interface addresses, 1-12
interface clear (IFC) control line, 1-7
interface functions
  controller, 1-6
  listener, 1-6
  talker, 1-6
interpolative correction, 1-58
INTM, 1-63
ISOD, 1-63

K

key codes, 1-30
KEY[D], 1-49, 1-63
key select codes, 1-31
KITD, 1-63
kit done, 1-63
KOR?, 1-49

L

LABEFWDT[$], 1-64
label cal kit, 1-64
label class, 1-64
label standard, 1-64
LABERESI[$], 1-64
LABERESP[$], 1-64
LABES11A[$], 1-64
LABES11B[$], 1-64
LABES11C[$], 1-64
LABK[$], 1-64
LABS[$], 1-64
LaserJet, 1-72
LE0 (no extended listener capabilities), 1-9

learn string and calibration kit string, 1-23
LEFL, 1-64
LEFU, 1-64
levels of data, 1-22
LIF
  directory size, 1-60
LIF format, 1-62
LIMD[D], 1-64
LIMIAMPO[D], 1-64
LIMILINE, 1-64
LIMIMAOF[D], 1-64
LIMISTIO[D], 1-64
LIMITEST, 1-64
limit line, 1-64
limit line amplitude offset, 1-64
limit line stimulus offset, 1-64
limit table
  edit, 1-60
limit test, 1-64
limit test beeper, 1-55
limit test fail, 1-62
limit testing HP-IB commands, 1-46
limit test pass, 1-62
LIML[D], 1-64
LIMM[D], 1-64
LIMS[D], 1-64
LIMTFL, 1-64
LIMTSL, 1-64
LIMTSP, 1-64
LIMU[D], 1-64
linear sweep, 1-64
line feeds, 1-5
lines for control, 1-7
lines for handshaking, 1-7
line type
  data, 1-64
  memory, 1-64
LINFREQ, 1-64
LINM, 1-64
lin mag, 1-64
LINTDATA[D], 1-64
LINTMEMO[D], 1-64
LISFREQ, 1-64
list
  clear, 1-57
listener interface function, 1-6
listen mode (L), 1-10
list sweep, 1-64
list values, 1-64
  print, 1-72
LISV, 1-64
L (listen mode), 1-10
LOAD<I>, 1-65
LOADSEQ<I>, 1-65
local command (GTL), 1-12
LOCAL HP-IB commands, 1-38
local lockout command (LLO), 1-13
LOGFREQ, 1-65
LOGM, 1-65
log mag, 1-65
log sweep, 1-65
LOOC[D], 1-65
loop counter
  decrement, 1-58
  increment, 1-63
loop counter value, 1-65
lower limit
  segment, 1-64
low pass frequency, 1-76
low pass impulse, 1-65
low pass step, 1-65
LOWPIMPU, 1-65
LOWPSTEP, 1-65

M

MANTRIG, 1-65
MARKBUCK[D], 1-49
MARKCENT, 1-66
MARKCONT, 1-66
MARKCOUP, 1-66
MARKCW, 1-66
MARKDELA, 1-66
MARKDISC, 1-66
marker bandwidth search, 1-80
marker data, 1-16
MARKER FCTN HP-IB commands, 1-42
MARKER HP-IB commands, 1-41
marker parameters
  print, 1-72
marker range, 1-66
markers
  continuous, 1-66
  discrete, 1-66
  displayed, 1-60
markers coupled, 1-66
marker search
  left, 1-75
  maximum, 1-75
  minimum, 1-75
  off, 1-75
  right, 1-75
  target, 1-75
  tracking, 1-79
markers off, 1-66
marker statistics, 1-67
markers uncoupled, 1-66
marker to CW frequency, 1-66
marker to limit offset, 1-64
marker to middle
  segment, 1-66

marker to stimulus
segment, 1-66
marker width, 1-80
marker zero, 1-66
MARKFAUV[D], 1-66
MARKFSTI[D], 1-66
MARKFVAL[D], 1-66
MARK&<I>[D], 1-66
MARKMIDD, 1-66
MARKMINI, 1-66
MARKOFF, 1-66
MARKREF, 1-66
MARKSPAN, 1-66
MARKSTAR, 1-66
MARKSTIM, 1-66
MARKSTOP, 1-66
MARKUNCO, 1-66
MARKZERO, 1-66
MAXF[D], 1-66
MEASA, 1-66
MEASB, 1-67
MEAS HP-IB commands, 1-38
MEASR, 1-67
MEASTAT, 1-67
measurement calibration, 1-28
measurement restart, 1-74
memory channel 1
color, 1-70
memory channel 2
color, 1-70
MENU, 1-67
MENUAVG, 1-50, 1-67
MENUCAL, 1-50, 1-67
MENUCOPY, 1-50, 1-67
MENUDISP, 1-50, 1-67
MENUFORM, 1-50, 1-67
MENUMARK, 1-50, 1-67
MENUMEAS, 1-50, 1-67
MENUMRKF, 1-50, 1-67
MENU<ON|OFF>, 1-50
MENURECA, 1-50, 1-67
MENUSAVE, 1-50, 1-67
MENUSCAL, 1-50, 1-67
MENUSEQU, 1-50, 1-67
MENUSTIM, 1-50, 1-67
MENUSYST, 1-50, 1-67
message transfer scheme, 1-8
meta-messages, 1-12
methods of HP-IB operation, 1-6
middle value
segment, 1-64
MINF[D], 1-67
MINMAX<ON|OFF>, 1-53, 1-67
min/max recording , 1-67
modes
analyzer bus, 1-11
pass-control, 1-11
system-controller, 1-10
talker/listener, 1-11
modes for bus device, 1-10
MODI1, 1-67
modify cal kit, 1-67
modify colors, 1-58
modify sequence, 1-67
multiple-controller capability, 1-8

N

naming conventions, 1-3
NEWSEQ<I>, 1-67
new sequence, 1-67
NEXP, 1-67
next page, 1-67
no extended talker capabilities (TEO), 1-9
NOOP, 1-67
number of HP-IB devices allowed, 1-6
number of listeners allowed, 1-6
NUMG[D], 1-67

O

OFSD[D], 1-67
OFSL[D], 1-67
OFSZ[D], 1-68
OPC, 1-49, 1-68
OPC-compatible commands, 1-14
open capacitance values, 1-56
OPEP, 1-68
operating parameters, 1-68
operational capabilities for HP-IB, 1-9
operation complete, 1-14
operation of analyzer, 1-14
operation of HP-IB, 1-6
OUTPACTI, 1-51
OUTPAMAX, 1-52, 1-68
OUTPAMIN, 1-52, 1-68
OUTPCALC<I>, 1-51, 1-68
OUTPCALK, 1-51, 1-68
OUTPDAPT, 1-53
OUTPDATA, 1-51, 1-68
OUTPDATF, 1-51, 1-68
OUTPDATP, 1-68
OUTPDATR, 1-53, 1-68
OUTPERRO, 1-51, 1-68
OUTPFAIP, 1-53, 1-68
OUTPFORF, 1-51, 1-68
OUTPFORM, 1-51, 1-68
OUTPICAL<I>, 1-51, 1-69
OUTPIDEN, 1-51, 1-69
OUTPKEY, 1-51, 1-69
OUTPLEAS, 1-51, 1-69
OUTPLIM1, 1-53, 1-69

OUTPLIM2, 1-53, 1-69
OUTPLIMF, 1-51, 1-69
OUTPLIML, 1-51, 1-69
OUTPLIMM, 1-52, 1-69
OUTPMARK, 1-52, 1-69
OUTPMEMF, 1-52, 1-69
OUTPMEMO, 1-52, 1-69
OUTPMSTA, 1-52, 1-69
OUTPMWID, 1-52, 1-69
OUTPMWIL, 1-52, 1-69
OUTPOPTS, 1-69
OUTPPLOT, 1-69
OUTPPRIN, 1-69
OUTPPRNALL, 1-52, 1-70
OUTPRAF<I>, 1-52, 1-70
OUTPRAW1, 1-70
OUTPRAW<I>, 1-52
OUTPSEGAF, 1-53, 1-70
OUTPSEGAM, 1-52, 1-70
OUTPSEGF, 1-53, 1-70
OUTPSEGM, 1-70
OUTPSEGM[D], 1-52
OUTPSEQ<I>, 1-51, 1-70
OUTPSERN, 1-51, 1-70
OUTPSTAT, 1-52, 1-54, 1-70
OUTPTITL, 1-52, 1-70
output
  plot string, 1-69
output ch1 status, 1-69
output ch2 status, 1-69
output data by point, 1-68
output data by range, 1-68
output-data command, 1-15
output limit test min/max, 1-70
output limit test status, 1-70
output max values, 1-68
output min values, 1-68
output number of failures, 1-68
output of errors, 1-27
output queue, 1-15
output segment number, 1-70
output syntax, 1-15
outputting trace-related data, 1-16

P

PaintJet, 1-72
parallel poll configure, 1-13
parallel poll non response (PPO), 1-9
PARAOUT[D], 1-70
pass control, 1-80
pass control capabilities (C10), 1-9
pass-control mode, 1-11
pass control mode, 1-13
PAUS, 1-70
pause, 1-70
pause to select sequence, 1-73
PCB[D], 1-70
PCOLDATA1, 1-70
PCOLDATA2, 1-70
PCOLGRAT, 1-70
PCOLMEMO1, 1-70
PCOLMEMO2, 1-70
PCOLTEXT, 1-70
PCOLWARN, 1-70
PDATA, 1-70
PENNDATA[D], 1-70
PENNGRAT[D], 1-70
PENNMARK[D], 1-70
PENNMEMO[D], 1-70
PENNTEXT[D], 1-71
pen number
  data, 1-70
  graticule, 1-70
  markers, 1-70
  memory, 1-70
  text, 1-71
peripheral
  address, 1-55
peripheral addresses, 1-12
PGRAT, 1-71
PHAO[D], 1-71
PHAS, 1-71
phase, 1-71
phase offset, 1-71
PLOS, 1-71
PLOT, 1-71
plot data, 1-70
plot graticule, 1-71
plot markers, 1-71
plot memory, 1-71
plot quadrant, 1-64, 1-74
plot scale, 1-74
plot softkeys, 1-72
plot speed, 1-71
plot string
  output, 1-69
plotter
  address, 1-55
  auto feed, 1-71
  form feed, 1-71
plotter default setup, 1-59
plotter port
  disk, 1-71
  HP-IB, 1-71
plotter type, 1-71
plot text, 1-73
plot to disk
  title, 1-79
PLTPRTDISK, 1-71
PLTPRTHPIB, 1-71

PLTTRAUTF, 1-71
PLTTRFORF, 1-71
PLTTYPHPGL, 1-71
PLTTYPPLTR, 1-71
PMEM, 1-71
PMKR, 1-71
PMTRTTIT, 1-71
POIN[D], 1-71
points
  specify, 1-71
POLA, 1-71
polar, 1-71
polar markers, 1-71
POLMLIN, 1-71
POLMLOG, 1-71
POLMRI, 1-71
PORE, 1-71
PORT1[D], 1-71
PORT2[D], 1-71
port extensions, 1-71
PORTR[D], 1-71
PORTT[D], 1-71
POWE[D], 1-71
power level, 1-71
power loss range
  edit, 1-71
power loss table, 1-73
  edit, 1-71
power meter
  address, 1-55
power meter cal factor, 1-56
power meter into title string, 1-71
power meter type, 1-72
power ranges, 1-72
power slope, 1-76
power sweep, 1-72
power trip, 1-72
POWLFREQ[D], 1-71
POWLLIST, 1-71
POWLLOSS[D], 1-71
POWM, 1-72
POWS, 1-72
POWT, 1-72
POWT<ON|OFF>, 1-50
PPO (does not respond to parallel poll, 1-9
PRAN, 1-72
PRIC, 1-72
PRINALL, 1-72
PRINSEQ<I>, 1-72
PRINTALL, 1-72
print color, 1-72
printer
  address, 1-55
  auto feed, 1-72
  form feed, 1-72
printer default setup, 1-59
print monochrome, 1-72
print sequence, 1-72
print softkeys, 1-72
PRIS, 1-72
PRNTRAUTF, 1-72
PRNTRFORF, 1-72
PRNTYP540, 1-72
PRNTYPDJ, 1-72
PRNTYPEP, 1-72
PRNTYPLJ, 1-72
PRNTYPPJ, 1-72
PRNTYPTJ, 1-72
processing data chain, 1-21
PSOFT, 1-72
PSOFT<ON|OFF>, 1-49
PTEXT, 1-73
PTOS, 1-73
purge file, 1-73
PURG<I>, 1-73
PWRLOSS, 1-73
PWRR, 1-73

Q

Q<I>, 1-73
query command, 1-15
query syntax, 1-5
queue for output, 1-15

R

RAID, 1-73
RAIISOL, 1-73
RAIRESP, 1-73
raw data
  include with disk files, 1-61
raw measured data, 1-21
reading analyzer data, 1-15
REAL, 1-73
RECA<I>, 1-73
recall colors, 1-73
recall register, 1-73
recall sequence, 1-65
RECAREG<I>, 1-73
RECO, 1-73
reference line value, 1-73
reference position, 1-73
  set to mkr, 1-66
reflection, 1-57
REFT, 1-73
REFV[D], 1-73
remote enable (REN) control line, 1-7
remote/local capability (RL1), 1-9
remote mode, 1-13
remote operation (R), 1-10
REN (remote enable) control line, 1-7

reporting of errors, 1-24
reporting on status, 1-24
RESC, 1-73
RESD, 1-73
reset color, 1-74
RESPDONE, 1-74
response cal done, 1-74
REST, 1-74
restart averaging, 1-55
restore display, 1-73
resume cal sequence, 1-73
RFLP, 1-74
RIGL, 1-74
RIGU, 1-74
RL1 (complete remote/local capability), 1-9
R (remote operation), 1-10
RSCO, 1-74
RST, 1-74
rules for code naming, 1-3

