Agilent E5100A/B Network Analyzer Programming Manual

Manual Change

Agilent Part No. N/A

June 2007

Change 1

Change the note of ANARANGE command (page D-4) as follows.

Note

- The waveform analysis range is independent of the marker search range.
- You can set the range for each channel independently. Therefore, you need to set the analysis channel using ANAOCH1 or ANAOCH2 before using ANARANG.
- The waveform analysis range is not automatically changed to an appropriate range when the stimulus range or Number of Point setting is modified. If any stimulus setting is changed, set the waveform analysis range so as to fit the displayed stimulus range.
- Store the waveform analysis range setting using SAVE ALL or STATE ONLY.
- The waveform analysis range is set to equal to the displayed stimulus range when the power is turned on.

Change 2

Add the following note to the “Maximum/Minimum/Mean Value Search Commands” (page D-7).

Note

- The waveform analysis stimulus range is not automatically changed to an appropriate range when the sweep Start, Stop or Number of Point setting is modified. To ensure correct waveform analysis results, you need to set appropriate stimulus range for waveform analysis command by using ANARANG, ANARANGP or ANARFULL command before the maximum/minimum/mean value search command is executed.

マニュアル　チェンジ

変更 1

ANARANGE コマンドの「記」の記述（ページ D-2）を以下に変更して下さい。

記

- 波形解析レンジは、マーカ・サーチ・レンジとは独立に設定されます。
- 各チャンネルに独立して設定する事ができます。このため、ANARANG を使用する前に、ANAOCH1、ANAOCH2 で解析対象となるチャンネルをあらかじめ選択する必要があります。
- 表示レンジや測定点数の設定を変更したとき、波形解析レンジは適切なレンジに設定されません。もし設定を変更した場合は、表示された範囲に合うように波形解析レンジを設定して下さい。
- 波形解析レンジの設定は、SAVE ALL または STATE ONLY でセーブされます。
- 電源投入時の初期設定は、画面表示レンジ全体です。

変更 2

最大/最小/平均値解析コマンドの記述（ページ D-5）に以下を追加して下さい。

記

- 表示レンジ、スタート値、ストップ値や測定点数の設定を変更したとき、波形解析レンジは適切なレンジに設定されません。正しい結果を得るために、最大/最小/平均値解析コマンドが実行される前に、ANARANGE, ANARNAGP および ANARFULL コマンドを使用して、適切なレンジを設定する必要があります。

Agilent E5100A/B Network Analyzer

Programming Manual

SERIAL NUMBERS

This manual applies directly to instruments with serial number prefix JP2KC,JP3KC,JP4KC,JP5KC and MY405, or firmware revision 2.xx and 3.xx. For additional important information about serial numbers, read "Serial Number" in Appendix A.

Agilent Technologies

Agilent Part No. E5100-90087
Printed in JAPAN July 2001

Eleventh Edition

Notice

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

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

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

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

Agilent Technologies DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

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

Manual Printing History

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

July 1995 .... First Edition

October 1995 .... Second Edition

November 1995 .... Third Edition

December 1995 .... Fourth Edition

September 1996 .... Fifth Edition

May 1997 .... Sixth Edition

December 1997 .... Seventh Edition

January 1998 .... Eighth Edition

September 1998 .... Nineth Edition (part number: E5100-90077)

March 2000 .... Tenth Edition (part number: E5100-90077)

July 2001 .... Eleventh Edition (part number: E5100-90087)

Typeface Conventions

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

Contents

1. Introduction
How to Use This Manual 1-1
Target Reader 1-1
What's in This Manual? 1-1
How to Use the Program Modules 1-2
Building a Working Program Using Program Modules 1-2
Initializing Module 1-2
Example 1-3
GPIB Overview 1-4
Controller 1-4
Device Selector 1-5
Preparation for Operation 1-6
Using Instrument BASIC as the Controller (except for Option UKR) 1-6
1. Connecting the GPIB Cables For Instrument BASIC 1-6
2. Setting the GPIB Address For Instrument BASIC 1-6
3. Setting Up Instrument BASIC 1-6
Using an External Controller 1-7
1. Connecting External GPIB Cables 1-7
2. Setting the GPIB Address 1-7
3. Preparing For HP BASIC Operation 1-7
Sample Program Disk 1-8
Loading a Program from Disk 1-8
Reading the Sample Program Disk on a PC 1-8
Related Documentation Information 1-9

2. Setup and Measurement Program
Overview of GPIB Control 2-1
Sending GPIB Commands 2-1
Sending a Query and Reading the Response 2-2
Automating a Measurement Procedure 2-3
1. Setting the Active Channel 2-3
2. Setting the Measurement and Format 2-3
3. Sweep Setup 2-4
4. Calibration 2-4
Calibration Process 2-5
5. Device Connection 2-5
6. Triggering a Measurement 2-6
When You are Using an External Controller 2-6
7. Data Processing and Transfer 2-6
8. Exiting the Program 2-6
Sample Program -1: Basic Measurement 2-7
Column Parameters and Variables 2-8

3. Data Processing and Transfer
Introduction 3-1
Data Arrays 3-1
Raw Data Array 3-4
Data Array 3-4
Unformatted Data Array 3-4
Formatted Data Array 3-5
Calibration Coefficient Array 3-5
Accessing Arrays 3-5
Arrays for Memory Trace 3-6
Accessing Memory Array 3-6
Accessing Trace Array 3-6
Data Transfer Methods 3-7
ASCII Transfer 3-7
Binary Transfer 3-8
Data Header 3-9
Getting Data from the Analyzer 3-10
Sample Program -2: Binary Data Transfer 3-11

4. Using the I/O Port
Introduction 4-1
Output Port (Pin 5 to 28) 4-1
Data Output 4-1
Input Port (Pin 21 to 28) 4-2
Data Input 4-2
INPUT1 Input (Pin 2), OUTPUT1 Output (Pin 3), OUTPUT2 Output (Pin 4) . . . 4-3
Examples of Data I/O 4-4
The Parallel I/O Mode A (Option 005 Only) 4-6
Data Output (4 Bit) 4-6
Data Input (8 Bit) 4-6

5. Using Status Reporting Function
General Status Register Model 5-1
Event Register 5-2
Enable Register 5-2
Status Byte Register 5-2
Transition Filter and Condition Register 5-3
Status Register Structure 5-4
OSPT, OSNT 5-7
OSPT (Operational Status Positive Transition Filter) 5-7
OSNT (Operational Status Negative Transition Filter) 5-8
How to Use the Status Registers in a Program 5-9
Reading an Event Register Directly 5-9
SRQ and Interrupt 5-10

6. Programming Miscellaneous
Introduction 6-1
Using an External Controller and the E5100A/B Together 6-1
Locking Out Local Operation (LOCAL LOCKOUT) 6-1
How To Pass Control (PASS CONTROL) 6-2
Controlling Instrument BASIC From an External Controller 6-3
Referring to and Transferring an Array 6-3
Referring to a Numeric Variable 6-3
Referring to a String Character Variable 6-3
Transferring a Numeric Variable 6-4

Transferring a String Character Variable 6-4
Referring to and Transferring an Array 6-4
Remote Processing a BASIC Command (PROG:EXEC) 6-4
Controlling the Process Status 6-5
Controlling the Process Status of Instrument BASIC 6-5
Checking the Process Status of an Instrument BASIC Program 6-5
Transferring a Program 6-6
Transferring from an External Controller to Instrument BASIC 6-6
Transferring from Instrument BASIC to an External Controller 6-7
Sample Program -3: Controlling Instrument BASIC from an External Controller 6-8
The Program for Instrument BASIC 6-9
The Program for the External Controller 6-11
Programming Techniques 6-13
Using a Disk 6-13
Using a Softkey Label (ON KEY LABEL) 6-14
Displaying the Softkey Label 6-14
Measuring the Processing Time (TIMEDATE) 6-15
Checking GPIB Errors (OUTPERRO?) 6-15
Tips for Increasing Speed 6-15
Sample Program -4: Reading and Running a Program by a Softkey Operation . 6-17
Using the Parallel Processing 6-19
Programming for Parallel Processing 6-19
Sample Program -4a :Parallel Processing 6-20
Notes for Parallel Processing 6-22

7. Application Sample Programs
Sample Program -5: Setting Up and Performing the List Sweep 7-2
Saving and Resetting Calibration Data 7-4
Sample Program -6: Saving Calibration Data in an External Controller 7-4
Sample Program -7: Setting Up Calibration Data from an External Controller . 7-5
Sample Program -8: Analyzing a Ceramic Resonator 7-8
Sample Program -9: Analyzing a Crystal Resonator 7-10

8. Command Reference
Common Command 8-1
*CLS 8-1
*ESE <*value*> 8-1
*ESR? 8-1
*IDN? 8-1
*OPC 8-2
*PCB <*value*> 8-2
*RST 8-2
*SRE <*value*> 8-2
*STB? 8-3
*TRG 8-3
*TST? 8-3
*WAI 8-3
Command Reference 8-4
ADTOTRAC 8-4
ANAMODE {GAINP|ZREFL|ZTRAN} 8-4
ANAOCH{1-4} 8-4
ANAODATA 8-4
ANAOMEMO 8-4
ANAPOINS?␣<*value*> 8-5
ANARANG␣<*value1*>,<*value2*> 8-5

ANARANGP␣<*value1*>,<*value2*> . . . 8-5
ANARFULL . . . 8-5
ANASTIMP?␣<*value*> . . . 8-6
ATRC␣{1|2} . . . 8-6
ATTI{A|B|C|R|}␣{0|25} . . . 8-6
ATTIAAUTO␣{OFF|ON|0|1} . . . 8-7
ATTIBAUTO␣{OFF|ON|0|1} . . . 8-7
ATTICAUTO␣{OFF|ON|0|1} . . . 8-7
ATTIRAUTO␣{OFF|ON|0|1} . . . 8-7
ATTW␣<*value*> . . . 8-8
AUTO . . . 8-8
BASL . . . 8-8
BEEPDONE␣{OFF|ON|0|1} . . . 8-8
BEEPFAIL␣{OFF|ON|0|1} . . . 8-9
BEEPWARN␣{OFF|ON|0|1} . . . 8-9
BINSIZE␣<*value*> . . . 8-9
BLIGHT␣{OFF|ON|0|1} . . . 8-10
BOTV␣<*value*> . . . 8-10
CALCOPY␣<*value1*>, <*value2*> . . . 8-10
CALI␣{NONE|RESP|RAI|ONEP} . . . 8-10
CALK{O|S|L}{LS|RS|CP}␣<*value*> . . . 8-11
CENS␣<*value1*>,<*value2*> . . . 8-11
CENT␣<*value*> . . . 8-12
CHAD␣<*string*> . . . 8-12
CHAN␣{1|2|3|4} . . . 8-12
CIN . . . 8-12
CIVAL␣<*value*> . . . 8-12
CLACT␣<*Value*> . . . 8-13
CLASS11{A|B|C} . . . 8-13
CLEL . . . 8-14
CLEM{1-4} . . . 8-14
CLEMNU3 . . . 8-14
CLES . . . 8-14
CLOSE . . . 8-14
CLTGT␣<*Value*> . . . 8-15
CONT . . . 8-15
CONV␣{OFF|ZTRA|YTRA|ZREF|YREF} . . . 8-15
COPYRIGHT␣{OFF|ON|0|1} . . . 8-16
CORR␣{OFF|ON|0|1} . . . 8-16
CORRS? . . . 8-16
COUC␣{OFF|ON|0|1} . . . 8-16
COUT . . . 8-17
CRED␣<*string*> . . . 8-17
CURD? . . . 8-17
CURMPOIN? . . . 8-17
CWFREQ␣<*value*> . . . 8-17
DATAM␣{NONE|RESP|RAI|ONEP} . . . 8-18
DATAMN␣{1-10} . . . 8-18
DATI . . . 8-18
DAYMYEAR . . . 8-19
DELA . . . 8-19
DELO . . . 8-19
DELR{1-4} . . . 8-19
DELRFIXM . . . 8-19
DIN . . . 8-20