S

S11, 1-74
S21, 1-74
SADD, 1-74
SAV1, 1-74
SAVC, 1-74
save cal kit, 1-74
save colors, 1-78
save format, 1-74
SAVE<I>, 1-74
SAVE/RECALL HP-IB commands, 1-43
SAVEREG<I>, 1-74
save register, 1-74
save sequence, 1-78
SAVEUSEK, 1-74
SAVUASCI, 1-74
SAVUBINA, 1-74
SCAL[D], 1-74
scale
  auto, 1-55
SCALE REF HP-IB commands, 1-44
SCAP, 1-74
SDEL, 1-74
SDON, 1-74
SEAL, 1-75
SEAMAX, 1-75
SEAMIN, 1-75
SEAOFF, 1-75
SEAR, 1-75
SEATARG[D], 1-75
SEDI[D], 1-75
segment
  add, 1-74
  delete, 1-74
  edit, 1-75
segment edit done, 1-60
segment select, 1-77
select first point[D], 1-75
select last point[D], 1-75
select point number[D], 1-75
select segment number[D], 1-75
select sequence, 1-73, 1-75
select standard, 1-77
SELL[D], 1-49
SELMAXPT[D], 1-53, 1-75
SELMINPT[D], 1-53, 1-75
SELPT[D], 1-53, 1-75
SELSEG[D], 1-53, 1-75
sensor input selection, 1-80
SEQ HP-IB commands, 1-44
SEQ<I>, 1-75
sequence wait, 1-75
SEQWAIT[D], 1-75
serial poll, 1-13
service request asserted by the analyzer (S), 1-10
service request (SRQ) control line, 1-7
set bandwidth, 1-62
SETBIT[D], 1-76
SETF, 1-76
setting HP-IB addresses, 1-12
SETZ[D], 1-76
SH1 (full-source handshake), 1-9
SHOM, 1-76
show menus, 1-76
SING, 1-76
single bus concept, 1-10
single point type, 1-64
SLID, 1-76
sliding load, 1-76
  done, 1-76
  set, 1-76
SLIL, 1-76
SLIS, 1-76
SLOPE[D], 1-76
sloping line type, 1-64
SLOPO, 1-76
SMIC, 1-76
SMIMGB, 1-76
SMIMLIN, 1-76
SMIMLOG, 1-76
SMIMRI, 1-76
SMIMRX, 1-76
Smith chart, 1-76
Smith markers, 1-76
SMOOAPER[D], 1-76
SMOOO, 1-76
smoothing, 1-76
smoothing aperture, 1-76
SOFR, 1-49, 1-76

SOFT<I>, 1-76
SOFT[I], 1-54
SOUP, 1-76
source power on/off, 1-76
SPAN[D], 1-76
SPECFWDT, 1-77
specify class, 1-77
specify gate menu, 1-77
specify points, 1-71
SPECRESI[I], 1-77
SPECRESP[I], 1-77
SPECS11A[I], 1-77
SPECS11B[I], 1-77
SPECS11C[I], 1-77
SPEG, 1-77
SPLD, 1-77
split display, 1-77
SR1 (complete service request capabilities), 1-9
SRE[D], 1-54
SRQ (service request) control line, 1-7
SSEG[D], 1-77
S (service request asserted by the analyzer), 1-10
STANA, 1-77
STANB, 1-77
STANC, 1-77
STAND, 1-77
standard defined, 1-77
standard definition, 1-59
standard labelling, 1-64
standard offsets, 1-67
standard type, 1-77
STANE, 1-77
STANF, 1-77
STANG, 1-77
STAR[D], 1-77
statistics
  marker, 1-67
status bit definitions, 1-24
status byte, 1-24, 1-26
status indicators, 1-10
status reporting, 1-24
STB?, 1-77
STDD, 1-77
STDTARBI, 1-77
STDTDELA, 1-77
STDTLOAD, 1-77
STDTOPEN, 1-77
STDTSHOR, 1-78
step down, 1-60
step up, 1-80
STIMULUS HP-IB commands, 1-45
stimulus menu HP-IB commands, 1-40
stimulus value
  segment, 1-64
STOP[D], 1-78
storage
  disk, 1-61
  internal memory, 1-63
store to disk, 1-78
STOR<I>, 1-78
STORSEQ<I>, 1-78
STPSIZE[D], 1-78
string for calibration kit, 1-23
structure of command syntax, 1-4
structure of HP-IB bus, 1-7
structure of status reporting, 1-24
SVCO, 1-78
SWEA, 1-78
SWET[D], 1-78
SWR, 1-78
syntax for commands, 1-3
syntax for output, 1-15
syntax structure, 1-4
syntax types, 1-5
system controller capabilities (C1,C2,C3), 1-9
system-controller mode, 1-10, 1-11
SYSTEM HP-IB commands, 1-46

T

T6 (basic talker), 1-9
take-control command, 1-13
talker interface function, 1-6
talker/listener, 1-78
talker/listener mode, 1-11
TALKLIST, 1-78
talk mode (T), 1-10
TE0 (no extended talker capabilities), 1-9
TERI[D], 1-78
terminal impedance, 1-78
terminators, 1-5
TESS?, 1-49, 1-78
text
  color, 1-70
ThinkJet, 1-72
TIMDTRAN, 1-78
time domain bandpass, 1-55
time domain gate, 1-62
time domain HP-IB commands, 1-47
time specify, 1-78
TINT, 1-78
TITF<I>, 1-79
TITL[$], 1-79
title
  CRT, 1-79
  plot to disk, 1-79
title disk file, 1-79
title register, 1-79

title sequence, 1-79
title string to trace memory, 1-79
title to peripheral, 1-79
title to printer, 1-79
TITP[$], 1-79
TITREG<I>, 1-79
TITR<I>, 1-79
TITSEQ<I>, 1-79
TITSQ, 1-79
TITTMEM, 1-79
TITTPERI, 1-79
TITTPRIN, 1-79
trace-data transfers, 1-19
trace memory, 1-21
trace-related data, 1-16
TRACK, 1-79
transfer of data, 1-7
transfers of trace-data, 1-19
transform, 1-78
TRAP, 1-79
TRIG, 1-80
trigger
  continuous, 1-58
  external, 1-61
  hold, 1-62
  number of groups, 1-67
  single, 1-76
trigger device, 1-13
tri-state drivers (E2), 1-9
TST?, 1-80
T (talk mode), 1-10
TTLHPULS, 1-80
TTLLPULS, 1-80
types of syntax, 1-5

U

units, 1-4
units as a function of display format, 1-17
universal commands, 1-12
UP, 1-80
upper limit
  segment, 1-64
USEPASC, 1-80
user-defined cal kits, 1-56
user-defined kit
  save, 1-74
user graphics
  include with disk files, 1-61
USESENSA, 1-80
USESENSB, 1-80
use sensor A, 1-80
use sensor B, 1-80

V

valid characters, 1-4
velocity factor, 1-80
VELOFACT[D], 1-80

W

WAIT, 1-80
waiting-for-group-execute-trigger, 1-13
waiting-for-reverse-get bit, 1-13
warning
  color, 1-70
warning beeper, 1-55
WAVD, 1-80
WAVE, 1-80
WIDT, 1-80
WIDV[D], 1-80
WINDMAXI, 1-80
WINDMINI, 1-80
WINDNORM, 1-80
window
  maximum, 1-80
  minimum, 1-80
  normal, 1-80
  shape, 1-80
  value, 1-80
WINDOW[D], 1-80
WINDUSEM, 1-80
WRSK<I>[$], 1-54, 1-80

Z

Z0, 1-76

Contents

2. HP BASIC Programming Examples

Introduction . . . 2-1
Required Equipment . . . 2-1
Optional Equipment . . . 2-2
System Setup and HP-IB Verification . . . 2-2
HP 8752C Network Analyzer Instrument Control Using BASIC . . . 2-4
Command Structure in BASIC . . . 2-4
Command Query . . . 2-5
Running the Program . . . 2-6
Operation Complete . . . 2-7
Running the Program . . . 2-7
Preparing for Remote (HP-IB) Control . . . 2-7
I/O Paths . . . 2-8
Measurement Process . . . 2-10
Step 1. Setting Up the Instrument . . . 2-10
Step 2. Calibrating the Test Setup . . . 2-10
Step 3. Connecting the Device under Test . . . 2-10
Step 4. Taking the Measurement Data . . . 2-11
Step 5. Post-Processing the Measurement Data . . . 2-11
Step 6. Transferring the Measurement Data . . . 2-11
BASIC Programming Examples . . . 2-12
Program Information . . . 2-13
Analyzer Features Helpful in Developing Programming Routines . . . 2-13
Analyzer-Debug Mode . . . 2-13
User-Controllable Sweep . . . 2-13
Example 1: Measurement Setup . . . 2-14
Example 1A: Setting Parameters . . . 2-14
Running the Program . . . 2-15
Example 1B: Verifying Parameters . . . 2-16
Running the Program . . . 2-17
Example 2: Measurement Calibration . . . 2-18
Calibration kits . . . 2-18
Example 2A: Response Measurement Calibration . . . 2-19
Running the Program . . . 2-20
Example 2B: Reflection 1-Port Measurement Calibration . . . 2-21
Running the Program . . . 2-22
Example 3: Measurement Data Transfer . . . 2-23
Trace-Data Formats and Transfers . . . 2-23
Example 3A: Data Transfer Using Markers . . . 2-24
Running the Program . . . 2-25
Example 3B: Data Transfer Using FORM 4 (ASCII Transfer) . . . 2-26
Running the Program . . . 2-28
Example 3C: Data Transfer Using Floating-Point Numbers . . . 2-29
Running the Program . . . 2-31
Example 3D: Data Transfer Using Frequency-Array Information . . . 2-32
Running the Program . . . 2-34

Example 3E: Data Transfer Using FORM 1, Internal-Binary Format 2-35
Running the Program 2-36
Example 4: Measurement Process Synchronization 2-37
Status Reporting 2-37
Example 4A: Using the Error Queue 2-38
Running the Program 2-39
Example 4B: Generating Interrupts 2-41
Running the Program 2-43
Example 5: Network Analyzer System Setups 2-44
Saving and Recalling Instrument States 2-44
Example 5A: Using the Learn String 2-44
Running the Program 2-45
Example 5B: Reading Calibration Data 2-46
Running the Program 2-48
Example 5C: Saving and Restoring the Analyzer Instrument State 2-49
Running the Program 2-51
Example 6: Limit-Line Testing 2-52
Using List-Frequency Mode 2-52
Example 6A: Setting Up a List-Frequency Sweep 2-52
Running the Program 2-54
Example 6B: Selecting a Single Segment from a Table of Segments 2-55
Running the Program 2-57
Using Limit Lines to Perform PASS/FAIL Tests 2-58
Example 6C: Setting Up Limit Lines 2-58
Running the Program 2-60
Example 6D: Performing PASS/FAIL Tests While Tuning 2-61
Running the Program 2-63
Example 7: Report Generation 2-64
Example 7A1: Operation Using Talker/Listener Mode 2-64
Running the Program 2-65
Example 7A2: Controlling Peripherals Using Pass-Control Mode 2-66
Running the Program 2-68
Example 7B: Plotting to a File and Transferring File Data to a Plotter 2-69
Running the Program 2-71
Utilizing PC-Graphics Applications Using the Plot File 2-71
Example 7C: Reading ASCII Disk Files to the System Controller Disk File . . . 2-72
Running the Program 2-76
Limit Line and Data Point Special Functions 2-77
Overview 2-78
Example Display of Limit Lines 2-80
Limit Segments 2-81
Output Results 2-82
Constants Used Throughout This Document 2-83
Output Limit Test Pass/Fail Status Per Limit Segment 2-84
Output Pass/Fail Status for All Segments 2-85
Example Program of OUTPSEGAF Using BASIC 2-85
Output Minimum and Maximum Point Per Limit Segment 2-87
Output Minimum and Maximum Point For All Segments 2-88
Example Program of OUTPSEGAM Using BASIC 2-89
Output Data Per Point 2-90
Output Data Per Range of Points 2-91
Output Limit Pass/Fail by Channel 2-92

Index

Figures

2-1. The HP 8752C Network Analyzer System with Controller 2-2
2-2. Status Reporting Structure 2-37
2-3. Limit Segments Versus Limit Lines 2-80

Tables

2-1. Additional BASIC 6.2 Programming Information 2-1
2-2. Additional HP-IB Information 2-1
2-3. HP 8752C Network Analyzer Array-Data Formats 2-26
2-4. Limit Line and Data Point Special Functions Commands 2-78
2-5. Limit Segment Table for Figure 2-4 2-81
2-6. Example Output: OUTPSEGAM (min/max of all segments) 2-82
2-7. Pass/Fail/No_Limit Status Constants 2-83
2-8. Min/Max Test Constants 2-83
2-9. Example Output: OUTPSEGAF (pass/fail for all segments) 2-85
2-10. Example Output: OUTPSEGM (min/max per segment) 2-87
2-11. Example Output: OUTPSEGAM (min/max for all segments) 2-88
2-12. Example Output: OUTPDATP (data per point) 2-90
2-13. Example Output: OUTPDATPR (data per range of points) 2-91

2

HP BASIC Programming Examples

Introduction

This is an introduction to the remote operation of the HP 8752C network analyzer using an HP 9000 Series 300 computer. It is a tutorial introduction using BASIC programming examples. The examples used in this chapter are on the "HP 8752C 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 8752C 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 2-1 for a list of manuals supporting the BASIC revision being used. For more information concerning the Hewlett-Packard Interface Bus, see Table 2-2.