DISA⊔{ALLI|HIHB|ALLB|BASS} 8-20
DISG⊔{OFF|ON|0|1} 8-20
DISP⊔{DATA|MEMO|DATM|DDM|DMM} 8-20
DIST⊔{OFF|ON|0|1} 8-21
DONE 8-21
DOUT 8-21
DSKEY 8-22
DUAC⊔{OFF|ON|0|1} 8-22
EDITDONE 8-22
EDITLIS1 8-22
EDITLIS2 8-23
EDITLIST 8-23
ELED⊔*<value>* 8-23
ENKEY 8-23
EQUC0?⊔*<value>* 8-23
EQUCPARA5? 8-24
EQUCPARA? 8-24
EQUCPARS4? 8-25
EQUCPARS? 8-25
EQUM⊔*<value>* 8-26
ESB? 8-26
ESNB⊔*<value>* 8-26
EXET⊔*<value>* 8-26
EXPC⊔{OFF|ON|0|1} 8-27
EXPZP⊔{OFF|ON|0|1} 8-27
EXTRLOCK? 8-27
EXTT⊔{OFF|ONSWEE} 8-27
FILC⊔*<string1>*,*<string2>*,*<string3>*,*<string4>* 8-28
FMT⊔{LOGM|PHAS|DELA| ... |ADMG|ADMB|MAGZDF} 8-28
FNAME?⊔*<value>* 8-29
FNUM? 8-30
FSIZE?⊔*<string>* 8-30
GRAPCOL⊔{FIXC|VARC|MONO} 8-30
GRAPFORM⊔{PCL|TIFF} 8-30
GRODAPER *<value>* 8-31
HOLD 8-31
IDN? 8-31
IFBW⊔*<value>* 8-31
IFBWAUTO 8-32
INID 8-32
INP8IO 8-32
INPT? 8-33
INPUCALC{01-03}⊔*<value1>*,*<value2>*, ... ,*<valuen>* 8-33
INPUDATA⊔*<value1>*,*<value2>*, ... ,*<valuen>* 8-33
INPUDATAM{01-03}⊔*<value1>*,*<value2>*, ... ,*<valuen>* 8-33
INPUFORM⊔*<value1>*,*<value2>*, ... ,*<valuen>* 8-34
INPUIFORM⊔*<value1>*,*<value2>*, ... ,*<valuen>* 8-34
INPUMEMO⊔*<value1>*,*<value2>*, ... ,*<valuen>* 8-34
INPURAW⊔*<value1>*,*<value2>*, ... ,*<valuen>* 8-34
INPURFORM⊔*<value1>*,*<value2>*, ... ,*<valuen>* 8-35
INPURTMEM⊔*<value1>*,*<value2>*, ... ,*<valuen>* 8-35
INPUSTIM⊔*<value1>*,*<value2>*, ... ,*<valuen>* 8-35
INPUTRAC⊔*<value1>*,*<value2>*,*<value3>* 8-35
INPUTRACB⊔*<value1>*,*<value2>*,*<value3>* 8-36
IOPO? 8-36

LIMILINE␣{OFF|ON|0|1} . . . 8-37
LIMITEST␣{OFF|ON|0|1} . . . 8-37
LISDFBASE . . . 8-37
LISDOBASE . . . 8-38
LISFREQ . . . 8-38
LISSLIS1 . . . 8-38
LISSLIS2 . . . 8-38
LMAX?␣<*value*> . . . 8-38
LMAXS?␣<*value*> . . . 8-39
LMIN?␣<*value*> . . . 8-39
LMINS?␣<*value*> . . . 8-39
LOWELIMI␣<*value1*><*value2*><*value3*>, . . . ,<*valuen*> . . . 8-40
MARD␣{OFF|ON|0|1} . . . 8-40
MARK{1-4}␣<*value*> . . . 8-40
MARKBUCK␣<*value*> . . . 8-41
MARKCENT . . . 8-41
MARKCONT . . . 8-41
MARKCOUP . . . 8-41
MARKDISC . . . 8-41
MARKFSTI␣<*value*> . . . 8-42
MARKFVAL␣<*value*> . . . 8-42
MARKL␣{OFF|ON|0|1} . . . 8-42
MARKODATA . . . 8-43
MARKOFF . . . 8-43
MARKOMEMO . . . 8-43
MARKREF . . . 8-43
MARKSPAN . . . 8-43
MARKSTAR . . . 8-44
MARKSTOP . . . 8-44
MARKTIME␣{OFF|ON|0|1} . . . 8-44
MARKUNCO . . . 8-44
MARKZERO . . . 8-45
MAXPOIN? . . . 8-45
MAXPORT? . . . 8-45
MEAS1PT?␣<*value*> . . . 8-45
MEAS␣{AR|BR|CR|RA|BA|CA|RB|AB|CB|RC|AC|BC|R|A|B|C} . . . 8-46
MEASA . . . 8-46
MEASR . . . 8-46
MEASTAT␣{OFF|ON|0|1} . . . 8-46
MENU3␣<*value*>,<*string1*>,<*string2*>,<*string3*> . . . 8-47
MOHMSW␣{A|B},{ON|OFF} . . . 8-47
MONDYEAR . . . 8-47
MOVADARY . . . 8-47
MULC␣{OFF|ON|0|1} . . . 8-48
NEGL . . . 8-48
NEXPK? . . . 8-48
NEXNPK? . . . 8-48
NOMF␣<*Value*> . . . 8-49
NPEAK? . . . 8-49
NPEAKLIST? . . . 8-49
NPEAKSORT?␣<*value*> . . . 8-50
NUMC␣<*value*> . . . 8-50
NUMG␣<*value*> . . . 8-50
NUMLMAX? . . . 8-50
NUMLMIN? . . . 8-51

NUMLMINMAX? 8-51
OSE␣*<value>* 8-51
OSER? 8-51
OSNT␣*<value>* 8-51
OSPT␣*<value>* 8-52
OSR? 8-52
OUT1ENVH 8-52
OUT1ENVL 8-52
OUT1H 8-53
OUT1L 8-53
OUT2ENVH 8-53
OUT2ENVL 8-53
OUT2H 8-53
OUT2L 8-53
OUT8IO␣*<value>* 8-54
OUTAIO␣*<value>* 8-54
OUTBIO␣*<value>* 8-54
OUTCIO␣*<value>* 8-54
OUTDIO␣*<value>* 8-54
OUTEIO␣*<value>* 8-55
OUTFIO␣*<value>* 8-55
OUTGIO␣*<value>* 8-55
OUTHIO␣*<value>* 8-55
OUTPCALC{01-03}? 8-55
OUTPCERR? 8-56
OUTPCF2?␣*<value1>*,*<value2>*,*<value3>*,*<value4>*,*<value5>* ...
,*<valuen+4>* 8-56
OUTPCFIL?␣*<value1>*,*<value2>*, ... , *<value6>* 8-57
OUTPDATA? 8-58
OUTPDATAM{01-03}? 8-58
OUTPDATAP␣*<value>* 8-59
OUTPDATAT? 8-59
OUTPDATTP?␣*<value>* 8-59
OUTPERRO? 8-60
OUTPFILT?␣*<value>* 8-60
OUTPFORM? 8-60
OUTPFORMP?␣*<value>* 8-61
OUTPIFORM? 8-61
OUTPINP8IO? 8-61
OUTPINPCIO? 8-62
OUTPINPDIO? 8-62
OUTPINPEIO? 8-62
OUTPIRFORM? 8-62
OUTPIRTMEM? 8-62
OUTPMARK? 8-63
OUTPMAX? 8-63
OUTPMEAN? 8-63
OUTPMEMO? 8-64
OUTPMEMOT? 8-64
OUTPMEMTP?␣*<value>* 8-64
OUTPMIN? 8-64
OUTPMINMAX? 8-65
OUTPMSTA? 8-65
OUTPMWID? 8-65
OUTPMWIL? 8-66

OUTPMWLF? 8-66
OUTPRAW? 8-66
OUTPRESF?⊔<*value1*>,<*value2*> 8-67
OUTPRESO? 8-67
OUTPRESR? 8-67
OUTPRFORM? 8-68
OUTPRFORMP?⊔<*value*> 8-68
OUTPRTMEM? 8-68
OUTPRTMEMP?⊔<*value*> 8-69
OUTPSTIM? 8-69
OUTPTITL? 8-69
OUTPTRAC?⊔<value> 8-69
OUTPTRACB?⊔<*value1*>,<*value2*> 8-70
OUTPXF2?⊔<*value1*>,<*value2*>,<*value3*>, ... ,<*valuen+3*> 8-70
OUTPXFIL?⊔<*value1*>,<*value2*>,<*value3*>,<*value4*>,<*value5*> 8-71
PARS⊔{OFF|ON|0|1} 8-72
PARSMODE⊔{OFF|ON|0|1} 8-73
PARSRANG⊔<*value1*>, <*value2*> 8-73
PEAK? 8-73
PEAKLIST?⊔<*value*> 8-74
PEAKSORT?⊔<*value*> 8-74
PHAO⊔<*value*> 8-74
PHAS 8-74
PICIRC⊔{OFF|ON|0|1} 8-75
POIN⊔<*value*> 8-75
POLE?⊔<*value*> 8-75
POSL 8-76
POWE⊔<*value*> 8-76
POWS 8-76
POWU⊔{DBM|WATT|AMP} 8-76
PRES 8-77
PRINALL 8-77
PRIR⊔<*string*> 8-77
PTABORT⊔{OFF|ON|0|1} 8-77
PTFOVHD⊔<*value*> 8-78
PTFR⊔<*value*> 8-78
PTFRSR⊔<*value*> 8-78
PTPARA⊔<*value*> 8-78
PTRACK⊔{OFF|ON|0|1} 8-79
PTREPN⊔<*value*> 8-79
PTSTAT? 8-79
PTTRGLMT⊔<*value*> 8-79
PTTRGPHS⊔<*value*> 8-80
PURG⊔<*value*> 8-80
RAID 8-80
RAIISOL 8-80
RAIRESP 8-80
READ? 8-81
RECD⊔<*value*> 8-81
REFP⊔<*value*> 8-81
REFV⊔<*value*> 8-81
RESAVD⊔<*string*> 8-82
RESPDONE 8-82
REST 8-82
RFOUTSW⊔{1|2} 8-82

ROPEN⊔<*string*> 8-83
RPLENV? 8-83
RPLHEI? 8-83
RPLLHEI? 8-84
RPLMEA? 8-84
RPLMM? 8-84
RPLPP? 8-84
RPLPPS? 8-85
RPLRHEI? 8-85
RPLVAL? 8-85
RPOS 8-85
SADD 8-86
SAV1 8-86
SAVCA⊔{OFF|ON|0|1} 8-86
SAVDA⊔{OFF|ON|0|1} 8-86
SAVDALL⊔<*string*> 8-87
SAVDASC⊔<*string*> 8-87
SAVDDAT⊔<*string*> 8-87
SAVDGRAP⊔<*string*> 8-87
SAVDMNU3⊔<*string*> 8-88
SAVDSTA⊔<*string*> 8-88
SAVFA⊔{OFF|ON|0|1} 8-88
SAVMA⊔{ON|OFF|1|0} 8-88
SAVRA⊔{OFF|ON|0|1} 8-89
SAVTA⊔{OFF|ON|0|1} 8-89
SAVTMA⊔{OFF|ON|0|1} 8-89
SCAFDATA 8-90
SCAFMEMO 8-90
SCAL⊔<*value*> 8-90
SCAY⊔{1|0} 8-91
SDEL 8-91
SDON 8-91
SEAL 8-91
SEALMAX 8-91
SEALMIN 8-92
SEAM⊔{OFF|MAX|MIN|TARG|MEAN|LMAX|LMIN|PPEAK} 8-92
SEAMAX 8-92
SEAMEAN 8-92
SEAMIN 8-93
SEAOFF 8-93
SEAPPEAK 8-93
SEAR 8-93
SEARSTOR 8-93
SEATARG⊔<*value*> 8-94
SEDI⊔<*value*> 8-94
SEET⊔{OFF|ON|0|1} 8-94
SET1PT⊔<*value*> 8-94
SETCDATE⊔<*numeric (year)*>,<*numeric (month)*>,<*numeric (day)*> 8-95
SETCTIME⊔<*numeric (hour)*>,<*numeric (minute)*>,<*numeric (second)*> . . . 8-95
SETZ⊔<*value*> 8-95
SING? 8-96
SINSPEAK⊔{OFF|ON|0|1} 8-96
SMOO⊔{OFF|ON|0|1} 8-96
SMOOAPER⊔<*value*> 8-96
SPAN⊔<*value*> 8-97