Table 2-1. Additional BASIC 6.2 Programming Information

Description	HP Part Number
HP BASIC 6.2 Programming Guide	98616-90010
HP BASIC 6.2 Language Reference (2 Volumes)	98616-90004
Using HP BASIC for Instrument Control, Volume I	82303-90001
Using HP BASIC for Instrument Control, Volume II	82303-90002

Table 2-2. Additional HP-IB Information

Description	HP Part Number
HP BASIC 6.2 Interface Reference	98616-90013
Tutorial Description of the Hewlett-Packard Interface Bus	5021-1927

Required Equipment

To run the examples in this chapter, in addition to the HP 8752C network analyzer, the following equipment is required:

Computer ..... HP 9000 Series 300
BASIC operating system ..... BASIC 6.2
"HP 8752C HP BASIC Programming Examples" disk 08752-10004
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 8752C Programming Examples" disks. Label them "HP 8752C 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 8752C 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).

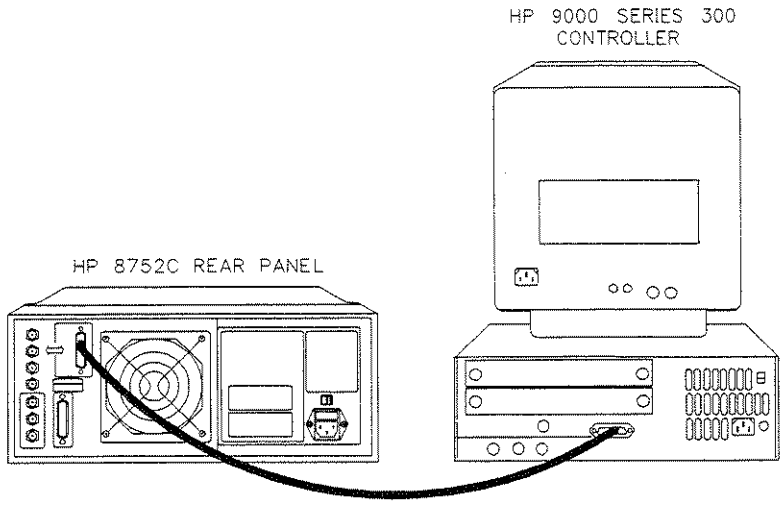


Figure 2-1. The HP 8752C Network Analyzer System with Controller

1. Connect the analyzer to the computer with an HP-IB cable as shown in Figure 2-1.
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:

      (LOCAL) SET ADDRESSES ADDRESS: 8752

      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:

      (1) (6) (x1) (PRESET)

   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 1, "HP-IB Programming and Command Reference." To set the system-control mode, press:

      (LOCAL) TALKER/LISTENER

      or

      (LOCAL) USE PASS CONTROL

5. Check the interface bus by performing a simple command from the computer controller. Type the following command on the controller:

   OUTPUT 716;"PRES;" (EXECUTE) or (RETURN)

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 8752C 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	`OUTPUT` - The BASIC data-output statement.
the appendage	`716` - The data is directed to interface 7 (HP-IB), and on to the device at address 16 (the HP 8752C). This appendage is terminated with a semicolon. The next appendage is `STAR`, the instrument mnemonic for setting the analyzer's start frequency.
data	`10` - a single operand used by the root mnemonic `STAR` to set the value.
unit	`MHZ` - the units that the operand is expressed in.
terminator	`;` - indicates the end of a command, enters the data, and deactivates the active-entry area.

The `"STAR 10 MHZ;"` command performs the same function as pressing the following keys on the analyzer's front panel:

(START) (1) (0) (M/u)

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, "HP-IB Programming and 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;"DUACON;"
```

DUACON 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, switching 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;"DEBUOFF;"
```

Command Query

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-query 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) (x1). 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 1, "HP-IB Programming and Command 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 be also be queried. 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, querying 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 queried.

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.

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 queried 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
```

This halts all bus action and gives active control to the computer.

`20 CLEAR 716`	*This clears all HP-IB errors, resets the HP-IB interface, and clears the syntax errors. It does not affect the status-reporting system.*
`30 OUTPUT 716;"PRES;"`	*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.*
`40 END`	*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 (RETURN).

Type EDIT and press (RETURN).

Type in the following program:

```
10 ASSIGN @Nwa TO 716          Assigns the analyzer to ADDRESS 716.
20 OUTPUT @Nwa;"STAR 10 MHZ;"  Sets the analyzer's start frequency to 10 MHz.
```

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.

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: Response Measurement Calibration" and "Example 2B: Reflection 1-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 1-4 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.

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.

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 8752C HP BASIC Programming Examples" disk received with the analyzer.

- Example 1: Measurement Setup
  - Example 1A: Setting Parameters
  - Example 1B: Verifying Parameters
- Example 2: Measurement Calibration
  - Example 2A: Response Measurement Calibration
  - Example 2B: Reflection 1-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 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 7B: Plotting to a File and Transferring the File Data to a Plotter
    - Utilizing PC-Graphics Applications Using the Plot File
  - Example 7C: Reading ASCII Disk Files to the Instrument Controller's Disk File

Program Information

The following information is provided for every example program included on the "HP 8752C 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 effects 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 `DEBUON;` command and disable the debug mode using the `DEBUOFF;` executable.

User-Controllable Sweep

There are three 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 be 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

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 8752C 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 type of measurement, the display
20  ! format and then sets the specified start and stop frequencies.
30  ! The analyzer display is then autoscaled.
40  !
50  ! EXAMP1A
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;"RFLP;"                   ! Reflection measurement
190 OUTPUT @Nwa;"LOGM;"                   ! Log magnitude display
200 !
210 OUTPUT @Nwa;"CHAN2;"                  ! Channel 2
220 OUTPUT @Nwa;"RFLP;"                   ! Reflection measurement
230 OUTPUT @Nwa;"PHAS;"                   ! Phase display
240 !
250 OUTPUT @Nwa;"DUACON;"                 ! 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 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 8752C Programming Examples" disk received with the network analyzer.

This example shows how to read analyzer settings into your program. Chapter 1, "HP-IB Programming and 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;"            ! Preset the analyzer and wait
140 ENTER @Nwa;Reply                    ! Read in the 1 returned
150 !
160 ! Query network analyzer parameters
170 OUTPUT @Nwa;"POIN?;"                ! Read in the default trace length
180 ENTER @Nwa;Num_points
190 PRINT "Number of points ";Num_points
200 PRINT
210 !
220 OUTPUT @Nwa;"STAR?;"                ! Read in the start frequency
230 ENTER @Nwa;Start_f
240 PRINT "Start Frequency ";Start_f
250 PRINT
260 !
270 OUTPUT @Nwa;"AVERO?;"               ! Averaging on?
280 ENTER @Nwa;Flag
```

```
290 PRINT "Flag =";Flag;"   ";
300 IF Flag=1 THEN                          ! Test flag and print analyzer state
310   PRINT "Averaging ON"
320 ELSE
330   PRINT "Averaging OFF"
340 END IF
350 !
360 OUTPUT @Nwa;"OPC?;WAIT;"                ! Wait for the analyzer to finish
370 ENTER @Nwa;Reply                        ! Read the 1 when complete
380 LOCAL @Nwa                              ! Release HP-IB control
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: 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.

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 a reflection 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 a reflection 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 S_{11} 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: Response Measurement Calibration

Note This program is stored as EXAMP2A on the "HP 8752C Programming Examples" disk received with the network analyzer.

The following program performs a response measurement calibration, using the HP 85032B 50Ω 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 8752C 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 response measurement calibration sequence is run.
- The response measurement 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 response calibration on the HP 8752C.
20  ! It guides the operator through the response calibration by
30  ! connecting a thru connection between the test ports.
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;"TRAP;"                   ! Select transmission measurement
200 OUTPUT @Nwa;"CALKN50;"                ! Select CAL kit type
210 OUTPUT @Nwa;"MENUOFF;"                ! Turn softkey menu off.
220 !
230 OUTPUT @Nwa;"CALIRESP;"               ! Response CAL initiated
240 !
250 CALL Waitforkey("CONNECT THRU BETWEEN PORTS")
260 OUTPUT @Nwa;"OPC?;STANE;"             ! Select the fifth standard, E
```

```
270 ENTER @Nwa;Reply                        ! Read in the 1 returned
280 !
290 OUTPUT 717;"PG;"                        ! Clear the analyzer display
300 !
310 DISP "COMPUTING CALIBRATION COEFFICIENTS"
320 !
330 OUTPUT @Nwa;"OPC?;RESPDONE;"            ! Finished with the CAL cycle
340 ENTER @Nwa;Reply                        ! Read in the 1 returned
350 !
360 DISP "RESPONSE CAL COMPLETED. CONNECT TEST DEVICE."
370 OUTPUT @Nwa;"MENUON;"                   ! Turn on the softkey menu
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 !
430 END
440 !
450 ! ************************** Subroutines ******************************
460 !
470 Waitforkey:  ! Prompt routine to read a keypress on the controller
480 SUB Waitforkey(Lab$)
490   !   Position and display text on the analyzer display
500   OUTPUT 717;"PG;PU;PA390,3700;PD;LB";Lab$;", PRESS ENTER WHEN READY^;"
510   !
520   DISP Lab$&"  Press ENTER when ready"; ! Display prompt on console
530   INPUT A$                              ! Read ENTER key press
540   !
550   OUTPUT 717;"PG;"                      ! Clear analyzer display
560 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 50Ω 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: Reflection 1-Port Measurement Calibration

Note This program is stored as EXAMP2B on the "HP 8752C Programming Examples" disk received with the network analyzer.

The following example program performs a reflection 1-port measurement calibration using the HP 85032B 50Ω type-N calibration kit. 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 reflection 1-port calibration sequence is run.
- 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 8752C.
20  ! It guides the operator through a 1-port reflection 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 ! EXAMP2B
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;"RFLP;"                   ! Select reflection measurement
210 OUTPUT @Nwa;"MENUOFF;"                ! Turn softkey menu off.
220 !
230 OUTPUT @Nwa;"CALIS111;"               ! S11 1 port CAL initiated
240 !
250 CALL Waitforkey("CONNECT OPEN AT RELECTION PORT")
260 OUTPUT @Nwa;"CLASS11A;"               ! Open reflection CAL
270 OUTPUT @Nwa;"OPC?;STANB;"             ! Select the second standard, B
280 ENTER @Nwa;Reply                      ! Read in the 1 returned
290 !
300 CALL Waitforkey("CONNECT SHORT AT REFLECTION PORT")
310 OUTPUT @Nwa;"CLASS11B;"               ! Short reflection CAL
320 OUTPUT @Nwa;"OPC?;STANB;"             ! Select the second standard, B
```

```
330 ENTER @Nwa;Reply                          ! Read in the 1 returned
340 !
350 CALL Waitforkey("CONNECT LOAD AT REFLECTION PORT")
360 OUTPUT @Nwa;"OPC?;CLASS11C;"              ! Reflection load CAL
370 ENTER @Nwa;Reply                          ! Read in the 1 returned
380 !
390 OUTPUT 717;"PG;"                          ! Clear the analyzer display
400 !
410 DISP "COMPUTING CALIBRATION COEFFICIENTS"
420 !
430 OUTPUT @Nwa;"DONE;"                       ! Finished with the CAL cycle
440 OUTPUT @Nwa;"OPC?;SAV1;"                  ! Save the ONE PORT CAL
450 ENTER @Nwa;Reply                          ! Read in the 1 returned
460 !
470 DISP "REFLECTION PORT CAL COMPLETED. CONNECT TEST DEVICE."
480 OUTPUT @Nwa;"MENUON;"                     ! Turn on the softkey menu
490 !
500 OUTPUT @Nwa;"OPC?;WAIT;"                  ! Wait for the analyzer to finish
510 ENTER @Nwa;Reply                          ! Read the 1 when complete
520 LOCAL @Nwa                                ! Release HP-IB control
530 !
540 END
550 !
560 ! ************************** Subroutines ******************************
570 !
580 Waitforkey:  ! Prompt routine to read a keypress on the controller
590 SUB Waitforkey(Lab$)
600   !   Position and display text on the analyzer display
610   OUTPUT 717;"PG;PU;PA390,3700;PD;LB";Lab$;", PRESS ENTER WHEN READYA;"
620   !
630   DISP Lab$&"  Press ENTER when ready"; ! Display prompt on console
640   INPUT A$                              ! Read ENTER key press
650   !
660   OUTPUT 717;"PG;"                      ! Clear analyzer display
670 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 50Ω type-N, PORT 1 being a female test port and PORT 2 being a male test port. The HP 85032B 50Ω 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. 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.

Trace-Data Formats and Transfers

Refer to Table 1-4. 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 in chapter 1, 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 1-4.

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 8752C 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 4, 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 1-4, "HP 8752C 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 IFC (Interface Clear)
120 CLEAR @Nwa                            ! SDC (Selective Device Clear)
130 OUTPUT @Nwa;"OPC?;PRES;"              ! Preset the analyzer and wait
140 ENTER @Nwa;Reply                      ! Read in the 1 returned
150 !
160 OUTPUT @Nwa;"OPC?;SING"               ! Single sweep mode and wait
170 ENTER @Nwa;Reply                      ! Read 1 when sweep complete
180 !
190 OUTPUT @Nwa;"MARK1;"                  ! Turn on marker 1
200 OUTPUT @Nwa;"SEAMAX;"                 ! Find the maximum
210 !
220 OUTPUT @Nwa;"OUTPMARK;"               ! Request the current marker value
230 ENTER @Nwa;Value1,Value2,Stim         ! Read three marker values
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 8752C Programming Examples" disk received with the network analyzer.

Table 2-3. HP 8752C Network Analyzer Array-Data Formats

Format type	Type of Data	Bytes per Data Value	Bytes-per-point 2 data values	201 Bytes-per-trace	Total Bytes add header
FORM 1	Internal Binary	3	6	1206	1210
FORM 2	IEEE 32-bit Floating-Point	4	8	1608	1612
FORM 3	IEEE 64-bit Floating-Point	8	16	3216	3220
FORM 4	ASCII Numbers	24	50	10,050	10,050*
FORM 5	PC-DOS 32-bit Floating-Point	4	8	1206	1220

*No header is used in FORM 4.

The next most common data transfer is to transfer a trace array from the analyzer. 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. 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 2-3 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 querying 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:

$$F = Start\_frequency + (N-1) \times Span/(Points-1)$$

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
40  ! corresponding frequency data is calculated from the analyzer settings.
50  !
60  ! EXAMP3B
70  !
80  ASSIGN @Nwa TO 716                    ! Assign an I/O path to the analyzer
90  !
100 CLEAR SCREEN
110 ! Initialize
120 ABORT 7                               ! Generate an IFC (Interface Clear)
130 CLEAR @Nwa                            ! SDC (Selective Device Clear)
140 OUTPUT @Nwa;"OPC?;PRES;"              ! Preset the analyzer
150 ENTER @Nwa;Reply                      ! Read the 1 when complete
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;"POIN 11;"                ! Set trace length to 11 points
210 OUTPUT @Nwa;"OPC?;SING;"              ! Single sweep mode and wait
220 ENTER @Nwa;Reply                      ! Read reply
230 !
240 OUTPUT @Nwa;"FORM4;"                  ! Set form 4 ASCII format
250 OUTPUT @Nwa;"OUTPFORM;"               ! Send formatted trace to controller
260 ENTER @Nwa;Dat(*)                     ! Read in data array from analyzer
270 !
280 ! Now to calculate the frequency increments between points
290 OUTPUT @Nwa;"POIN?;"                  ! Read number of points in the trace
300 ENTER @Nwa;Num_points
310 OUTPUT @Nwa;"STAR?;"                  ! Read the start frequency
320 ENTER @Nwa;Startf
330 OUTPUT @Nwa;"SPAN?;"                  ! Read the 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.4D,3X,3D.4D   ! 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;"               ! Discrete marker mode
470 OUTPUT @Nwa;"MARK1 3E+4;"            ! Position marker at 30 KHz
```

```
480 !
490 OUTPUT @Nwa;"OPC?;WAIT;"                ! Wait for the analyzer to finish
500 ENTER @Nwa;Reply                        ! Read the 1 when complete
510 LOCAL 7                                 ! Release HP-IB control
520 !
530 PRINT
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

Note This program is stored as EXAMP3C on the "HP 8752C 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                    ! Assign an I/O path for
the analyzer
100 ASSIGN @Nwadat TO 716;FORMAT OFF      ! Binary data path
definition
110 !
120 ABORT 7                               ! Generate an IFC
(Interface Clear)
130 CLEAR @Nwa                            ! SDC (Selected Device
Clear)
140 OUTPUT @Nwa;"OPC?;PRES;"              ! Preset the analyzer and
wait
150 ENTER @Nwa;Reply                      ! Read the 1 when
completed
160 !
170 INTEGER Dheader,Dlength               ! Integer variables for
header info
180 Numpoints=11                          ! Number of points in the
trace
190 OUTPUT @Nwa;"POIN";Numpoints;";"      ! Set number of points in
trace
200 !
210 !  Set up data transfer
220 OUTPUT @Nwa;"OPC?;SING"               ! Single sweep and wait
230 ENTER @Nwa;Reply                      ! Read the 1 when
completed
240 !
250 OUTPUT @Nwa;"FORM3;"                  ! Select form 3 format
260 OUTPUT @Nwa;"OUTPFORM;"               ! Send formatted output
trace
270 !
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.4D,6X,3D.4D
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+4;"           ! Position marker at 30 KHz
450 !
460 OUTPUT @Nwa;"OPC?;WAIT;"             ! Wait for the analyzer to
finish
470 ENTER @Nwa;Reply                     ! Read the 1 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 8752C 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 the 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                               ! Generate an IFC ( Interface Clear)
160 CLEAR @Nwa                            ! SDC (Selective Device Clear)
170 OUTPUT @Nwa;"OPC?;PRES;"              ! Preset the analyzer and wait
180 ENTER @Nwa;Reply                      ! Read the 1 when completed
190 !
200 INTEGER Dheader,Dlength               ! Integer variables for header info
210 !
220 OUTPUT @Nwa;"POIN 11;"                ! Set trace length to 11 points
230 OUTPUT @Nwa;"STAR 10.E+6;"            ! Start frequency 10 MHz
240 OUTPUT @Nwa;"STOP 200.E+6;"           ! Stop frequency 200 MHz
250 OUTPUT @Nwa;"LOGFREQ;"                ! Set log frequency sweep
260 !
270 !  Set up data transfer
280 OUTPUT @Nwa;"OPC?;SING"               ! Single sweep and wait
290 ENTER @Nwa;Reply                      ! Read the 1 when completed
300 !
310 OUTPUT @Nwa;"FORM3;"                  ! Select form 3 trace format
320 OUTPUT @Nwa;"OUTPFORM;"               ! Output formatted trace
330 !
340 ENTER @Nwadat;Dheader,Dlength         ! Read headers from trace data
350 !
360 ALLOCATE Dat(1:Dlength/16,1:2)        ! Use length to determine array size
370 ENTER @Nwadat;Dat(*)                  ! Read in trace data
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;"               ! Request limit line test results
460 ENTER @Nwa;Freq(*)                    ! Read 4 values per point
470 !
480 ! Display table of freq and data
490 !
500 PRINT " Freq (MHz)","Mag (dB)"        ! Print table header
510 FOR I=1 TO 11                         ! Cycle through the trace values
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 8752C 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;"OPC?;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;"OPC?;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;"OUTPDATA;"               ! 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;"OPC?;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

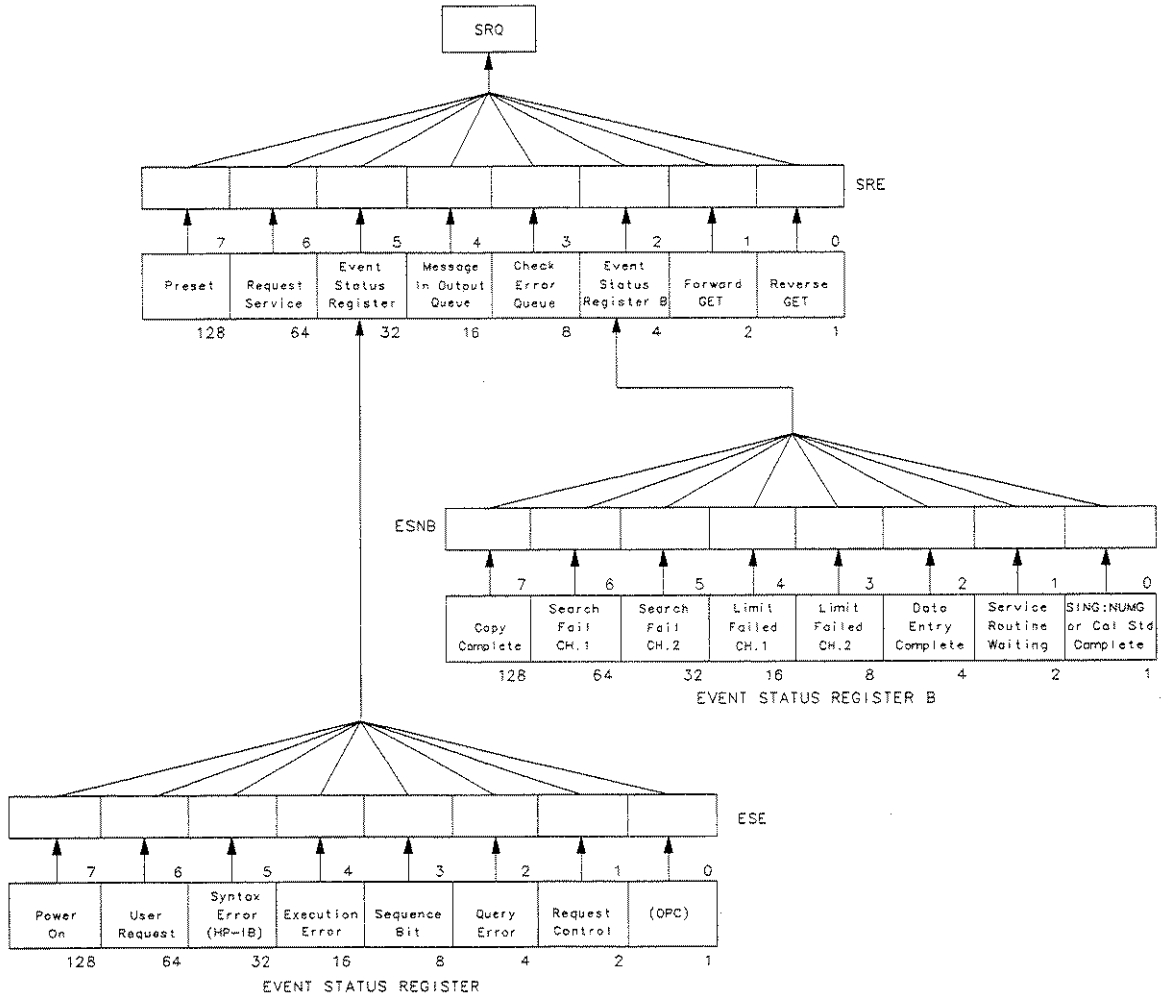


Figure 2-2. Status Reporting Structure

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 1, "HP-IB Programming and Command 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 1 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 8752C 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 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.

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                    ! Assign an I/O path for the analyzer
90  !
100 CLEAR SCREEN
110 ! Initialize the analyzer
120 ABORT 7                               ! Generate an IFC (Interface Clear)
130 CLEAR @Nwa                            ! SDC (Selective Device Clear)
140 OUTPUT @Nwa;"OPC?;PRES;"              ! Preset the analyzer and wait
150 ENTER @Nwa;Reply                      ! Read the 1 when complete
160 !
170 DIM Error$[50]                        ! String for analyzer error message
180 !
190 LOCAL @Nwa                            ! Release analyzer from remote control
200 !
210 LOOP                                  ! Endless loop to read error queue
220   REPEAT
230     Stat=SPOLL(@Nwa)                  ! Read status byte with serial poll
240   UNTIL BIT(Stat,3)                   ! Test for error queue present
250   !
260 !                 Error queue bit is set
270   REPEAT                              ! Loop until error number is 0
280     OUTPUT @Nwa;"OUTPERRO;"           ! Request error queue contents
290     ENTER @Nwa;Err,Error$             ! Read error number and string
300     PRINT Err,Error$                  ! Print error messages
310   UNTIL Err=0                         ! No more errors in queue
320 !
330   LOCAL @Nwa                          ! Release analyzer from remote
340   BEEP 600,.2                         ! Beep to attract attention
350 END LOOP                              ! Repeat error search
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 wait 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 the SYSTEM key

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 Chapter 10 of the *HP 8752C 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 8752C 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, `ESNBn` 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 an error.
- 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, the program prints: `SYNTAX ERROR FROM ANALYZER`.
- If bit 5 in the event-status register is NOT set, the 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 TO 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;"OPC?;PRES;"              ! Preset the analyzer and wait
160 ENTER @Nwa;Reply                      ! Read the one from the analyzer
170 !
180 DIM Error$[50]                        ! String for analyzer error message
190 ! Set up syntax error interrupt
200 OUTPUT @Nwa;"CLES;"                   ! Clear the status registers
210 !
220 ! Generate SRQ when bit 5 is set
230 OUTPUT @Nwa;"ESE 32;"                 ! Event status register bit 5 enabled
240 !
250 ! Generate bit 5 in status register when syntax error occurs
260 OUTPUT @Nwa;"SRE 32;"                 ! 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
310 ENABLE INTR 7;2                       ! Set interrupt on HP-IB bit 2 (SRQ)
320 !
330 DISP "Waiting for bad syntax"
340 WAIT 2                                ! Pause for 2 seconds
350 !
360 OUTPUT @Nwa;"STIP 2GHZ;;"             ! Send bad STOP command syntax
370 !
380 WAIT 2                                ! Pause for 2 seconds
390 DISP ""                               ! Clear display line
400 GOTO Finish                           ! Exit program example
410 !
420 !************************* Subroutines ******************************
430 !
440 Srq_det:                              ! SRQ handler
450 Stat=SPOLL(@Nwa)                      ! Read serial poll byte from HP-IB
460 PRINT "Stat from Serial Poll";Stat
470 IF BIT(Stat,6) THEN                   ! Test for SRQ
480   PRINT "SRQ received from analyzer"
```

```
490 ELSE                                    ! No SRQ from analyzer
500   PRINT "SRQ from other device"
510   STOP                                  ! Stop if not from analyzer
520 END IF
530 !
540 IF BIT(Stat,5) THEN                     ! Event status register bit set
550   PRINT "Event Status Register caused SRQ"
560 ELSE                                    ! Some other bit set
570   PRINT "Some other bit caused the SRQ"
580   STOP                                  ! Stop if bit not set
590 END IF
600   !
610 REPEAT
620   OUTPUT @Nwa;"OUTPERRO;"               ! Read analyzer error queue
630   ENTER @Nwa;Err,Error$                 ! Read error number and string
640   PRINT Err,Error$                      ! Print error message
650 UNTIL Err=0                             ! No more errors in queue
660 !
670 PRINT                                   ! White space
680 ENABLE INTR 7;2                         ! Re-enable SRQ interrupt on HP-IB
690 RETURN
700 !
710 !************************* End Subroutines ******************************
720 !
730 Finish:                                 ! End of program and exit
740 DISP "Finished"
750 OFF INTR 7                              ! Turn off interrupt
760 LOCAL @Nwa                              ! Release HP-IB control
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 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 8752C 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 8752C 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.

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:

```
1   ! This program shows how to retrieve a learn string from the analyzer
2   ! into a string array. The state of the analyzer is then changed and the
3   ! learn string re-loaded to return the analyzer to the previous settings.
4   !
5   ! EXAMP5A
6   !
7 OPTION BASE 1
8 ASSIGN @Nwa TO 716                        ! Assign an I/O path for the analyzer
9 ASSIGN @Nwa_bin TO 716;FORMAT OFF
10  !
11 CLEAR SCREEN
12  ! Initialize the analyzer
13 ABORT 7                                  ! Generate an IFC (Interface Clear)
14 CLEAR @Nwa                               ! SDC (Selected Device Clear)
15  !
16 INTEGER Header,Length                    ! 2-byte header and length of string
17  !
18 OUTPUT @Nwa;"OUTPLEAS;"                  ! Output the learn string
19 ENTER @Nwa_bin;Header,Length             ! Read header and length first
20  !
21 ALLOCATE INTEGER State(Length/2)         ! Integer array to contain the string
22  !
23 ENTER @Nwa_bin;State(*)                  ! Read the string
24 LOCAL @Nwa                               ! Release HP-IB control
25  !
26 INPUT "Change state and press ENTER",A$
27  !
28 OUTPUT @Nwa;"INPULEAS;"                  ! Send the learnstring to analyzer
29 OUTPUT @Nwa_bin;Header,Length,State(*)
30 DISP "Analyzer state has been restored!"
31  !
32 OUTPUT @Nwa;"OPC?;WAIT;"                  ! Wait for the analzyer to finish
33 ENTER @Nwa;Reply                         ! Read the 1 when complete
34 LOCAL @Nwa                               ! Release HP-IB control
35 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 sending the learn string to the analyzer.

Example 5B: Reading Calibration Data

Note This program is stored as EXAMP5B on the "HP 8752C 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 three 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 data formats also apply to the transfer of calibration-coefficient arrays. The Preset State and Memory Allocation chapter in the *HP 8752C 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, or OUTPCALC03. 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  ! EXAMP5B
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 "NOOP",0
210 !
220 INTEGER Hdr,Lgth,I,J                  ! Integers for reading headers
230 !
240 READ Calt$,Numb                       ! Read CAL type from data statement
250 IF Numb=0 THEN GOTO 680               ! If no CAL type is present Exit
260 OUTPUT @Nwa;Calt$;"?;"                ! Query if CAL type is active
270 ENTER @Nwa;Active                     ! Read 1 if active
280 IF NOT Active THEN GOTO 240           ! Load another CAL type and re-try
290 !
300 PRINT Calt$,Numb                      ! Active CAL and number of arrays
310 !
320 OUTPUT @Nwa;"FORM3;"                  ! Form 3 IEEE 64 bit floating point
330 OUTPUT @Nwa;"POIN?;"                  ! Request trace length
340 ENTER @Nwa;Poin                       ! Read number of points
350 ALLOCATE Cal(1:Numb,1:Poin,1:2)       ! Arrays for CAL arrays
360 !         Number of arrays, number of points real and imag value per point
370 !
380 FOR I=1 TO Numb                           ! Read arrays
390   OUTPUT @Nwa USING "K,ZZ";"OUTPCALC",I   ! Format I to add 0 in command
400   ENTER @Nwa_bin;Hdr,Lgth                 ! Read header & length from array
410   FOR J=1 TO Poin                         ! Read elements for CAL array
420     ENTER @Nwa_bin;Cal(I,J,1),Cal(I,J,2)  ! Read real & imag pair elements
430   NEXT J                                  ! Next location in array
440 NEXT I                                    ! Next CAL array
450 !
```

```
460 ! All CAL arrays have been read
470 !
480 INPUT "PRESS RETURN TO RE-TRANSMIT CALIBRATION",Dum$
490 !
500 OUTPUT @Nwa;"FORM3;"                          ! Use same format as read
510 OUTPUT @Nwa;Calt$;";"                         ! Send CAL type to analyzer
520 !
530 FOR I=1 TO Numb                               ! Send each array in CAL
540   DISP "TRANSMITTING ARRAY: ",I               ! Show array number
550   OUTPUT @Nwa USING "K,ZZ";"INPUCALC",I       ! Send array number 0 format
560   OUTPUT @Nwa_bin;Hdr,Lgth                    ! Send header & array length
570   FOR J=1 TO Poin                             ! Send each array element
580     OUTPUT @Nwa_bin;Cal(I,J,1),Cal(I,J,2)     ! Real and Imag pair
590   NEXT J                                      ! Next element in array
600 NEXT I                                        ! Next array
610 !
620 OUTPUT @Nwa;"SAVC;"                           ! Activate CAL
630 !
640 OUTPUT @Nwa;"CONT;"                           ! Restore continuous sweep
650 OUTPUT @Nwa;"OPC?;WAIT;"                      ! Wait for analyzer to finish
660 ENTER @Nwa;Reply                              ! Read the 1 when complete
670 !
680 DISP "Finished with CAL transfer"
690 LOCAL @Nwa                                    ! Release HP-IB control
700 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 8752C 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 the are other internal-memory limitations.

This example demonstrates how to use 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  ! EXAMP5C
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 !
```

```
510 DIM Learn2$[3000]                    ! String for learn string storage
520 ASSIGN @File TO "DATA_FILE:,1406"    ! Open file for reading arrays
530 ENTER @File;Learn2$                  ! Read learn string from file
540 !
550 ENTER @File;Head,Length              ! Read CAL data headers from file
560 Size=Length/16                     ! Array is 2 numbers, 8 bytes per number
570 ALLOCATE Cal_array2(1:Size,1:2)      ! new cal array from file record
580 ENTER @File;Cal_array2(*)            ! Read cal array from disk file
590 !
600 ! Send Learn string back
610 OUTPUT @Nwa;"INPULEAS;",Learn2$      ! Send learn string array
620 !
630 ! Send Cal array back
640 OUTPUT @Nwa;"CALIRESP;"              ! Send CAL type (Response)
650 OUTPUT @Nwa;"INPUCALC01;"            ! Output CAL array to analyzer
660 OUTPUT @Nwa_bin;Head,Length,Cal_array2(*)
670 OUTPUT @Nwa;"OPC?;SAVC;"             ! Save the CAL array
680 ENTER @Nwa;Reply                     ! Read the 1 when complete
690 !
700 OUTPUT @Nwa;";CONT;"                 ! Start the analzyer sweeping
710 OUTPUT @Nwa;"OPC?;WAIT;"             ! Wait for the analyzer to finish
720 ENTER @Nwa;Reply
730 LOCAL @Nwa                           ! Release HP-IB control
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.
- COR 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 8752C 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 8752C 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 1 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  !  EXAMP6A
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                             ! Generate an IFC (Interface Clear)
140 CLEAR @Nwa                          ! SDC (Selective Device Clear)
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   GOSUB Loadpoin
280 NEXT I
290 !
300 LOOP
310   INPUT "DO YOU WANT TO EDIT? Y OR N",An$
320 EXIT IF An$="N"
330   INPUT "ENTRY NUMBER?",I           ! Get an entry number
340   GOSUB Loadpoin                    ! Go load point
350 END LOOP
360 !
370 OUTPUT @Nwa;"EDITLIST"              ! Send the new list to the analyzer
```

```
380 FOR I=1 TO Numb                                  ! Send one segment at a time
390   OUTPUT @Nwa;"SADD;"                            ! Add a segment
400   OUTPUT @Nwa;"STAR";Table(I,1);"MHZ;"           ! Start frequency
410   OUTPUT @Nwa;"STOP";Table(I,2);"MHZ;"           ! Stop frequency
420   OUTPUT @Nwa;"POIN",Table(I,3),";"              ! Number of points
430   OUTPUT @Nwa;"SDON;"                            ! Segment done
440 NEXT I                                  ! Next segment to send to the analyzer
450 !
460 OUTPUT @Nwa;"EDITDONE;"                 ! Done with list
470 OUTPUT @Nwa;"LISFREQ;"                  ! Set list frequency mode
480 !
490 OUTPUT @Nwa;"OPC?;WAIT;"                ! Wait for analyzer to finish
500 ENTER @Nwa;Reply                        ! Read the 1 when complete
510 LOCAL @Nwa                              ! Release HP-IB control
520 STOP                                    ! End of main program
530 !
540 !   **************************Subroutines ******************************
550 !
560 Loadpoin:                               ! Sub to read in each segment value
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 8752C 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 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 [0] 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 TO 716                  !  Assign an I/O path for the analyzer
110 !
120 CLEAR SCREEN
130 ! Initialize the analyzer
140 ABORT 7                             ! Generate an IFC (Interface Clear)
150 CLEAR @Nwa                          ! SDC (Selected Device Clear)
160 !
170 ! Print header for table of existing segments
180 PRINT USING "10A,15A,15A,20A";"SEGMENT","START(MHZ)","STOP(MHZ)","NUMBER OF
```

```
    POINTS"
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;"OUTPACTI;"              ! 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;"OUTPACTI;"              ! 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;"OUTPACTI;"              ! 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 8752C 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 1 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 8752C 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                               ! Generate an IFC (Interface clear)
140 CLEAR @Nwa                            ! SDC (Selected Device Clear)
150 OUTPUT @Nwa;"OPC?;PRES;"              ! Preset the analyzer and wait
160 ENTER @Nwa;Reply                      ! Read the 1 when completed
170 !
180 OUTPUT @Nwa;"EDITLIML;"               ! Edit limit lines
190 OUTPUT @Nwa;"CLEL;"                   ! Clear any existing limits
200 INPUT "NUMBER OF LIMITS?",Numb        ! Request the number of limits
210 ALLOCATE Table(1:Numb,1:3)            ! Create a table
220 ALLOCATE Limtype$(Numb)[2]            ! Create string array of limit types
230 !
240 ! Print out the header for the table
250 PRINT USING "10A,20A,15A,20A";"SEG","STIMULUS (MHz)","UPPER (dB)","
    LOWER (dB)", "TYPE"
260 !
270 ! Prompt the operator to enter the limit values
280 FOR I=1 TO Numb                       ! Cycle through the limits
290   GOSUB Loadlimit                     ! Go read limit values
300 NEXT I                                ! Next limit value
310 !
320 ! Allow the operator to edit the array entered
330 LOOP                                  ! Cycle to edit limit lines
340   INPUT "DO YOU WANT TO EDIT? Y OR N",An$
350 EXIT IF An$="N"                       ! Exit loop on N and send to analyzer
360   INPUT "ENTRY NUMBER?",I             ! Read limit number to edit
370   GOSUB Loadlimit                     ! Go read limit values
380 END LOOP                              ! Next edit entry
390 !
400 ! Send the limit line array segments to the analyzer
410 OUTPUT @Nwa;"EDITLIML;"               ! Edit the limit
420 FOR I=1 TO Numb                       ! Each segment of the limit
430   OUTPUT @Nwa;"SADD;"                   ! Add segment
440   OUTPUT @Nwa;"LIMS";Table(I,1);"MHZ"   ! Send segment stimulus value
450   OUTPUT @Nwa;"LIMU";Table(I,2);"DB"    ! Upper limit value
460   OUTPUT @Nwa;"LIML";Table(I,3);"DB"    ! Lower limit value
470   IF Limtype$(I)="FL" THEN OUTPUT @Nwa;"LIMTFL;" ! Flat limit
480   IF Limtype$(I)="SL" THEN OUTPUT @Nwa;"LIMTSL;" ! Sloping limit
490   IF Limtype$(I)="SP" THEN OUTPUT @Nwa;"LIMTSP;" ! Point limit
500   OUTPUT @Nwa;"SDON;"                 ! Segment done
510 NEXT I                                ! next segment
520 !
```

```
530 OUTPUT @Nwa;"EDITDONE;"                ! Edit complete
540 OUTPUT @Nwa;"LIMILINEON;"              ! Turn limit line on
550 OUTPUT @Nwa;"LIMITESTON;"              ! Turn limit test on
560 !
570 OUTPUT @Nwa;"OPC?;WAIT;"               ! Wait for the analyzer to finish
580 ENTER @Nwa;Reply                       ! Read the 1 when complete
590 !
600 LOCAL @Nwa                             ! Release HP-IB control
610 STOP                                   ! End of main program
620 !
630 !************************ Subroutines *****************************
640 !
650 Loadlimit:                             ! Sub to interact to load data
660 INPUT "STIMULUS VALUE? (MHz)",Table(I,1)  ! and print table created
670 INPUT "UPPER LIMIT VALUE? (DB)",Table(I,2)
680 INPUT "LOWER LIMIT VALUE? (DB)",Table(I,3)
690 INPUT "LIMIT TYPE? (FL=FLAT, SL=SLOPED, SP=SINGLE POINT)",Limtype$(I)
700 !
710                      ! Format and display table values
720 PRINT TABXY(0,I+1);I;TAB(10);Table(I,1);TAB(30);Table(I,2);TAB(45),
    Table(I,3),TAB(67);Limtype$(I)
730 RETURN                                  ! Next limit value
740 !
750 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 8752C 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 1-5 "Status Reporting Structure" and Table 1-3 "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 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.
- 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  ! EXAMP6D
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) 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 1 for more details.