SPLD␣{OFF|ON|0|1} 8-97
SRCHFR?␣{1|2|3|4|5|6},<*value*> 8-97
SRCHR␣<*Value*> 8-98
SRCHTRFL?␣{1|2|3|4|5|6},<*Value*> 8-98
STANC 8-99
STAR␣<*value*> 8-99
STAS␣<*value1*>,<*value2*> 8-99
STAW␣<*value*> 8-100
STIDROUT{1-16}␣<*value*> 8-100
STIMROUT{1-16}␣<*value*> 8-100
STOD{DISK|MEMO} 8-101
STOMDISK 8-101
STOP␣<*value*> 8-101
STPSIZE␣<*value*> 8-101
STR␣{OFF|ON|0|1} 8-102
SWED␣{DOWN|UP} 8-102
SWET␣<*value*> 8-102
SWETAUTO 8-103
SWPT␣{LINF|LOGF|LIST|POWE|RAMPF} 8-103
TARL?␣<*value*> 8-103
TARR?␣<*value*> 8-103
TARSUBL?␣<*value1*>[,<*value2*>] 8-104
TARSUBR?␣<*value1*>[,<*value2*>] 8-104
THRR␣<*value*> 8-104
TIMO␣{OFF|ON|0|1} 8-105
TITL␣<*string*> 8-105
TOPV␣<*value*> 8-105
TOTIME␣<value> 8-105
TRABGE 8-106
TRABLE 8-106
TRACK␣{OFF|ON|0|1} 8-106
TRAFDATA 8-107
TRAFMEMO 8-107
TRAP␣{OFF|ON|0|1} 8-107
TRAR␣<*value1*>,<*value2*> 8-107
TRIGMEAS 8-108
TRIM␣{CONT|HOLD|SING} 8-108
UPDD␣{OFF|ON|0|1} 8-108
UPPELIMI␣<*value1*><*value2*><*value3*> ... <*valuen*> 8-109
WIDT␣{ON|OFF|0|1} 8-109
WIDV␣<*value*> 8-109
WOPEN␣<*string*> 8-110
WRITE␣<*block*> 8-110
WRIT16 8-110
WRIT24 8-110
SCPI Command (PROGram sub-system command) 8-111
:PROGram:CATalog? 8-111
:PROGram[:SELected]:DEFine␣<*block*> 8-111
:PROGram[:SELected]:DELete[:SELected] 8-112
:PROGram[:SELected]:DELete:ALL 8-113
:PROGram[:SELected]:EXECute␣<*string*> 8-113
:PROGram[:SELected]:NUMBer␣<*string*>,<*numeric (1)*>[,<*numeric (2)*>[, ... [,<*numeric (n)*>] 8-113
:PROGram[:SELected]:STATe␣{RUN|PAUSe|STOP|CONTinue} 8-113

:PROGram[:SELected]:STRing␣<*string (varname)*>,<*string (value1)*>[,<*string (value2)*>[, ... [,<*string (value n)*> 8-114
:PROGram[:SELected]:WAIT 8-114

A. Manual Changes
Introduction A-1
Manual Changes A-1
Serial Number A-2
Change 1 . . . . A-3
Change 2 . . . . A-3
Change 3 . . . . A-4
Change 4 . . . . A-4
Change 5 . . . . A-5

B. Command Summary
Meas/Format B-1
Meas/Format Menu B-1
Function Menu B-1
Port Select Menu B-2
Gain-Phase Format Menu B-2
Z Format Menu B-3
Display B-3
Display Menu(1/3) (2/3) (3/3) B-3
Linear Scale Menu B-4
Log Scale Menu B-4
Define Trace Menu B-4
Basic Allocation Menu B-4
Cal B-5
Gain-Phase CAL Menu B-5
Thru CAL Menu B-5
Response & Isolation CAL Menu B-5
Gain-Phase 3 Term CAL Menu B-5
Gain-Phase CAL STD value menu B-6
Z:Refl CAL Menu B-6
Z: Refl CAL Menu B-6
Z: Refl CAL STD value menu B-6
Z: Trans CAL Menu B-6
Z:Trans 3 Term CAL Menu B-7
1 Term CAL Menu B-7
Z: Trans CAL STD value menu B-7
Marker B-7
Marker Menu B-7
Marker Serch Menu B-7
Serch Range Menu B-8
Marker Utility Menu B-8
ΔMode Menu B-8
Fixed Marker Position Menu B-8
Marker Mode Menu B-8
Sweep B-9
Sweep Menu B-9
Sweep Type Menu B-9
Sweep Time Menu B-9
Trigger B-10
Trigger Menu B-10
Start Stop Center Span B-10

→ Function Menu B-10
System B-11
System Menu(1/3) (2/3) (3/3) B-11
Clock Menu B-11
Att Setting Menu B-11
Save/Recall B-11
Save/Recall Menu B-11
Save Menu B-12
File Utility Menu B-12
Binary Define Save Data Menu B-12
ASCII Define Save Data Menu B-12
MISC Save Menu B-13
Menu2 . . . . B-13

C. Data I/O Format
Data Format C-1
Form 2 . . . . C-1
Form 3 . . . . C-1
Form 4 . . . . C-2
Form 5 . . . . C-2
Internal Data Array C-3

D. Waveform Analysis Commands
Conventions and Definitions D-2
Waveform Analysis Setup Commands D-3
ANAOCH{1|2|3|4} D-3
ANARANG D-4
ANARANGP D-4
ANARFULL D-5
ANAODATA D-5
ANAOMEMO D-5
THRR D-6
Maximum/Minimum/Mean Value Search Commands D-7
OUTPMAX? D-7
OUTPMIN? D-7
OUTPMINMAX? D-8
OUTPMEAN? D-8
PEAK? D-9
NEXPK? D-9
NUMLMAX? D-10
NUMLMIN? D-10
LMAX? D-11
LMIN? D-11
TARR? D-12
TARL? D-12
Ripple Analysis Commands D-13
RPLPP? D-13
RPLHEI? D-14
RPLRHEI? D-15
RPLLHEI? D-16
RPLENV? D-17
RPLMEA? D-18
RPLMM D-19
RPLVAL? D-20
POLE? D-21

Filter and Resonator Analysis Commands D-22
OUTPFILT? D-23
OUTPXFIL? D-25
OUTPXF2? D-28
OUTPCFIL? D-30
OUTPCF2? D-33
OUTPRESO? D-36
OUTPRESR? D-38
OUTPRESF? D-40
OUTPCERR? D-42
Equivalent circuit analysis commands D-43
EQUCPARA? D-44
EQUCPARA5? D-46
EQUCPARS? D-47
EQUC0? *value* D-48
EQUCPARS4? D-50

E. Command 87510A v.s. E5100A
COMMAND E-1

F. Keyword Guide to Porting
Trigger Function F-2
Single Trigger (SRQ) F-2
Single Trigger Mode (SING) F-2
Calibration F-2
Calibration Measurement (OPC) F-2
Calibration Done (OPC) F-2
Data Transfer Command F-2
Binary Transfer Commands (FORM2, FORM3, FORM4) F-2
Meas/Format F-3
Input Port (AR, MEASA, MEASR) F-3
Format (LOGM, PHAS, DELA, LINM, EXPP, LOGMP, LOGMD, REAL, IMAG, POLAR) F-3
Impedance Conversion Function (CONVZTRA, CONVZREF, CONVYTRA, CONVYREF) F-4
Analysis Commands F-5
BIN sort (BINSLINE, BINS, BINOA, BINOB, EDITBINL) F-5
Limit Line Edit (EDITLIS1, EDITLIS2) F-5
Other Commands F-5
KEY command F-5

G. Option 022
Overview G-1
Using E5100A/B Option 022 . . . . G-2
Required Parameters G-4
The Sample Program List G-5
GPIB Commands of Option 022 . . . . G-9
BINSIZE␣<*value*> G-9
INPUTRAC␣<*value1*>,<*value2*>,<*value3*> G-9
INPUTRACB␣<*value1*>,<*value2*>,<*value3*> G-9
MEAS1PT? <*value*> G-10
OUTPTRAC?␣<value> G-10
OUTPTRACB?␣<value1>,<value2> G-11
TIMO␣{ON|OFF|0|1} G-11
TOTIME␣<value> G-11

TRABGE G-11
TRABLE G-12
TRAFDATA G-12
TRAFMEMO G-12
TRAP⊔{OFF|ON|0|1} G-13
TRAR⊔<*value1*>,<*value2*> G-13
WRIT16 . . . . G-13
WRIT24 . . . . G-14

H. Option 023
Overview H-1
Measuring Drive Level Characteristics using Phase Tracking Function H-2
1. Measurement Settings for Phase Tracking H-2
2. Setting for Using π Network Test Fixture H-3
3. Settings for the Drive Level Measurement H-3
4. Settings Parameters for Tracking Function H-4
5. Searching for Resonant Point H-4
6. Tracking Measurement H-4
7. The Measurement Result H-4
Sample Program List H-5
Measuring Aging Characteristics H-9
Compensation of Sweep Time for Aging Characteristics H-9
Sub-routine for Measuring Aging Characteristics H-9
Notes on the Phase Tracking H-9
When the Phase Tracking Fails H-9
Limitations on the Phase Tracking H-10
GPIB Commands for Option 023 . . . . H-10
FMT⊔MAGZDF H-10
PTABORT⊔{ON|OFF} H-10
PTFOVHD⊔<*value*> H-10
PTFR⊔<*value*> H-11
PTFRSR⊔<*value*> H-11
PTPARA⊔<*value*> H-11
PTRACK⊔{OFF|ON} H-11
PTREPN⊔<*value*> H-12
PTSTAT? H-12
PTTRGLMT⊔<*value*> H-12
PTTRGPHS⊔<*value*> H-13
SRCHFR?⊔{1|2|3|4|5|6},<*value*> H-13

I. Partial Sweep
Commands for the partial sweep function. I-1
Sample Program I-1

J. File Transfer Function
Overview J-1
Commands for the File Transfer Function. J-2
Sample Programs J-3
File Transfer from E5100A/B to External Controller J-3
File Transfer from External Controller to E5100A/B J-5
Displaying List of Files in Current Directory J-8

K. Measuring Load Resonance Frequency (FL) Using a CL Board with Fixed C
Introduction K-1
Basic Program Flow K-2
Commands K-3
Program Listing K-4

Index

Figures

1-1. GPIB Device and Address . . . 1-5
3-1. Simplified Data Processing Flow 1 . . . 3-2
3-2. Simplified Data Processing Flow 2 . . . 3-3
3-3. Simplified Internal Process of ASCII and Binary Transfer . . . 3-7
3-4. IEEE 64-bit floating point format . . . 3-8
3-5. IEEE 32-bit floating point format . . . 3-9
3-6. Binary Data Header . . . 3-9
4-1. The I/O Port Pin Layout . . . 4-1
4-2. INPUT1, OUTPUT1, OUTPUT2 Timing Chart . . . 4-4
5-1. General Status Register Model . . . 5-1
5-2. Transition Filter and Condition Register . . . 5-3
5-3. Status Reporting Structure . . . 5-4
5-4. Example of Reading Status Byte (1) . . . 5-5
5-5. Example of Reading Status Byte (2) . . . 5-5
5-6. Example of Generating a Service ReQuest (SRQ) . . . 5-7
5-7. SRQ Generation Sequence . . . 5-11
6-1. Controlling Instrument BASIC from an External Controller (The Control Flow) . 6-9
6-2. Parallel Processing . . . 6-19
8-1. The Data Output Timing Chart . . . 8-36
A-1. Serial Number Plate (Sample) . . . A-2
C-1. Form 2 Data Transfer Format . . . C-1
C-2. Form 3 Data Transfer Format . . . C-1
C-3. Data Processing Flow Diagram . . . C-3
D-1. THRR . . . D-6
D-2. RPLPP? . . . D-13
D-3. RPLHEI? . . . D-14
D-4. RPLRHEI? . . . D-15
D-5. RPLLHEI? . . . D-16
D-6. RPLENV? . . . D-17
D-7. RPLMEA? . . . D-18
D-8. RPLVAL? . . . D-20
D-9. POLE? . . . D-21
D-10. OUTPFILT? . . . D-23
D-11. OUTPXFIL? . . . D-26
D-12. OUTPCFIL? . . . D-31
D-13. OUTPRESO? . . . D-36
D-14. OUTPRESR? . . . D-38
D-15. OUTPRESF? . . . D-40
D-16. OUTPCERR? . . . D-42
D-17. Six-Device Equivalent Circuit of Crystal Resonator . . . D-44
D-18. Four-Device Equivalent Circuit of Crystal Resonator . . . D-50
D-19. Admittance Characteristic Circle Diagram . . . D-51
G-1. The Trap Function . . . G-1
G-2. How E5100A/B option 022 Works with the Default Setup . . . G-2
G-3. The Displayed Measurement Result . . . G-4
G-4. The Data Output Timing Chart . . . G-10

H-1. Measuring Drive Level Characteristics H-1
H-2. The Aging Characteristic Measurement H-2
H-3. An Example of the Measurement Result H-3
H-4. A Displayed Result H-5
I-1. Sample Program : Partial Sweep I-1
J-1. Procedure of executing commands to read/write data J-2
J-2. Fixed length block format J-3
K-1. . . . . K-2

Tables

2-1. Setting the Active Channel 2-3
2-2. Setting the Measurement and Format 2-3
2-3. Setting the Sweep 2-4
2-4. Calibration 2-4
5-1. Status Bit Definitions of the Status Byte (STB) 5-5
5-2. Status Bit Definitions of the Event Status Register (ESR) 5-6
5-3. Status Bit Definitions of the Event Status Register B (ESB) 5-6
5-4. Status Bit Definitions of the Operational Status Register (OSR) 5-7
A-1. Manual Changes by Serial Number A-1
A-2. Manual Changes by Firmware Version A-1
C-1. Data Array Output Commands C-3
C-2. Data Array Input Commands C-3
F-1. . . . . F-3
F-2. . . . . F-4
F-3. . . . . F-4
K-1. Commands K-3

1

Introduction

How to Use This Manual

This manual introduces GPIB programming for the E5100A/B. It provides additional information on how to write programs that might be difficult to understand when using only the GPIB Command Reference. It also provides information, techniques, and examples of how to effectively control GPIB instruments.

To use this manual effectively, you need one of the following GPIB controllers:

- HP Instrument BASIC (The E5100A/B has this function unless Option UKR Delete HP Instrument BASIC is installed). That is an internal GPIB controller in the E5100A/B.
- An HP 9000 Series 200 or 300 computer that has HP BASIC 3.0 or later. The HP BASIC 3.0 must have the GPIB, GRAPH, IO, KBD, and ERR binaries.

This manual helps you to learn how to write programs that control the E5100A/B. To help you learn quickly, many sample modules and programs are provided.

Target Reader

The target reader of this manual is a programmer who wants to control the E5100A/B through the GPIB interface.

This manual explains GPIB programming using HP BASIC. Therefore, you should have some experience using BASIC. If you have never written a program in BASIC, review the applicable documentation listed at the end of this chapter before starting this manual. This manual does not require extensive knowledge of BASIC programming.

This manual assumes you understand the operations and features of the E5100A/B. If you have never operated the E5100A/B, read the *User's Guide* to learn how to operate the E5100A/B.

What's in This Manual?

The following chapters are provided in this manual:

- Chapter 1 "Introduction" provides an introduction to this manual, how to use a sample program, an GPIB overview, hardware preparation, and a description of the sample program disk. This chapter provides important information that is used throughout this manual. You should read this chapter first.
- Chapter 2 "Setup and Measurement Program" provides GPIB command basics. It also shows how to build a measurement program including setups, compensating, triggering, and getting data. If you want to build an automated measurement program, read this chapter.
- Chapter 3 "Data Processing and Transfer" shows the data processing flow and the arrays of the analyzer, describes how to access an internal data array (including trace data or calibration data). If you want to get measured trace data from the analyzer, read this chapter.

- Chapter 5 “Using Status Reporting Function” describes the status reporting system of the analyzer and how to use it. This chapter also describes an SRQ interrupt. If you want to obtain the analyzer’s status using a BASIC program, read this chapter.
- Chapter 4 “Using the I/O Port” provides information on how to use the I/O port on the rear panel. If you want to use the I/O port for communicating with an external instrument (such as a handler), read this chapter.
- Chapter 6 “Programming Miscellaneous” provides information not directly concerned with measurements, but useful for programming. This includes accessing the disk, controlling Instrument BASIC, or debugging a program.
- Chapter 7 “Sample Programs” describes sample programs for some applications.
- Chapter 8 “Command Reference” provides a summary of all available GPIB commands.
- Appendix A “Manual Changes” shows revision information for this manual.

How to Use the Program Modules

This manual provides many sample program modules that are not in a complete program style. You can easily understand the module’s objective because the program module does not include unnecessary code. You can use these modules to build your own program by combining them.

The program modules are provided in the following style and typeface:

```
SAMPLE CODE   This is a comment for the sample code.
```

As shown in the example above, a module has no line number, no initializing steps, and no `END` statement. All these are required for an executable BASIC program.

Building a Working Program Using Program Modules

To make a program that uses sample program modules, perform the following steps:

1. Add an initializing module at the beginning of your program.
2. Arrange the program modules.
3. Add an `END` statement on the last line executed by your program.

The line numbers are added automatically by the BASIC editor.

Initializing Module

The initializing module defines a hardware identifier as a variable to eliminate the difference between Instrument BASIC and HP BASIC. Usually, you can use the same program for Instrument BASIC and HP BASIC by changing the initializing module. The initializing module also initializes the GPIB.

The following are typical initializing modules for a program:

`ASSIGN @E5100 TO 800`	*Assigning GPIB address to 800.*
`Scode=8`	*Assigning interface select code to 8.*
`ABORT Scode`	*Get active control.*
`CLEAR @E5100`	*Preset the interface.*
`OUTPUT @E5100;"DISAHIHB"`	*Set display mode.*

Module 1-1. Instrument BASIC Initialization

`ASSIGN @E5100 TO 717`	*Assigning GPIB address to 717.*
`Scode=7`	*Assigning interface select code to 7.*
`ABORT Scode`	*Get active control.*
`CLEAR @E5100`	*Preset the interface.*

Module 1-2. External Controller Initialization

Each module of this manual assumes that one of the initializing modules exists at the beginning of the program, and uses the following variables without additional explanation:

`@E5100`
: Represents the device selector of the E5100A/B. 800 is for Instrument BASIC and 717 is for the external controller.

`Scode`
: Represents the interface select code to which the E5100A/B is connected. 8 is for Instrument BASIC and 7 is for the external controller.

Example

For example, a complete program using Module 2-2 in Chapter 2 and Instrument BASIC, is shown below:

```
10  ASSIGN @E5100 TO 800       ! Module 1-1
20  Scode=8                     !
30  ABORT Scode                 !
40  CLEAR @E5100               !
50  OUTPUT @E5100;"DISAHIHB"   !
60  !
70  OUTPUT @E5100;"PRES;CHAN2" ! Module 2-2
80  !
90  END
```

GPIB Overview

The GPIB is a general purpose digital interface system that is used to integrate the controller, measurement instruments, and peripherals into a system.

Controller

The controller is a device that can address an GPIB device to talk (output data) or listen (receive data).

The active controller can control the other devices on the bus at that time (when multiple controllers are connected). Only one controller can be active at a time. The active controller can pass control to another controller by using the `PASS CONTROL` command.

Only one controller can be the system controller on the same bus. The system controller is the active controller when the system is turned on. When another controller is the active controller, the system controller can become the active controller at any time by executing `ABORT` *select-code*.

Device Selector

GPIB device control is accomplished by sending commands from the active controller. The active controller can select the target device for the commands by specifying the device selector.

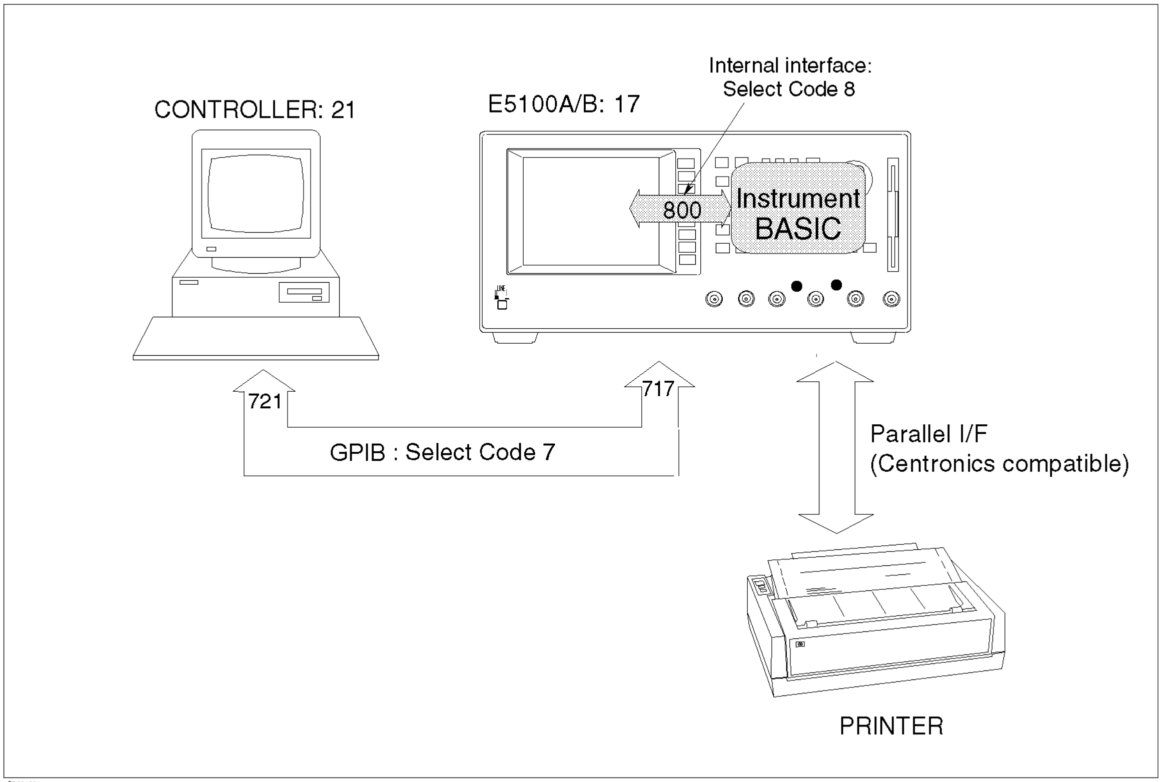


Figure 1-1. GPIB Device and Address

Figure 1-1 shows the relationship between the GPIB address and the device selector. For example, the device selector of the E5100A/B on GPIB with an address of "17," is "717" on the GPIB.

HP Instrument BASIC is connected in the E5100A/B internally by the internal interface. The interface select code of the internal interface is "8" to distinguish it from the external select code of "7."

You can use any address from "00" to "30" to specify the internally connected analyzer from Instrument BASIC, because only the analyzer is connected to the internal interface. This manual uses address "00," thus the device selector is "800."

Preparation for Operation

This section describes GPIB cable connection and address setting for GPIB control. You can choose one of the following two methods to control the analyzer:

- Using the Instrument BASIC controller (except for Option UKR).
- Using an external controller.

In both cases, you can use the following procedure to prepare the controller and the analyzer:

1. Connect the GPIB cables.
2. Set the GPIB Address.
3. Prepare the controller for use.

Using Instrument BASIC as the Controller (except for Option UKR)

The HP Instrument BASIC system is a BASIC controller that is built into the analyzer. HP Instrument BASIC is a subset of HP BASIC.

You can control the E5100A/B internally by using HP Instrument BASIC. HP Instrument BASIC has the capability to be a system controller. The other GPIB instruments are also controllable through the GPIB connector that is located on the rear panel of the E5100A/B.

1. Connecting the GPIB Cables For Instrument BASIC

A connection between the analyzer and Instrument BASIC is not required because they are already connected via the internal interface in the analyzer. See Figure 1-1. However, if any other GPIB instruments must be connected, see "Connecting External GPIB Cables" on the next page.

2. Setting the GPIB Address For Instrument BASIC

This is not required because the front panel setting of the GPIB address does not affect the internal interface. You can use any address to specify the analyzer via the internal interface as described in "Device Selector".