Example 7A1: Operation Using Talker/Listener Mode

Note This program is stored as EXAMP7A1 on the "HP 8752C Programming Examples" disk received with the network analyzer.

The commands OUTPPLOT and OUTPPRIN 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 OUTPPLOT; 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                    ! Assign an I/O path for the analyzer
100 !
110 CLEAR SCREEN
120 ! Initialize analzyer without preset to preserve data
130 ABORT 7                               ! Generate an IFC ( Interface Clear)
140 CLEAR @Nwa                            ! SDC (Selected Device Clear)
150 !
160 OUTPUT @Nwa;"OPC?;SING;"              ! Stop sweep and prepare for plot
170 ENTER @Nwa;Reply                      ! Read in "1" when completed
180 !
190 OUTPUT @Nwa;"OUTPPLOT;"               ! Send plot command
200 SEND 7;UNL LISTEN 5 TALK 16 DATA      ! Unlisten address devices and plot
210 DISP "Plotting and waiting for EOI"
220 WAIT .5                               ! Pause 500 mS to start process
230 !
240 REPEAT                                ! Loop until EOI detected bit is set
250   STATUS 7,7;Stat                     ! Read HP-IB interface register 7
260 UNTIL BIT(Stat,11)                    ! Test bit 11 EOI on HP-IB
270 !
280 End_plot:DISP "End of plot"
290 !
300 OUTPUT @Nwa;"CONT;"                   ! Restore continuous sweep
310 OUTPUT @Nwa;"OPC?;WAIT;"              ! Wait for analyzer to finish
320 ENTER @Nwa;Reply                      ! Read the 1 when complete
330 LOCAL @Nwa                            ! Release remote control
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.

Example 7A2: Controlling Peripherals Using Pass-Control Mode

Note This program is stored as EXAMP7A2 on the "HP 8752C 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 1.

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. 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. Once the printer-dump operation is complete, the analyzer passes control back to the controller and the controller continues programming the analyzer.

Note The analyzer assumes that the address of the computer is correctly stored in its HP-IB addresses menu under (LOCAL) SET ADDRESSES 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                    ! Assign an I/O path for the analyzer
120 !
130 CLEAR SCREEN
140 ! Initialize the analyzer without preset to preserve data
150 ABORT 7                               ! Generate an IFC ( Interface Clear)
160 CLEAR @Nwa                            ! SDC (Selected Device Clear)
170 !
180 OUTPUT @Nwa;"OPC?;SING;"              ! Single sweep and stop for print
190 ENTER @Nwa;Reply                      ! Read in "1" when complete
200 !
210 OUTPUT @Nwa;"CLES;"                   ! Clear status registers
220 OUTPUT @Nwa;"ESE2;"                   ! Enable bit 1 of ESR request control
230 OUTPUT @Nwa;"SRE32;"                  ! Enable ESR interrupt for SRQ
240 !
250 OUTPUT @Nwa;"USEPASC;"                ! Enable pass control mode
260 OUTPUT @Nwa;"PRINALL;"                ! Begin printer dump
270 !
280 REPEAT                                ! Loop until SRQ bit is set
290   Stat=SPOLL(@Nwa)                    ! Read status byte with serial poll
300 UNTIL BIT(Stat,6)                     ! Test for bit 6, SRQ
310 !
320 Pass_control:                         ! SRQ detected. Test for pass control
330 IF BIT(Stat,5) THEN                   ! Requested pass control
340   PASS CONTROL @Nwa                   ! Send take control message
350 ELSE                                  ! Not bit 5, some other event
360   DISP "SRQ but not request pass control"
370   STOP                                ! Halt program
380 END IF
390 !
400 DISP "Printing from analyzer and waiting for control"
```

```
410 !
420 REPEAT                                  ! Loop and wait for completion
430   STATUS 7,6;Hpib                      ! Read HP-IB interface register
440 UNTIL BIT(Hpib,6)                       ! Bit 6 is active controller
450 !
460 DISP "Control returned from analyzer"
470 OUTPUT @Nwa;"TALKLIST;"                 ! Set talker/listener mode again
480 OUTPUT @Nwa;"CLES;"                     ! Clear status byte assignments
490 !
500 OUTPUT @Nwa;"CONT;"                     ! Start analyzer sweeping again
510 OUTPUT @Nwa;"OPC?;WAIT;"                ! Wait for analyzer to finish
520 ENTER @Nwa;Reply                        ! Read the 1 when complete
530 !
540 LOCAL @Nwa                              ! Release HP-IB control
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 7B: Plotting to a File and Transferring File Data to a Plotter