3. Setting Up Instrument BASIC

To set up Instrument BASIC, perform the following steps:

1. Connect the DIN keyboard to the rear panel connector. (See the *User's Guide*.)
2. Turn the analyzer power on.
3. To allocate the Instrument BASIC output area on the LCD display for the `PRINT` statement, press (Display) `DISPLAY ALLOCATION` `HALF INSTR HALF BASIC`.
4. Press (System) `I-BASIC` `EDIT` to open the Instrument BASIC editor.

For more information on how to use the Instrument BASIC editor, see the *Using HP Instrument BASIC*.

If you connect the keyboard after turning on the analyzer, press (Preset) to enable key inputs.

Using an External Controller

You can use an external controller to control the analyzer by using the GPIB interface on the rear panel. Connect the controller and the analyzer using an GPIB cable.

1. Connecting External GPIB Cables

Connect the analyzer and external instruments with GPIB cables. The GPIB connector is on the rear panel of the analyzer.

The rules for connecting GPIB cables are as follows:

- The total number of GPIB devices can be up to 15 instruments.
- The total length of all the cables used can be up to 20m or 2m for each instrument.

You can connect the GPIB cables in any configuration (linear, star, or combination), as long as the above rules are satisfied.

Note Do not use a screwdriver when connecting the GPIB cables. The screwdriver slots in the lock screws are provided for REMOVAL only.

---

2. Setting the GPIB Address

The analyzer has no hardware switch for setting the GPIB address. You can only set it by front panel operation. To change the GPIB address of the analyzer, perform the following steps:

1. Press (System) `GPIB`.
2. Press `ADDRESS: E5100A` or `ADDRESS: E5100B`.
3. Enter the new address by using the numerical keys. (Avoid duplication with the GPIB address of the external controller.)
4. Press `x1` to complete the operation.

When you want to control another E5100A/B, change the GPIB address to avoid duplication of addresses on the same bus.

When the analyzer receives any GPIB command from an external controller, `Local` is appeared on the display to indicate the analyzer is in the remote mode. In remote mode, front panel key operation is disabled. To cancel the remote mode, press the `Local`.

3. Preparing For HP BASIC Operation

To prepare HP BASIC for operation, see your HP BASIC system manual.

Sample Program Disk

The sample programs (not the program modules) in this manual are included on the furnished *Sample Program Disk* (Agilent Part Number E5100-180X0). Furnished with special sample program disk (Agilent Part Number E5100-180X1) with original ones if Option 022/023 is designated. The number indicated by "x" in the part number, is incremented by 1 each time a revision is made.

Loading a Program from Disk

To use an IBASIC sample program, load it into Instrument BASIC and then run it.

1. Insert the *Sample Program Disk* into the internal disk drive that is below the display.
2. Press (Display) DISPLAY ALLOCATION BASIC STATUS.
3. MSI ":INTERNAL,4"
4. Type, GET "*filename*" (Return).

To use an external controller sample program, load it into an external controller and run it.

1. Set up HP BASIC on your external controller.
2. Insert the *Sample Program Disk* into the disk drive of the controller.
3. Set the current directory to the disk drive.
4. Type, GET "*finename*" (Return).

The applicable *filename* is printed in front of the sample programs in this manual.

Reading the Sample Program Disk on a PC

Because the sample program disk is provided in the LIF (Logical Interchange Format) that is used on an HP controller, you cannot read this disk on a PC. If you want to convert the program into DOS format that can be read by a PC, perform the following steps:

1. Prepare a DOS formatted disk (3.5 inch).
2. Load the program into the Instrument BASIC from the sample program disk. See "Loading a Program from Disk" for the procedure.
3. Remove the sample program disk, then insert the DOS formatted disk.
4. Type, SAVE "*filename*" (Return).

The analyzer supports either LIF or DOS format, and automatically detects which format disk is used.

Note You cannot initialize a disk in the LIF format with the E5100A/B.

Related Documentation Information

You can obtain more detailed information than provided by this manual by referring to the following documents.

The following manuals are provided with the E5100A/B.

- *E5100A/B Function Reference* explains all functions accessed from the front panel key.
- *E5100A/B User's Guide* to learn about the analyzer itself and its front panel key operation.

The following manual is also provided with the E5100A/B unless Option UKR (Delete HP Instrument BASIC) is installed.

- *HP Instrument BASIC User's Handbook Supplement* for Instrument BASIC information.

The following documents also provide related information:

- *HP BASIC Programming Guide* for learning HP BASIC programming. (Furnished with the HP BASIC system.)
- *Tutorial Description of the General Purpose Interface Bus* for an overview of the GPIB and IEEE 488 standard (Agilent literature no. 5952-0156).

2

Setup and Measurement Program

This chapter describes a basic measurement program that include setups, user calibration, triggering, and getting data. This chapter discusses the following topics:

- Overview of GPIB Control.
- Automating a Measurement Procedure.
- Sample Program: Basic Measurement Program.

Overview of GPIB Control

Before starting to program, you should know how to send an GPIB command to the analyzer. This section describes the following items:

- Sending GPIB commands.
- Sending a query and reading the response.

If you have experience programming GPIB instruments, you can skip this section and go to “Automating a Measurement Procedure”.

Sending GPIB Commands

HP BASIC and Instrument BASIC use the **OUTPUT** statement to send GPIB commands that control GPIB devices. For example:

```
OUTPUT @E5100;"PRES"    Presets the analyzer.
```

Module 2-1. Presetting the Analyzer

This line sends the GPIB command within the double quotes (**PRES**) to the GPIB device at address `@E5100`. This command presets the analyzer. This is equivalent to pressing the (Preset) key.

You can send multiple commands in a single line by separating each GPIB command with a semicolon (;).

```
OUTPUT @E5100;"PRES;DISAHIHB"    Sends PRES and DISAHIHB by single line.
```

Module 2-2. Sending Multiple Commands in a Line

Sending a Query and Reading the Response

There are commands that return a response after being sent. These commands are called queries. A query has a question mark (**?**) at the end of the command.

You can retrieve the response by using the `ENTER` statement as shown below:

```
OUTPUT @E5100;"CENT?"   This line queries center frequency setting.
ENTER @E5100;Center     This line retrieves the return value.
```

Module 2-3. Querying Center Frequency Setting

You must retrieve the response into the correct type variable. In the example above, the query returns a value, depending on the current center frequency setting. Therefore, the second line retrieves the response into a numeric type variable (**Center**).

The response data type, numeric or string, for each command is shown in the reference section of the *GPIB Command Reference*.

Automating a Measurement Procedure

This section describes the sample program modules and equivalent commands for setting up the analyzer using the following functions:

1. Setting the Active Channel
2. Setting the Measurement and Format
3. Sweep Setup
4. Calibration
5. Device Connection
6. Triggering a Measurement
7. Data Processing and Transfer
8. Exiting the Program

1. Setting the Active Channel

To begin setting the Measurement, Format, Calibration, Display, and Marker functions of the analyzer, specify the active channel first because it affects all these settings.

Table 2-1. Setting the Active Channel

Setups	Key Operations	GPIB Commands
Active channel 1	(Meas/Format) ACTIVE CH [CH2] [1]	CHAN1

1: Press this softkey to alter the label to [CH1] when the label is [CH2].

The equivalent program is as follows:

```
OUTPUT @E5100;"CHAN1"    Sets the active channel to channel 1.
```

Module 2-4. Setting the Active Channel

2. Setting the Measurement and Format

Table 2-2. Setting the Measurement and Format

Setups	Key Operations	GPIB Commands
Gain-Phase Mode	(Meas/Format) FUNCTION[ ] GAIN-PHASE	ANAMODE GAINP
A/R Measurement	Meas A/R	MEAS AR
LOG MAG Format	FORMAT LOG MAG	FMT LOGM

The equivalent program for setting Active Channel, Measurement, and Format is as follows:

OUTPUT @E5100;"CHAN1;MEAS AR;FMT LOGM"	*Sets the Active Channel to 1, the Measurement to A/R, and the Format to LOG MAG.*

Module 2-5. Active Channel/Measurement/Format Setup

3. Sweep Setup

Table 2-3. Setting the Sweep

Setups	Key Operations	GPIB Commands
Center Frequency to 70 MHz	(CENTER) 70 × M	CENT 70MHZ
Frequency Span to 100 kHz	(SPAN) 100 × k	SPAN 100KHZ

The equivalent program is as follows:

OUTPUT @E5100;"CENT 70MHZ"	*Sets the center frequency to 70 MHz.*
OUTPUT @E5100;"SPAN 100KHZ"	*Sets the span to 100 kHz.*

Module 2-6. Sweep Setup

Note

You must be careful with the order of commands when you execute them. The E5100A/B has 2 pairs of channel settings and 2 stimulus settings, and they can be set up separately. A setup is applied to the currently selected active channel or active stimulus. You must, therefore, set up the active channel/stimulus before performing other setups.

4. Calibration

The key operations and corresponding GPIB commands are described below.

Table 2-4. Calibration

Setups	Key Operations	GPIB Commands
Response Calibration Startup	(CAL) RESPONSE	CALI RESP
THRU Measurement	THRU	STANC
Completion of Calibration	DONE:RESPONSE	RESPDONE

As an example, let's look at a frequency response calibration procedure and its corresponding GPIB commands.

As described in the setup process, you must execute GPIB commands in the correct order using **OUTPUT** statements to perform calibration. In addition, there is another important consideration: You must pause the program when the E5100A/B is measuring the calibration standard or is calculating the calibration coefficient.

The reason that you must pause the program is that the measurement modules of the E5100A/B and Instrument BASIC process data independently. Instrument BASIC, therefore, would try to execute the next command before the instrument measurement module completes its job. This can occur when the measurement module is processing a time consuming job (for example, a sweep or a calibration coefficient calculation). The measurement module cannot process an GPIB command if the module receives it during a process. Therefore, you must synchronize the measurement module and the process used by the controller to execute GPIB commands successfully. The following modules show a way to synchronize these two procedures.

Calibration Process

1. Setup the calibration mode.

`OUTPUT @E5100;"CALI RESP"`	*Sets the calibration mode to Response.*

Module 2-7. Calibration Mode Setup

2. Connect a standard and perform a measurement.

   You can use the `STANC?` Query to make a calibration measurement and to check to check for the completion of the sweep because the sweep process requires extra time. When you execute **STANC?**, it returns `1` as soon as the GPIB command executed immediately before it is completed. The program does not proceed to the next step until the `ENTER` statement receives the return value from the Query.

`INPUT "CONNECT THRU, THEN PRESS ANY KEY",A$`	*Shows the prompt message and waits for the key entry.*
`OUTPUT @E5100;"STANC?"`	*Performs a measurement.*
`ENTER @E5100Tmp`	

Module 2-8. Connection and Measurement

3. The calibration coefficient is calculated when the measurement is completed. You can also pause the program during a calibration coefficient calculation. `RESPDONE?` is used to monitor for the completion of the calculation.

`OUTPUT @E5100;"RESPDONE?"`	*Performs the calculation of calibration coefficient.*
`ENTER @E5100;Tmp`	

Module 2-9. Calculation of the Calibration Coefficient

The calibration is now complete and the E5100A/B is ready to make a measurement.

5. Device Connection

You must tell the operator to connect a device. This statement uses **INPUT** to pause the program until data is entered from the keyboard.

`INPUT "CONNECT DUT, THEN PRESS [RETURN]",A$`	*Displays a message and waits for the data entry*

Module 2-10. Device Connection

The program starts a measurement when a device is connected. You need to pause the program, as you needed to do for a calibration, until the measurement completes.

6. Triggering a Measurement

If the measurement sweeps only once, you can omit the process to monitor for the completion of the sweep by executing the following line:

```
EXECUTE "SING"    Execute a single measurement.
```

Module 2-11. Triggering a Measurement

When the **SING** command is executed by the **EXECUTE** statement, the program pauses until the sweep completes.

Note You cannot use this method from an external controller because only the E5100A/B Instrument BASIC supports the **EXECUTE** command. You must use the same method as you do for the multiple sweep to synchronize the completion of sweep and the program.

When You are Using an External Controller

When you are using an external controller, you can use the ***SING?** Query to make a single measurement and check the completion of a sweep.

```
OUTPUT @E5100;"SING?"
ENTER @E5100;Tmp
```

Module 2-12. Triggering Module For an External Controller

7. Data Processing and Transfer

You can use a Query command to extract the measurement data required.

The following process executes the command **OUTPMAX?** (which returns the maximum value within the display range).

```
OUTPUT @E5100;"OUTPMAX?"            A Query to return the maximum value.
ENTER @E5100;Value
PRINT "Maximum value:",Value;"dB"   Display the return value.
```

Module 2-13. Data Processing and Transfer

8. Exiting the Program

Exit the program. You can also repeat a measurement using the **GOTO** statement.

```
END    Exits the program.
```

Module 2-14. Exiting the Program

Sample Program -1: Basic Measurement

The following is a list of the program described in this section. You can refer to the this program to check the explanations.

Disk

This program is contained in the attached sample disk with the file name **BASIC** (For Instrument BASIC). The program for an external controller has the name **BASIC_E**.

```
110   ASSIGN @E5100 TO 800          ! IBASIC INITIALIZATION
120   Scode=8                       ! [MODULE 1-1]
130   CLEAR @E5100                  !
140   OUTPUT @E5100;"DISAHIHB"      !
150   !
160   OUTPUT @E5100;"PRES"          ! ANALYZER PRESETTING
170                                 ! [MODULE 2-1]
180   !
190   OUTPUT @E5100;"CHAN1;ANAMODE GAINP;MEAS AR;FMT LOGM"
                                    ! CHAN/MEAS/FORMAT SETUP
200                                 ! [MODULE 2-5]
210   !
220   OUTPUT @E5100;"CENT 70MHZ"    ! SWEEP SETUP
230   OUTPUT @E5100;"SPAN 100KHZ"   ! [MODULE 2-6]
240   !
250   OUTPUT @E5100;"CALI RESP"     ! CAL MODE SETUP
260                                 ! [MODULE 2-7]
270   !
280   INPUT "CONNECT THRU, THEN PRESS [RETURN].",Dum$
                                    ! CONNECTION & CAL MEASUREMENT
290   OUTPUT @E5100;"STANC?"        ! [MODULE 2-8]
300   ENTER @E5100;Tmp              !
310   !
340   OUTPUT @E5100;"RESPDONE?"     ! CALCULATION OF CAL COEFFICIENT
350   ENTER @E5100;Tmp              ! [MODULE 2-9]
360   !
400   DISP "RESPONSE CAL COMPLETED" ! DISPLAY MESSAGE
410   !
420   INPUT "CONNECT DUT, THEN PRESS [RETURN].",Dum$
                                    ! PROMPT DEVICE CONNECTION
430                                 ! [MODULE 2-12]
440   !
450   EXECUTE "SING"                ! TRIGGERING MEASUREMENT
460                                 ! [MODULE 2-13]
470   !
480   OUTPUT @E5100;"AUTO"          ! AUTO SCALING
490   !
500   OUTPUT @E5100;"OUTPMAX?"      ! DATA PROCESSING & TRANSFER
510   ENTER @E5100;Value            ! [MODULE 2-14]
520   PRINT "MAXIMUM VALUE:",Value,"dB"
530   !
540   END                           ! EXITING PROGRAM
550                                 ! [MODULE 2-15]
```

Column Parameters and Variables

As explained in the previous section, an GPIB command that requires a parameter is executed as follows:

```
OUTPUT @E5100;"CENT 70MHZ"    Sets the center frequency to 70 MHz.
```

Module 2-15. Command with Fixed Value

When a command has a specified suffix, the suffix multiplier and suffix units can be used with parameters as follows (the suffix multiplier must be used with the suffix unit):

Parameter	Suffix Unit	Available Multipliers
Frequency	`HZ` (Hz)	`G` : G (giga; $\times 10^9$)
Power	`DBM` (dBm)	`MA` : M (mega; $\times 10^6$)
Voltage	`V` (Volt)	`K` : k (kilo; $\times 10^3$)
Current	`A` (Ampere)	`M` : m (milli; $\times 10^{-3}$)
Impedance[1]	`OHM` (Ω)	`U` : μ (micro; $\times 10^{-6}$)
Admittance[2]	`SIE` (Siemens)	`N` : n (nano; $\times 10^{-9}$)
Inductance	`H` (henry)	`P` : p (pico; $\times 10^{-12}$)
Capacitance	`F` (farad)	`F` : f (femto; $\times 10^{-15}$)
Time	`S` (second)	
Phase	`DEG` (°; default), `RAD` (radian)	
distance	`M` (meter)	

1 Resistance and reactance are the same

2 Conductance or susceptance are the same

The suffix is optional and can be omitted. If you omit it, the analyzer assumes that the default suffix is sent. You can use a variable to change the parameter depending on the situation.

```
OUTPUT @E5100;"CENT ";Center    Sets the center frequency to Center.
```

Module 2-16. Command with Valuable

This statement passes the variable **Center** to the GPIB command **CENT** as a parameter. You can place a parameter outside of double quotes and separate them with a semicolon to use a variable as a parameter. (You must always insert a space after a command.)

The following shows an GPIB command that takes a character string parameter.

```
OUTPUT @E5100;"SAVDGRA ""filename"""    Saves the graphic image as the filename.
```

Module 2-17. Command with Character String Parameter

Character strings are usually double quoted ("). In this case, however, you must double the double quotes ("") for the inside double quotes because an GPIB command itself is already double quoted.

You can also use single quotes(') instead of double double quotes("").

```
OUTPUT @E5100;"SAVDGRA 'filename'"
```

Module 2-18. Command with Character String Parameter

You can also use a variable as a string character parameter.

```
OUTPUT @E5100;"SAVDGRA '";filename$;"'"
```

Module 2-19. Command with Variable as String Character Parameter

This statement is basically the same as the numeric value variable example. You must, however, place a single quote before and after **filename$** because a string character must be single quoted.

To close a double double quote, you must use a quadruple double quote. You must use a double double quote ("") to close a single quote.

The following examples are both valid. (The command **PROG:EXEC**""**GET**. . . . . can be executed from an external controller only.)

```
OUTPUT @E5100;"PROG:EXEC ""GET """"Filename"""""""
```

Module 2-20. Command With Double Quotes

```
OUTPUT @E5100;"PROG:EXEC 'GET ""Filename""'"
```

Module 2-21. Command With Single Quotes

3

Data Processing and Transfer

Introduction

This chapter describes data processing and how to access the internal data arrays.

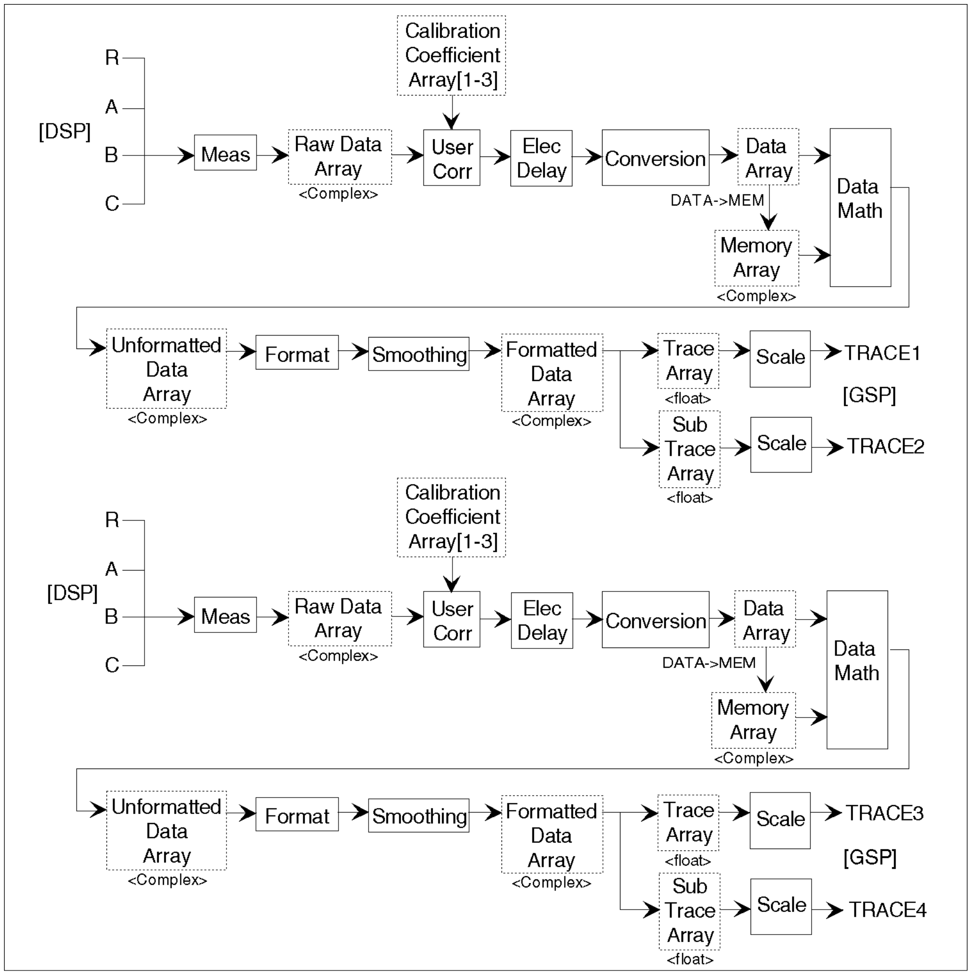
The following information is covered in this chapter:

- Data arrays.
- Data transfer method.
- Sample program: compensation data transfer.

Data Arrays

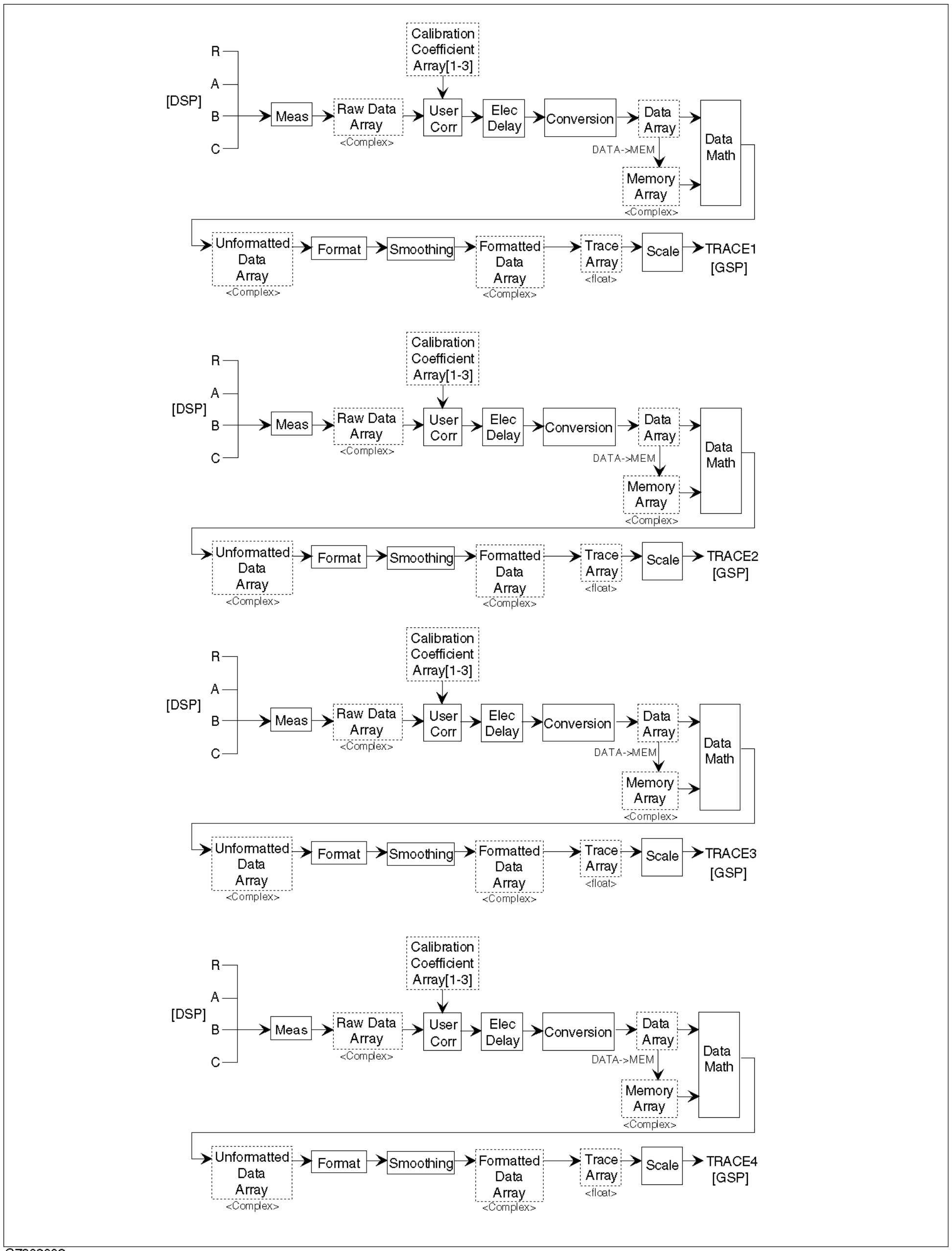
The analyzer has data arrays that contain measurement data, error correction data, and stimulus data. You can read or write data to these arrays using GPIB commands.