Note This program is stored as EXAMP7B on the "HP 8752C 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 in 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 `OUTPPLOT;` 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 is 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$[32000]                      ! Define string for plotter commands
180 !
190 OUTPUT @Nwa;"OPC?;SING;"              ! Stop sweep for plot and wait
200 ENTER @Nwa;Reply                      ! Read the 1 when complete
210 OUTPUT @Nwa;"OUTPPLOT;"               ! Request plotter output
220 !
230 ENTER @Nwa;Plot$                      ! 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 analzyer to finish
440 ENTER @Nwa;Reply                      ! Read the 1 when complete
450 LOCAL @Nwa                            ! Release HP-IB control
460 END
```

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.

Utilizing PC-Graphics Applications Using the Plot File

You can use this Example 7B 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.

Example 7C: Reading ASCII Disk Files to the System Controller Disk File

Note This program is stored as EXAMP7C on the "HP 8752C 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. An HP CS-80 disk drive is required to create a floppy disk-based file for this example. This floppy disk will be moved from the controller and then accessed to read the data file. 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 HP8752C.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
8.26080E-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 can be stored from the analyzer by remote control or by front-panel operations. See "Saving Instrument States" in the Printing, Plotting, and Saving Measurement Results chapter in the *HP 8752C 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 onto the analyzer's external 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 08752-10004) 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 external disk drive. There is also a program to read the data from the file into a data array for further processing or reformatting to another file type. 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 external 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 drive and the
40  ! data accessed.
50  !
60  ! EXAMP7C
70  !
80  ASSIGN @Nwa TO 716                    ! Assign an I/O path for the analyzer
90  !
100 CLEAR SCREEN
110 ABORT 7                               ! Generate an IFC (Interface Clear)
120 CLEAR @Nwa                            ! SDC (Selected Device Clear)
130 OUTPUT @Nwa;"OPC?;PRES;"              ! Preset the analyzer and wait
140 ENTER @Nwa;Reply                      ! Read the 1 when complete
150 !
160 DIM Record$[80]                       ! String to read the disk records
170 !
180 ! Set up analyzer
190 OUTPUT @Nwa;"STAR100MHZ;"             ! Start frequency 100 MHz
200 OUTPUT @Nwa;"STOP 200MHZ"             ! Stop frequency 200 MHz
210 OUTPUT @Nwa;"POIN11;"                 ! Trace length 11 points
220 OUTPUT @Nwa;"OPC?;SING;"              ! Single sweep and wait
230 ENTER @Nwa;Reply                      ! Read in the 1 when complete
240 !
250 ! Program disk storage operation
260 !
270 OUTPUT @Nwa;"USEPASC;"                ! Enable pass control mode
280 OUTPUT @Nwa;"EXTD;"                   ! Select external disk file
290 GOSUB Pass_control
300 OUTPUT @Nwa;"EXTMFORMON;"             ! Store formated data
310 OUTPUT @Nwa;"EXTMDATOON;"             ! Store data file only
320 INPUT "Enter data file name (5 chars)",File_name$ ! Get file name
330 File_name$=UPC$(File_name$)           ! File names are uppercase
340 OUTPUT @Nwa;"TITF1""";File_name$;""";" ! Title for save reg 1
350 OUTPUT @Nwa;"SAVUASCI;"               ! Save as ASCII file
360 !
370 ! Store data file on external disk
380 !
390 OUTPUT @Nwa;"STOR1;"                  ! Store data to disk file
400 GOSUB Pass_control
410 !
420 ! File storage is complete
430 !
440 INPUT "Place disk in controller disk drive, then press Return",A$
450 !
460 ! Read data file information
470 !
480 ASSIGN @File TO File_name$&"D1:,1400"  ! Open an I/O path for file
490 Record_cnt=1                          ! Counter to count records
500 !
510 PRINT CHR$(12);                       ! Formfeed to clear display
520 PRINT "Contents of data file"         ! Show contents of the data file
530 Readfile:   !
```

```
540 ON END @File GOTO End_file            ! Test for end of file and exit
550 ENTER @File;Record$                   ! Read ASCII record
560 PRINT Record_cnt,Record$              ! print record on display
570 Record_cnt=Record_cnt+1               ! Increment record counter
580 GOTO Readfile                         ! Read next record
590 !
600 End_file: !                           ! Reached the end of file
610 PRINT "End of File. ";Record_cnt-1;" Records found"
620 INPUT "Press Return to continue",A$
630 PRINT CHR$(12);                       ! Formfeed to clear display
640 !
650 ! Read file data into arrays
660 !
670 RESET @File                           ! Rewind file pointer to begining
680 FOR I=1 TO 6
690   ENTER @File;Record$                 ! Skip first six records
700 NEXT I
710 ENTER @File;Record$                   ! Read frequency data record
720 Record$=Record$[POS(Record$," ")+1]   ! skip SEG to first space + 1
730 Startf=VAL(Record$)                   ! Read start frequency
740 Record$=Record$[POS(Record$," ")+1]   ! Skip to next space + 1
750 Stopf=VAL(Record$)                    ! Read stop frequency
760 Record$=Record$[POS(Record$," ")+1]   ! Skip to next space +1
770 Num_points=VAL(Record$)               ! Read the number of points
780 PRINT " Number of points in file ";Num_points
790 PRINT                                 ! White space
800 !
810 Freq_inc=(Stopf-Startf)/(Num_points-1) ! Compute frequency increment
820 !
830 ALLOCATE Array(Num_points,2)          ! Allocate array from Num_points
840 ENTER @File;Record$                   ! Skip SEG_LIST_END record
850 ENTER @File;Record$                   ! Skip BEGIN record
860 !
870 ! Read in the data array
880 PRINT "Freq (MHz)  Data 1    Data 2"  ! Table header for data array
890 FOR I=1 TO Num_points                 ! Read in array entries
900   ENTER @File;Record$                 ! Read in the record of 2 entries
910   !
920   Array(I,1)=VAL(Record$)             ! Read first data value
930   Data$=Record$[POS(Record$,",")+1]   ! Skip to comma and next value
940   Array(I,2)=VAL(Data$)               ! Read second data value
950   !
960   Freq=Startf+(Freq_inc*(I-1))        ! Compute stimulus value for array
970   Freq=Freq/1.E+6                     ! Convert frequency to MHz
980   !
990   PRINT Freq,Array(I,1),Array(I,2)    ! Print data array values
1000 NEXT I                               ! Read next array data points
1010!
1020 OUTPUT @Nwa;"CONT;"                  ! Restore continuous sweep
1030 OUTPUT @Nwa;"OPC?;WAIT;"             ! Wait for analyzer to finish
1040 ENTER @Nwa;Reply                     ! Read the 1 when complete
1050 LOCAL @Nwa                           ! Release HP-IB control
1060 STOP
1070 !********************* Subroutines ****************************
1080 Pass_control: !
```

```
1090 PASS CONTROL @Nwa                    ! Pass control to the analyzer
1100 REPEAT                               ! Loop and read HP-IB status
1110  STATUS 7,6;Hpib                     ! Status from internal HP-IB
1120  ! Reading the controller register does not interfere with the
1130  ! HP-IB disk storage operations taking place. Bit 6 is set when
1140  ! control is returned to the controller.
1150  DISP "Waiting for control"
1160 UNTIL BIT(Hpib,6)                    ! Loop until control is returned
1170 RETURN
1180 !
1190 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.

Limit Line and Data Point Special Functions

The analyzer has special functions in the area of limit testing and in the detection of min/max data points within limit segments. The information in this section will teach you how to use these limit line and data point special functions. The following topics are included:

- Overview
- Constants Used Throughout This Document
- Output Limit Test Pass/Fail Status Per Limit Segment
- Output Pass/Fail Status For All Segments
- Output Minimum and Maximum Point Per Limit Segment
- Output Minimum and Maximum Point For All Segments
- Output Data Per Point
- Output Data Per Range of Points
- Output Limit Pass/Fail by Channel

Overview

The limit line and data point special functions are available as remote commands only. Each command is overviewed in Table 2-4.

Table 2-4. Limit Line and Data Point Special Functions Commands

Action	Mnemonic	Syntax	?	Description
MIN/MAX DATA DETECTION PER LIMIT SEGMENT				
Min/max recording	MINMAX<ON|OFF>	2	1,0	Enables/disables min/max recording per segment. Min and max values are recorded per limit segment.
Max values	OUTPAMAX	1		Outputs max values for all limit line segments. OUTPAMAX values and OUTPAMIN values are both output using OUTPSEGAM.
Min values	OUTPAMIN	1		Outputs min values for all limit line segments. OUTPAMIN values and OUTPAMAX values are both output using OUTPSEGAM.
Min/max values	OUTPSEGAM	1		Outputs limit test min/max for all segs. Outputs the segment number, max stimulus, max value, min stimulus, and min value for all active segments.†
Min/max value	OUTPSEGM	1		Outputs limit test min/max for a specified segment. See SELSEG[D].†
Segment	SELSEG[D]	3	D	Selects segment number for the OUTPSEGF and OUTPSEGM commands to report on. D can range from 1 to 18.†
OUTPUT TRACE DATA BY SELECTED POINTS				
Last point	SELMAXPT[D]	3	D	Selects the last point number in the range of points that the OUTPDATR command will report. D can range from 0 to the number of points minus 1.
First point	SELMINPT[D]	3	D	Selects the first point number in the range of points that the OUTPDATR command will report. D can range from 0 to the number of points minus 1.
Specify point	SELPT[D]	3	D	Selects the single point number that the OUTPDATP command will report. D can range from 0 to the number of points minus 1.
Data: point	OUTPDATP	1		Outputs a single trace data value indexed by point. (see SELPT[D])
Data: range	OUTPDATR	1		Outputs trace data for range of points. (see SELMINPT[D], SELMAXPT[D])

† For the definition of a limit segment, see "Example Display of Limit Lines."

Table 2-4 (cont). Limit Line and Data Point Special Functions Commands

Action	Mnemonic	Syntax	?	Description
LIMIT TEST STATUS BY CHANNEL				
Limit test: ch1	OUTPLIM1	1		Outputs status§ of limit test for channel 1.
Limit test: ch2	OUTPLIM2	1		Outputs status§ of limit test for channel 2.
LIMIT TEST STATUS BY SEGMENT				
Segment	SELSEG[D]	3	D	Selects the segment number for the OUTPSEGF and OUTPSEGM commands to report on. D can range from 1 to 18.†
Limit test status	OUTPSEGAF	1		Outputs the segment number and it's limit test status§ for all active segments.†
Limit test status	OUTPSEGF	1		Outputs the limit test status§ for a specified segment. See SELSEG[D].†
LIMIT TEST STATUS BY POINT				
Fail report	OUTPFAIP	1		This command is similar to OUTPLIMF except that it reports the number of failures first, followed by the stimulus and trace values for each failed point in the test (note: use command LIMITEST<ON> to function properly).

† For the definition of a limit segment, see "Example Display of Limit Lines."
§ Values returned for limit test status are: 1 (PASS), 0 (FAIL), −1 (NO_LIMIT)

Example Display of Limit Lines

The features that output data by limit segment are implemented based on the current definition of a limit segment. The actual limit lines formed by the limit table almost never have a 1-for-1 relationship with the segment numbers in the limit edit table. Out of 18 segments in the limit table, you can create 18 limit lines if (a) all limit segments are contiguous and (b) the last segment extends to the stop frequency. Otherwise, terminating a segment requires a single point which means that constructing a limit line requires two entries (segments) of the limit table. Thus you have a minimum of 9 lines available and those lines will not be referenced by sequential segment numbers.

Figure 2-3 is an example of a screen print of limit lines set up on the two instrument channels. The limit line examples shown are of Flat Line, Slope Line and Single Point Limits. See Table 2-5.

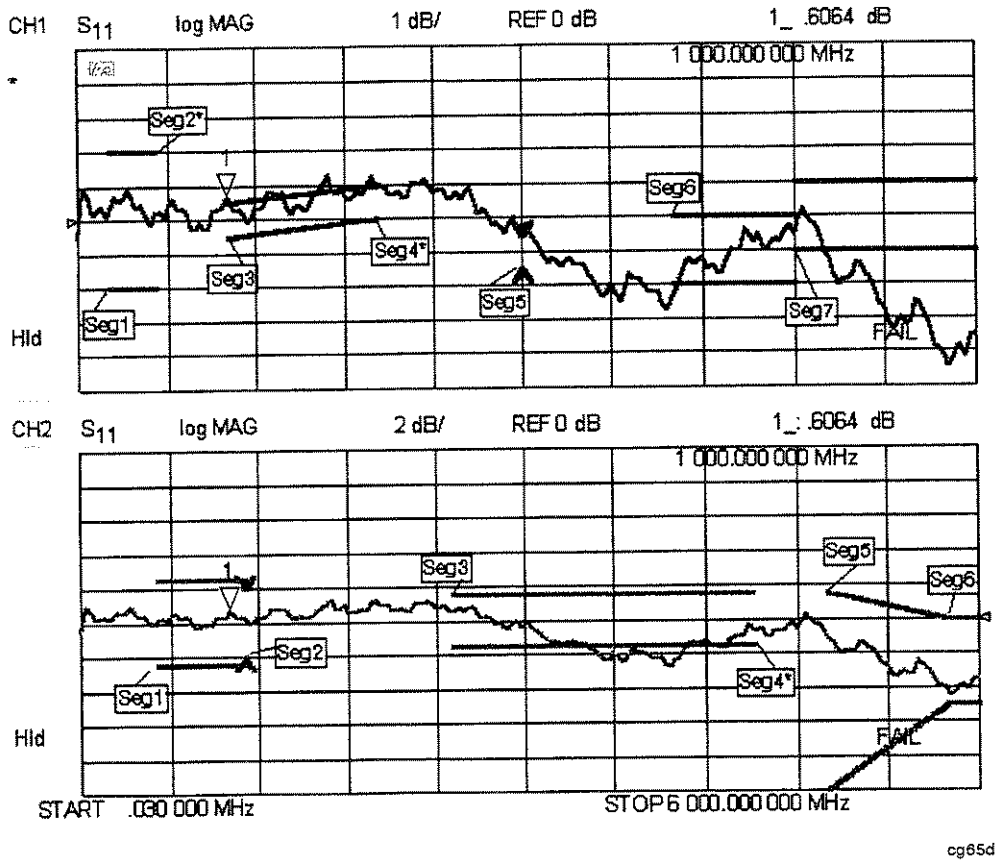


Figure 2-3. Limit Segments Versus Limit Lines

Limit Segments

The values in Table 2-5 were used to create the limit lines in Figure 2-4.

Table 2-5. Limit Segment Table for Figure 2-4

Segment Num.	Stimulus (Frequency)	Upper Limit (dB)	Lower Limit (dB)	Limit Type
Channel 1				
1	200 MHz	2	−2	Flat Line (FL)
2*	500 MHz	2	−2	Single Point (SP)
3	1000 MHz	0.5	−0.5	Slope LIne (SL)
4*	2000 MHz	1	0	Single Point (SP)
5	3000 MHz	−0.5	−1.5	Single Point (SP)
6	4000 MHz	0	−2	Flat Line (FL)
7	4800 MHz	1	−1	Flat Line (FL)
Channel 2				
1	500 MHz	2.5	−2.5	Flat Line (FL)
2	1100 MHz	2	−2	Single Point (SP)
3	2500 MHz	1.5	−1.5	Flat Line (FL)
4*	4500 MHz	1.5	−1.5	Single Point (SP)
5	5000 MHz	1.5	−10	Slope Line (SL)
6	5800 MHz	0	−5	Slope Line (SL)
* No test limit-segment is created.				

Note that if a single point limit is used to terminate slope lines, no test limit-segment is created. (See Figure 2-3: CH1, Seg4.) Also, if a single point limit is used to terminate a flat line, no test limit-segment is created. (See Figure 2-3: CH1, Seg2.) However, if the single point limit used to terminate the flat line limit has different limit values, a single-point test limit-segment is created. (See Figure 2-3: CH2, Seg2.)

Output Results

Table 2-6 shows the output of the OUTPSEGAM test (min/max of all active segments); note that the segments with asterisks (*) from Table 2-5 have no output in Table 2-6.

Table 2-6. Example Output: OUTPSEGAM (min/max of all segments)

Channel 1 Segment	Freq. at Minimum Value (Hz)	Minimum Value (dB)	Freq. at Maximum Value (Hz)	Maximum Value (dB)
1	480027600	−0.1268225	330028350	0.9590923
3	1140024300	−0.09223454	1680021600	1.258809
5	3000015000	−0.2199298	3000015000	−0.2199298
6	4020009900	−2.203248	4770006150	−0.2444123
7	5820000900	−4.473375	4860005700	0.23913
Channel 2 Segment				
1	780026100	−0.2838693	990025050	0.6258904
2	1110024450	0.2364199	1110024450	0.2364199
3	3960010200	−2.745585	2640016800	0.888033
5	5790001050	−4.136453	5010004950	−1.064739
6	5820000900	−4.472594	6000000000	−3.501008

Constants Used Throughout This Document

Note The logic values attached to pass and fail indicators were chosen to be consistent with the current logic used in the standard OUTPLIML and OUTPLIMF commands.

Table 2-7. Pass/Fail/No_Limit Status Constants

Status Definition	Status Indicator
PASS	1
FAIL	0
NO_LIMIT	−1

Table 2-7 is an interpretation of the Pass/Fail/No_Limit status constants. These constants are used to identify the Pass/Fail/No_Limit state on the data strings if status is returned.

Table 2-8. Min/Max Test Constants

String	Stimulus Value	Data Value
NO_DATA	0	−1000

Table 2-8 is an interpretation of the min/max test constants. If the selected segment has no associated limit, the NO_DATA string is generated, which reports a stimulus value of 0 and a data value of −1000.

Output Limit Test Pass/Fail Status Per Limit Segment

Two commands allow you to query the pass/fail test status on a limit segment basis.

- SELSEG[D] will select the segment.
- OUTPSEGF will return the status of the limit test for that segment: 1 (PASS), 0 (FAIL) or −1 (NO_LIMIT) if no limit exists for the selected segment number. Due to the non-sequential numbering of actual limit line segments on the screen, some segment numbers will have no associated limits and will thus return −1 (NO_LIMIT).

  Under the following conditions, OUTPSEGF will issue the following errors:

  - If the limit testing is OFF: "30: Requested Data Not Currently Available." To clear the error message, turn the limit test ON.
  - If the limit table is empty: "204: Limit Table Empty" (this is a new message). To clear the error message, enter a limit table.

    In both cases, the error is issued and the command responds with −1 (NO_LIMIT).

The argument for SELSEG[D] is limited by the maximum number of segments allowed in the limit table, which is currently 18. The minimum value for the argument is 1. If the user inputs a number that is outside this range, the active entry limits are invoked, causing the analyzer to return the status for limit 18.

Example:

Sending SELSEG3 and OUTPSEGF may return the following:

1 (segment number 3 passed)

Note The output is ASCII. Currently, the formatting for integer numbers appears to append a trailing space.

Output Pass/Fail Status for All Segments

The HP-IB command OUTPSEGAF will return the number of segments being reported, followed by pairs of data consisting of the segment number and its status. A segment is reported only if it has an associated limit. The output is only valid if limit test is on. See the previous discussion, "Output Limit Test Pass/Fail Status Per Limit Segment."

Example:

Sending OUTPSEGAF may return the following:

```
3
1 , 0
3 , 1
5 , 0
```

For an explanation of these results, see table Table 2-9.

Note A new Line Feed character [LF] is inserted after the number of segments and after each data pair.

Table 2-9. Example Output: OUTPSEGAF (pass/fail for all segments)

SEGMENTS REPORTED	SEGMENT NUMBER	STATUS	STATUS DEFINITION
3			
	1	0	FAIL
	3	1	PASS
	5	0	FAIL

Table 2-9 is an interpretation of the data returned by the command OUTPSEGAF. For clarification, status definition is also included.

Example Program of OUTPSEGAF Using BASIC

The following program is not included on the Programming Examples disk:

```
10 OUTPUT 716; "outpsegaf;"