Figure 3-1 and Figure 3-2 show simplified diagrams of the data processing in the analyzer.



C7203007

Figure 3-1. Simplified Data Processing Flow 1



C7203008

Figure 3-2. Simplified Data Processing Flow 2

Dotted line boxes represent data arrays that hold intermediate or processed data. The following section describes each of these data arrays.

Raw Data Array

The raw data array stores the results of all the preceding data processing operations including the correction by calibration data. The data is in a complex form (real/imaginary pairs) and read out with the commands **OUTPRAW?**. When you want to use your own compensation method for a measurement data, ① take out the raw data from the raw data array (see Module 3-1), ② apply your compensation method to this data, ③ enter the compensated data into the data array (Module 3-2).

The following example module queries for raw data and retrieves it:

```
DIM Dat(1:201,1:2)           Define NOP×2 for receiving complex data.
OUTPUT @E5100;"OUTPRAW?"     Query raw data array.
ENTER @E5100;Dat(*)
```

Module 3-1. Getting Raw Data Array

- Related GPIB Commands

  The following command is used for sending data to the raw data array:

  INPURAW,*data* Sends data to the raw data array of the active channel.

Data Array

This is the raw data with error correction applied. The array is for the currently measured parameter, and is in a complex form (real/imaginary pairs). The data array data is read out with **OUTPDATA?** or **OUTPDATAP?**. The **OUTPMEMO?** or **OUTPMEMOP?** query reads the trace memory if available (which is also error corrected). Neither raw nor data array data reflect post-processing functions such as electrical delay offset or trace math.

The following example module sets data for the data array:

```
OUTPUT @E5100;"INPUDATA?";Dat(*)    Sets data to data array.
```

Module 3-2. Setting Data Array Data

- Related GPIB Commands

  The following command is used to query the data array data:

  OUTPDATA,*data* Queries data in data array of the active channel.

Unformatted Data Array

This is the array of the complex number pairs that will be formatted in the next stage. *The unformatted data cannot be read out.*

Formatted Data Array

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

This data array is generally the most useful because it is the same information as that seen on the display. When you want to use data with a selected parameter unit, use this array.

The following example module queries the data trace array and retrieves it:

```
DIM Dat(1:201,1:2)          Define NOP×2 for receiving complex data.
OUTPUT @E5100;"OUTPFORM?"   Query data trace array.
ENTER @E5100;Dat(*)         OUTPFORM? outputs data in a complex format.
```

Module 3-3. Getting Data Trace Array

- Related GPIB Commands

  The following commands are used for accessing the data trace array:

  OUTPFORM? Outputs data trace array of the active channel.

  OUTPRTMEM? Outputs memory trace array of the active channel.

  Using the following commands, you can access a data trace array for the inactive channel. These commands are useful to get data from both traces without altering the active channel.

  OUTPIFORM? Outputs data trace array of inactive channel.

  OUTPRITMEM? Outputs memory trace array of the inactive channel.

Calibration Coefficient Array

The results of a calibration are stored as arrays of calibration coefficients that are used by the error correction routines. Each array corresponds to a specific error term in the error model. The calibration coefficients are read out with **OUTPCALC**{**01**|**03**}**?**.

For detailed information about the calibration and error model, see the *Function Reference*.

Accessing Arrays

If you want to enter calibration data from the controller to the calibration coefficient array, the analyzer must have previously done the calibration to enable the calibration data. To do this, perform the following steps:

1. Execute a dummy calibration to validate the correction.
2. Send the new calibration coefficients.

Note that the calibration coefficients are complex data.

- Related GPIB Commands

  The following commands are used to access the calibration coefficient array:

  OUTPCALC{**01**|**02**|**03**}**?** Outputs the specified calibration coefficient array data of the active channel.

  INPUCALC{**01**|**02**|**03**} Sets the specified calibration coefficient array data of the active channel.

Arrays for Memory Trace

When you store the trace data into the trace memory by sending the **DATI** command or by pressing DATA → MEM, the data array data is stored into the memory array. Memory array data passes through the formatting process, and then is stored into the memory trace array that is being displayed on the LCD display. See Figure 3-1 and Figure 3-2.

Besides the 16-trace limitation, the number of traces that can be stored into trace memory at one time depends on the capacity of the system memory and the number of points in the traces. The analyzer always reserves memory for 3 traces. The remaining memory is shared with the Instrument BASIC graphics.

Accessing Memory Array

You can only read data for the memory array that is activated. The memory array is read only. The following example module reads data from the memory array:

```
DIM Dat(1:201,1:2)
OUTPUT @E5100;"OUTPTMEM?"   Queries an activated memory array.
ENTER @E5100;Dat(*)         Receives the query response.
```

Module 3-4. Getting Memory Array Data

Accessing Trace Array

You can read or write trace array data. The following example module sets data for the trace array. You have to execute a trace-to-memory store procedure before setting a memory to display:

```
OUTPUT @E5100;"DATI"        Store trace data into trace array memory to allocate a trace array area.
OUTPUT @E5100;"OUTPTMEM?"   Queries a trace array.
OUTPUT @E5100;Dat(*)        Receives the query response.
```

Module 3-5. Setting Trace Array Data

- Related GPIB Commands

  The following commands are used to query the contents of the memory trace array:

  OUTPTMEM? Outputs the memory trace of the active channel.

  OUTPITMEM? Outputs the memory trace of the inactive channel.

Data Transfer Methods

This section describes the data transfer methods. When you get or send the data array's data, there are two methods for data transfer: ASCII and binary.

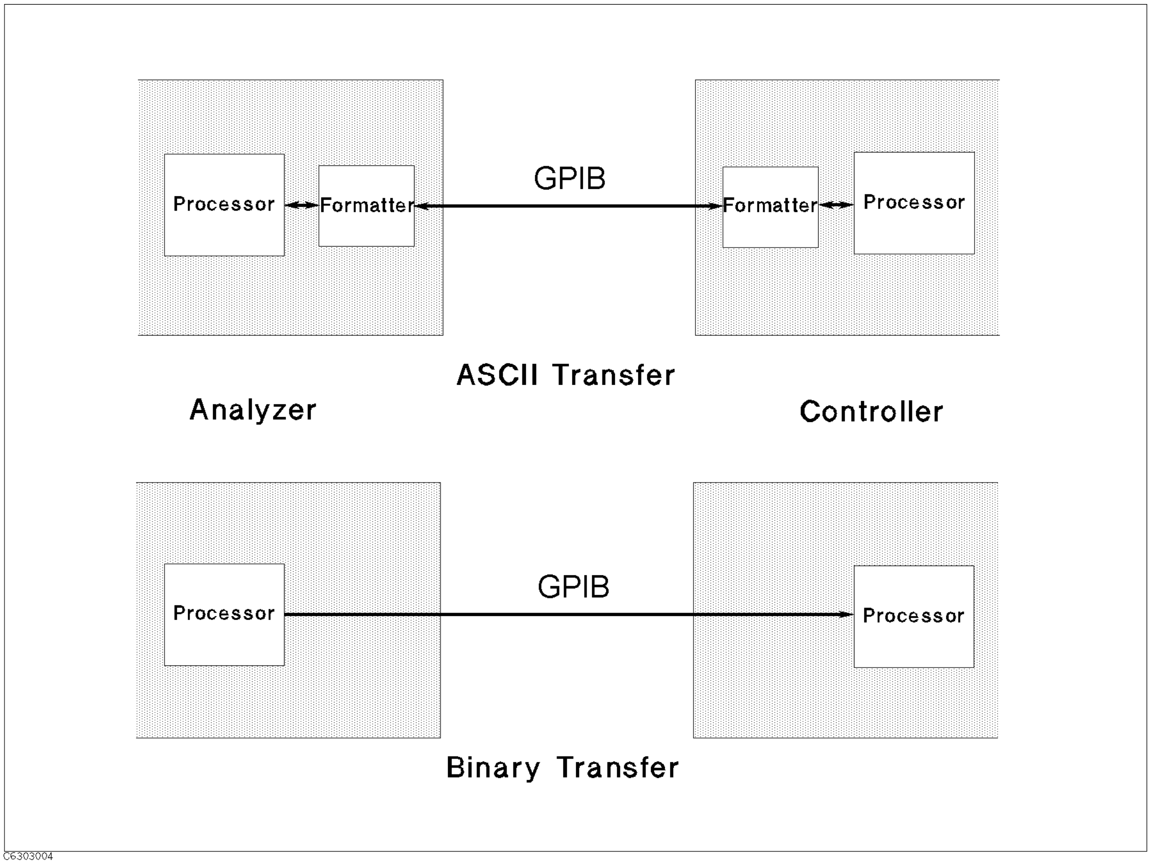


Figure 3-3. Simplified Internal Process of ASCII and Binary Transfer

Because the ASCII transfer passes through the formatted process, the program does not care about the data format. On the other hand, the binary transfer directly passes the data, but you have to indicate what data is transferred using the data header. The binary transfer is faster than the ASCII transfer.

ASCII Transfer

The ASCII format transfer is the easiest way to transfer array data between the analyzer and the controller. You do not have to worry about the data format because the analyzer and the controller automatically handles the formatting of the transferred data in this format. You can just send or retrieve array data by using the **OUTPUT** and **ENTER** statements.

The ASCII transfer format is sent as a 14-character (data) or 22-character (stimulus) string for each data point. This string includes a digit, sign or decimal point. Therefore, the data length of 201 points of complex data is 6030 bytes. (Including data delimiter "LF" for each data.)

To retrieve data from the analyzer using the ASCII format transfer, the following procedure is used:

1. Define a data array that is the same size as the data to be retrieved.

2. Specify the data transfer format is ASCII.
3. Send the data query command.
4. Retrieve the data.

```
DIM Dat(1:201,1:2)              Define the data array for receiving.
OUTPUT @E5100;"FORM4"           Specify the ASCII format.
OUTPUT @E5100;"OUTPDATA?"       Query the data trace.
ENTER @E5100;Dat(*)             Retrieve the data.
```

Module 3-6. Retrieving Data from the Analyzer Using ASCII Transfer

Sending data to the analyzer is easy, just specify a format, then send the data:

```
OUTPUT @E5100;"FORM4"                Specify the ASCII format.
OUTPUT @E5100;"INPUDATA ";Dat(*)     Send command and data.
```

Module 3-7. Sending Data to Analyzer by ASCII Transfer

Binary Transfer

For a faster data transfer, use the binary format. There are three formats for binary transfer. The following list shows the data format that the analyzer outputs when you query the data:

- IEEE 64-bit Floating Point Format

  Figure 3-4 shows the data transfer format of the IEEE 64-bit floating point format. Data is stored internally in the 200/300 series computer with the IEEE 64-bit floating point format, eliminating the need for any reformatting by the computer. In this mode, each number takes 8 bytes.