20 ENTER 716; Numsegs

30 PRINT "Receiving status for"; Numsegs; "segments."

40 IF Numsegs>0 THEN

50    FOR I=1 TO Numsegs

60      ENTER 716; Segnum, Pf

70      PRINT USING "DD, 2X, 8A"; Segnum, Pf

80    NEXT I
```

The example program shows how the OUTPSEGAF command can be used to request the number of active segments and their status. Notice that each segment result must use a new enter command as a line feed terminates each segment data result.

Output Minimum and Maximum Point Per Limit Segment

The command "MINMAX"<ON|OFF> toggles a feature which records the minimum and maximum data points in all active limit segments. Note that limit testing need not be turned on.

The command OUTPSEGM will report the min/max data for the segment previously selected by SELSEG[N]. The data is returned in a comma delimited string with the segment number, minimum point stimulus, minimum trace value, maximum point stimulus and maximum trace value.

Under the following conditions, OUTPSEGM will issue the following errors:

- If the min/max testing is OFF: "30: Requested Data Not Currently Available." To clear the error message, turn the min/max testing ON.
- If the limit table is empty: "204: Limit Table Empty" (this is a new message). To clear the error message, enter a new limit table.

  When the above error conditions occur, there is no data to report, thus no output is generated.

  If the selected segment has no associated limit, the NO_DATA string is generated, which reports a stimulus value of 0 and a data value of −1000.

Example:

Sending SELSEG3 and OUTPSEGM may return the following:

3, 1.900000000E+09, −9.900000E−01, 2.123456789E+09, 2.123456E+00

For an explanation of these results, see Table 2-10.

Table 2-10. Example Output: OUTPSEGM (min/max per segment)

SEGMENT	MIN PT STIMULUS (FREQUENCY)	MIN PT VALUE (dB)	MAX PT STIMULUS (FREQUENCY)	MAX PT VALUE (dB)
3	1.9 GHz	−.99	2.12 GHz	2.12

Table 2-10 is an interpretation of the min/max data returned using the SELSEG[N] and OUTPSEGM commands.

Note A new Line Feed character [LF] is inserted after the segment number and after each data pair.

Output Minimum and Maximum Point For All Segments

Three HP-IB commands allow the user to dump the min-or-max or min-and-max values for all active segments:

- OUTPSEGAM: outputs min and max data for each active segment.
- OUTPAMIN: outputs the min data for each active segment.
- OUTPAMAX: outputs the max data for each active segment.

The OUTPSEGAM output consists of :

- The total number of segments being reported.
- The following data for each segment:
  - segment number
  - min stimulus
  - min value
  - max stimulus
  - max value

Example:

Sending OUTPSEGAM may return the following:

5 ,

1 , 1.900000000E+09, −9.900000E−01, 2.123456789E+09, 2.123456E+00

3 , 2.300000000E+09, −10.00000E−01, 2.600000000E+09, 3.100000E+00

5 , 3.200000000E+09, −10.00000E−01, 3.400000000E+09, 3.100000E+00

7 , 4.300000000E+09, −10.00000E−01, 4.700000000E+09, 3.100000E+00

8 , 5.000000000E+09, −10.00000E−01, 5.400000000E+09, 3.100000E+00

For an explanation of these results, see table Table 2-11.

Note A new Line Feed character [LF] is inserted after the segment number and after each data pair.

Table 2-11. Example Output: OUTPSEGAM (min/max for all segments)

SEGMENTS REPORTED	SEGMENT NUMBER	MIN PT STIMULUS (FREQUENCY)	MIN PT VALUE (dB)	MAX PT STIMULUS (FREQUENCY)	MAX PT VALUE (dB)
5					
	1	1.9 GHz	−.99	2.12 GHz	2.12
	3	2.3 GHz	−1.0	2.6 GHz	3.1
	5	3.2 GHz	−1.0	3.4 GHz	3.1
	7	4.3 GHz	−1.0	4.7 GHz	3.1
	8	5.0 GHz	−1.0	5.4 GHz	3.1

Table 2-11 is an interpretation of the min/max data returned using the OUTPSEGAM command.

Example Program of OUTPSEGAM Using BASIC

The following program is not included on the Programming Examples disk:

```
10 Minmax:                    !

20 Mm:  IMAGE DD,":",2X,D.DDDE,2X,SD.DDDE,2X,D.DDDE,2X,SD.DDDE

30      PRINT "TESTING: OUTPSEGAM: min/max points for each segment"

40      OUTPUT 716;"minmaxon;"

50      OUTPUT 716;"outpsegam;"

60      ENTER 716;Numsegs

70      PRINT "receiving data for";Numsegs;"segments"

80      FOR I=1 TO Numsegs

90          ENTER 716;Segnum,Minstim,Minval,Maxstim,Maxval

100         PRINT USING Mm; Segnum, Minstim, Minval,Maxstim,
            Maxval

110     NEXT I
```

Output Data Per Point

The HP-IB command OUTPDATP returns the value of the selected point using FORM4 (ASCII). The point is selected using the SELPT[N] command. This returns the last point if the selected point is out of range. Otherwise, it uses the same format as that used by the marker value command. These formats are as follows:

Table 2-12. Example Output: OUTPDATP (data per point)

Display Format	Marker Mode	Marker Readout Format	Example Returns	
Log Mag		dB*	−3.521 (dB)	9.7E−39*
Phase		degrees*	157.8 (Deg)	5.3E−15*
Delay		seconds*	0.5068x10−9	0*
Smith Chart	LIN MKR	lin mag, degrees		
	LOG MKR	dB, degrees		
	Re/Im	real, imag		
	R + jX	real, imag ohms	10.37 Ω	9.399 Ω
	G + jB	real, imag Siemans		
POLAR	LIN MKR	lin mag, degrees	0.6667	157.8 (Deg)
	LOG MKR	dB, degrees	−3.521 (dB)	157.8 (Deg)
	Re/Im	real, imag	−0.6173	0.2518
LIN MAG		lin mag *	0.6667	0*
REAL		real *		
SWR		SWR *	5.001	0*
* Value is insignificant, but is included in data transfers.				

The commands in the following example are sent while using the format command LOGM.

Example:

Sending SELPT5 and OUTPDATP may return the following:

−3.513410E+00, 0.00915E+15 (Note that the second number is insignificant.)

Output Data Per Range of Points

The HP-IB command OUTPDATR returns the value of the selected points using FORM4 (ASCII). This ASCII format requires many data bytes per point for transfer. For a large number of points, it may be faster to make trace data dumps (OUTPDATA) using a binary format. The range of points is selected using the SELMINPT[N] and SELMAXPT[N] commands (select minimum point, select maximum point of desired point range). These commands return the last max point if the selected points are out of range. Only the SELMAXPT will be returned if the selected minimum point is greater than the selected maximum point.

The commands in the following example are sent while using the format command LOGM.

Example:

Sending SELMINPT5, SELMAXPT7 and OUTPDATR may return the following:

3.880465E−01, 0.000039E−01

1.901648E−01, 1.363988E+11

5.57587E−01, 1.258655E+30 (Note that the second number is insignificant)

For an explanation of these results see Table 2-13.

Note A new Line Feed character [LF] is inserted after the segment number and after each data pair.

Table 2-13. Example Output: OUTPDATPR (data per range of points)

POINT	VALUE	VALUE*
5	.3880465	0.000039E−01
6	.1901648	1.363988E+11
7	.557587	1.258655E+30
* These values are insignificant.		

Table 2-13 is an interpretation of the min/max data per range of points returned using the SELMINPT5, SELMAXPT7 and OUTPDATR commands.

Output Limit Pass/Fail by Channel

The HP-IB commands OUTPLIM1 and OUTPLIM2 output the status of the limit test for channel 1 and channel 2, respectively.

These commands return the values 1 (PASS), 0 (FAIL), or −1 (NO_LIMIT) if limit testing is disabled. Currently, the results of limit testing can be retrieved by reading a bit in the status register.

Example:

Sending OUTPLIM1 or OUTPLIM2 (channel 1 or channel 2) may return the following:

1 (PASS), 0 (FAIL), or if limit test not enabled then −1 (NO_LIMIT).

Index

A

abort sequence, 2-7
additional information, 2-1
  BASIC 6.2, 2-1
analyzer-debug mode, 2-13
analyzer features helpful in developing programs, 2-13
analyzer operating modes, 2-3
  pass-control mode, 2-3, 2-66
  system-control mode, 2-3
  talker/listener, 2-3, 2-64
array-data formats, 2-26
  FORM 1, 2-26
  FORM 2, 2-26
  FORM 3, 2-26
  FORM 4, 2-24, 2-26
  FORM 5, 2-26
ASCII disk files, 2-72
  reading, 2-72

C

calibrating the test setup, 2-10
calibration data, 2-46
  inputting, 2-46
  outputting, 2-46
  reading, 2-46
calibration example program, 2-18
calibration kit, 2-2
calibration kits, 2-18
clearing any messages waiting to be output, 2-7
clearing syntax errors, 2-7
clearing the input-command buffer, 2-7
clear sequence, 2-7
command structure, 2-4
command structure elements, 2-4
  appendage, 2-4
  BASIC command statement, 2-4
  data, 2-4
  terminators, 2-4
  unit, 2-4
compatible peripherals, 2-2
connecting the device under test, 2-10
connecting the test system, 2-2
CONSTANTS, 2-83
controlled sweep, 2-13
correction of errors example program, 2-18

D

data formats and transfers, 2-23
data taking, 2-11
data transfer, 2-11, 2-23
  to a plotter, 2-69
  using floating-point numbers, 2-29
  using FORM 1, 2-35
  using FORM 4, 2-26
  using frequency-array information, 2-32
  using markers, 2-24
debug mode, 2-5, 2-13
developing program features, 2-13
device connection, 2-10

E

equipment
  optional, 2-2
  required, 2-1
error-correction example program, 2-18
error queue, 2-38
event-status-register B, 2-61
event-status registers, 2-37
example
  measurement calibration, 2-18

F

features helpful in developing programming routines, 2-13
formats and transfers of trace-data, 2-23
frequency calculation equation, 2-26

H

helpful features for developing programs, 2-13
HP 9000 Series 300 computer, 2-1
HP-IB interconnect cables, 2-1

I

information on programs, 2-13
input/output path, 2-8
instrument setup, 2-10
instrument states, 2-44
  recalling, 2-44, 2-49
  saving, 2-44, 2-49

interrupts, generating, 2-41

K

kits of calibration standards, 2-18

L

limit line and data point special functions, 2-77
limit lines, 2-58
  setting up, 2-58
limit-line testing, 2-52
  list-frequency table, creating, 2-52
  list-frequency table, selecting a single segment, 2-55
  performing PASS/FAIL tests, 2-58
  using list-frequency mode, 2-52
limit-test table, 2-58
  creating, 2-58
  transmitting, 2-58
list-frequency mode, 2-52
local lockout, 2-5
local mode, 2-5

M

marker positioning, 2-24
  by data point location, 2-24
  by frequency location, 2-24
  by trace-data value, 2-24
measurement calibration example program, 2-18
measurement data post-processing, 2-11
measurement data taking, 2-11
measurement parameters
  required order, 2-14
  setting, 2-14
  verifying, 2-16
measurement process, 2-10
measurement setup, 2-14
measurement specifications, 2-28
  group delay, 2-28
  magnitude, 2-28
  phase, 2-28
memory requirements, 2-1
MINMAX<ON|OFF>, 2-78
modes
  debug, 2-13

O

operation complete commands, 2-7
OUTPAMAX, 2-78
OUTPAMIN, 2-78
OUTPDAPT, 2-78
OUTPDATR, 2-78
OUTPFAIP, 2-78
OUTPLIM1, 2-78
OUTPLIM2, 2-78
OUTPSEGAF, 2-78
OUTPSEGAM, 2-78
OUTPSEGF, 2-78
OUTPSEGM[D], 2-78
OUTPSERN, 2-78
Output Data Per Point, 2-90
Output Data Per Range of Points, 2-91
Output Limit Pass/Fail by Channel, 2-92
Output Limit Test Pass/Fail Status Per Limit Segment, 2-84
Output Minimum and Maximum Point For All Segments, 2-88
Output Minimum and Maximum Point Per Limit Segment, 2-87
Output Pass/Fail Status for All Segments, 2-85

P

PASS/FAIL tests, 2-61
PC-graphics applications example program, 2-71
plot file and PC-graphics example program, 2-71
plotting
  to a file, 2-69
plotting, remote, 2-64, 2-66
post-processing the measurement data, 2-11
preparing for remote operation, 2-7
presetting the instrument, 2-7
printing, remote, 2-64, 2-66
processing after taking measurement data, 2-11
process of measuring, 2-10
program debugging, 2-13
program development features, 2-13
program example
  measurement calibration, 2-18
program information, 2-13

Q

querying commands, 2-5

R

recommended disk drives, 2-2
recommended plotters, 2-2
recommended printers, 2-2
remote mode, 2-5
report generation, 2-64
routing debugging, 2-13

S

SELMAXPT[D], 2-78
SELMINPT[D], 2-78
SELPT[D], 2-78
SELSEG[D], 2-78
serial poll, 2-37
service request, 2-41
setting addresses, 2-2
setting the control mode, 2-2
setting up the instrument, 2-10
setting up the system, 2-2
status byte, 2-37
STATUS CONSTANTS, 2-83
status reporting, 2-37
step 1 of a measurement, 2-10
step 2 of a measurement, 2-10
step 3 of a measurement, 2-10
step 4 of a measurement, 2-11
step 5 of a measurement, 2-11
step 6 of a measurement, 2-11
sweep user-controlled, 2-13
synchronization, 2-37
system setups, 2-44
  reading calibration data, 2-46
  using the learn string, 2-44

T

taking the measurement data, 2-11
test port return cables, 2-2
test setup calibration, 2-10
trace-data formats and transfers, 2-23
transferring the measurement data, 2-11
transfers and formats of trace-data, 2-23
troubleshooting, 2-3, 2-5

U

user-controllable sweep, 2-13

V

verifying HP-IB operation, 2-2