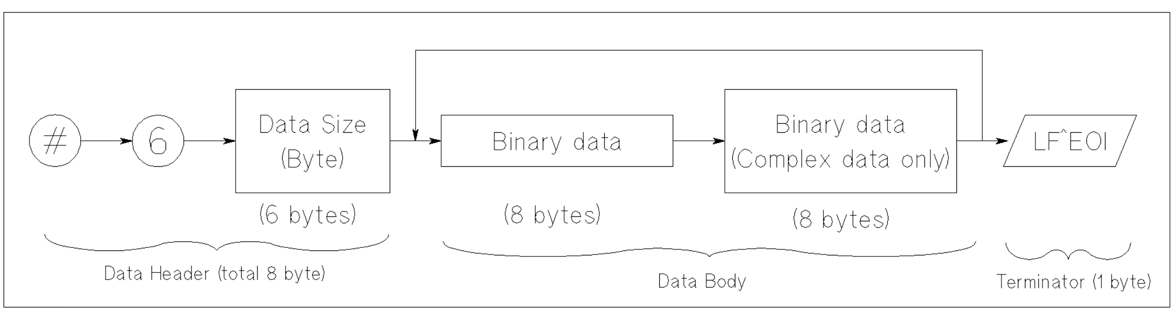


Figure 3-4. IEEE 64-bit floating point format

- IEEE 32-bit Floating Point Format

  Figure 3-5 shows the data transfer format the IEEE 32-bit floating point format. In this mode, each data point is 4 bytes. The difference from the 64-bit format is a significant digit. The 64-bit format has double the precision of this format.

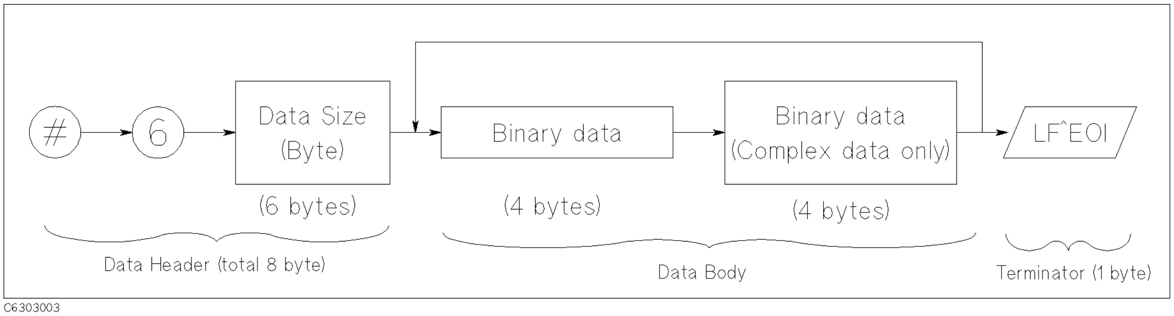


Figure 3-5. IEEE 32-bit floating point format

- MS-DOS® Format

  This mode is a modification of the IEEE 32-bit floating point format with the byte order reversed. The MS-DOS format also has a four-byte header that must be read to maintain the data order. In this mode, a PC can store the data internally without reformatting it.

Data Header

As shown in Figure 3-4 and Figure 3-5, the data header always precedes the data itself in binary format transfer. When you use a binary transfer, you must handle the data header with the data body.

When you query data in binary format, the analyzer outputs a fixed length (8 byte) data header. You can handle the data header as 8-byte strings for this purpose.

When you send the data to data array using binary transfer, you must prepare the data header for the data you send. The data header indicates the size of the transferred data. The data header consists of the following three parts: sharp (#), Number of bytes of "Data Size", and data size.

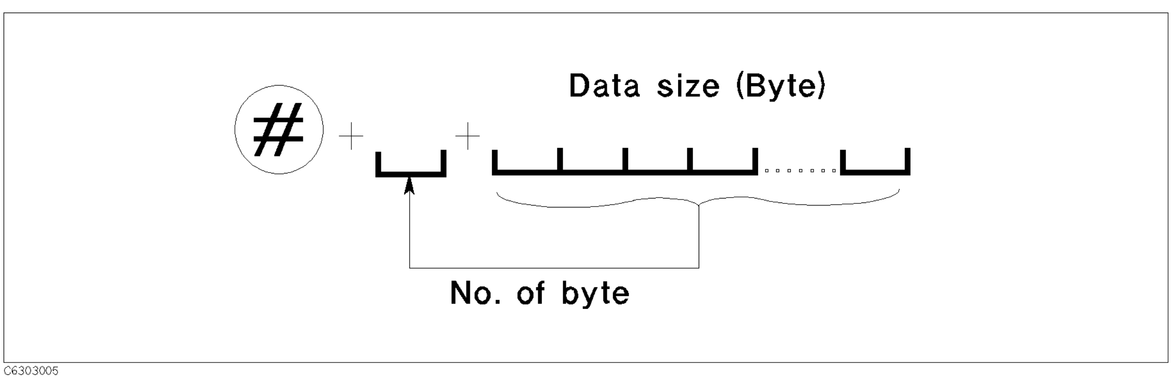


Figure 3-6. Binary Data Header

For example, the data size of 201 points of complex data in the 64-bit format is 3216 byte (=201×2×8). The "3216" is 4 digit (4 byte) number. Thus, the data header is "#43216". The queried data header that is generated from the analyzer is a fixed length header of 8 bytes that is obtained by adding "0" before "Data Size". For example, the data header above becomes "#6003216" as an 8-byte string. You can use either type of header to send data to the array.

Getting Data from the Analyzer

To get a data from the analyzer using the binary transfer method, the following procedure is used:

1. Assign a binary data path. (Specifying format off.)
2. Specify the data transfer format as binary.
3. Define a data array that is the same size as the data that will be retrieved.
4. Send the data query command.
5. Retrieve the data header.
6. Retrieve the data.
7. Retrieve the terminator.
8. Set the transfer format to ASCII mode if binary transfer is finished.

The binary data is sent in a mixed format of an ASCII header and a binary data body as shown in Figure 3-4 and Figure 3-5. To retrieve data correctly, you must retrieve the data header and data itself independently.

The following is a sample module for receiving data using the IEEE 64-bit format:

```
ASSIGN @Dt TO 800;FORMAT OFF          Binary path must turn off the formatting. Use 717 instead of 800
                                      for the external controller.
OUTPUT @E5100;"FORM3"                 Specify the format as IEEE 64-bit format.
DIM Dat(1:201,1:2)                    Assume that the receiving data size is 201×2.
OUTPUT @E5100;"OUTPDATA?"             Query the data trace array.
ENTER @E5100 USING "#,8A";Header$     Data header is 8 byte character.
ENTER @Dt;Dat(*)                      Receiving data via binary path.
ENTER @E5100;End$                     Reading terminator.
OUTPUT @E5100;"FORM4"                 Set ASCII mode if binary transfer is finished.
```

Module 3-8. Getting Data from Analyzer Using Binary Transfer

Note

Binary data transfer to the analyzer is not allowed. Use ASCII transfer for sending data to the analyzer.

- Related GPIB Commands

  The following commands are used to specify the data transfer format.

  FORM2 Selects IEEE 32-bit floating point format.

  FORM3 Selects IEEE 64-bit floating point format.

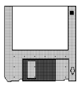
  FORM4 Selects ASCII format.

  FORM5 Selects MS-DOS format.

Sample Program -2: Binary Data Transfer

This is a sample program for the trace data transfer that uses the IEEE 64 bit floating decimal point format. This program transfers the data at each measured trace point in the binary format and displays them on the screen.

Disk



This program is contained in the attached sample disk with the file name **BINARY** (For Instrument BASIC). The program for an external controller has the name **BINARY_E**.

```
110  ASSIGN @E5100 TO 800;FORMAT ON! IBASIC INITIALIZATION
120  ASSIGN @Dt TO 800;FORMAT OFF !
130  Scode=8                      !
140  CLEAR @E5100                 !
150  !
160  DIM Dat(1:201),Stim(1:201)   ! VARIABLE DECLARATION
170  !
180  OUTPUT @E5100;"PRES"         ! MEASUREMENT SETUP
190  OUTPUT @E5100;"CHAN1;ANAMODE GAINP;MEAS AR;FMT LOGM"
200  INPUT "Enter center frequency (MHz)",F_cent
210  INPUT "Enter frequency span (kHz)",F_span
220  OUTPUT @E5100;"CENT ";F_cent*1.E+6
230  OUTPUT @E5100;"SPAN ";F_span*1000
240  !
250  EXECUTE "SING"               ! TRIGGERING MEASUREMENT
260                               ! [MODULE 2-12]
270  !
280  OUTPUT @E5100;"POIN?"        ! GETTING DATA USING BINARY TRANSFER
290  ENTER @E5100;Nop             !
300  OUTPUT @E5100;"FORM3"        !
310  OUTPUT @E5100;"OUTPRFORM?"   !
320  ENTER @E5100 USING "#,8A";Header$
330  ENTER @Dt;Dat(*)             !
340  ENTER @E5100 USING "#,1A";End$
350  !
360  OUTPUT @E5100;"OUTPSTIM?"    ! TRANSFER STIMULUS VALUE IN BINARY
370  ENTER @E5100 USING "#,8A";Header$
380  ENTER @Dt;Stim(*)            !
390  ENTER @E5100 USING "#,1A";End$
400  !
410  ASSIGN @Dt TO *              ! DISPLAY TRANSFERED DATA
420  FOR I=1 TO Nop               !
430  PRINT "POINT";I,Stim(I);"[Hz]",Dat(I);"[dB]"
440  NEXT I                       !
450  !
460  END                          ! END  OF PROGRAM
```

4

Using the I/O Port

Introduction

The E5100A/B has a 24 bit I/O port on the rear panel. This section explains how to use this I/O port from BASIC.

The 24 bit I/O port has the following pin layout. See Appendix D 'Parallel I/O (Standard and Option 006)' of the *User's Guide*.

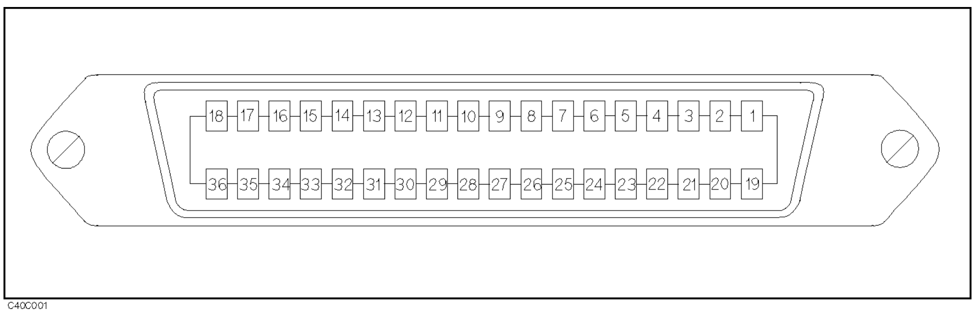


Figure 4-1. The I/O Port Pin Layout

Following is the description of how to use lines that can be controlled by Instrument BASIC.

Output Port (Pin 5 to 28)

The E5100A/B can select 8 types of output ports depending on the output data width. Note that port C (Pin 21 to 24) and port D (Pin 25 to 28) can also function as input ports.

There are 2 ways to output data to an I/O port.

- Use the **WRITEIO** statement of Instrument BASIC.
- Use **OUTAIO** to **OUTHIO** of the GPIB command.

Data Output

Use the following procedure to output data:

1. If you want to use output ports other than A, B or F, you must specify ports C and D as output ports.

Ports Used	GPIB Commands Executed
Port C (4 Bit)	COUT
Port D (4 Bit)	DOUT
Port E (8 Bit)	COUT,DOUT
Port G (20 Bit)	COUT
Port H (24 Bit)	COUT,DOUT

2. Output data from a controller through the I/O ports. Use the following commands to output data.

Ports Used	GPIB Commands	Instrument BASIC
Port A (8 Bit)	`OUTPAIO` *data*	`WRITEIO 15,0;`*data*
Port B (8 Bit)	`OUTPBIO` *data*	`WRITEIO 15,1;`*data*
Port C (4 Bit)	`OUTPCIO` *data*	`WRITEIO 15,2;`*data*
Port D (4 Bit)	`OUTPDIO` *data*	`WRITEIO 15,3;`*data*
Port E (8 Bit)	`OUTPEIO` *data*	`WRITEIO 15,4;`*data*
Port F (16 Bit)	`OUTPFIO` *data*	`WRITEIO 15,5;`*data*
Port G (20 Bit)	`OUTPGIO` *data*	`WRITEIO 15,6;`*data*
Port H (24 Bit)	`OUTPHIO` *data*	`WRITEIO 15,7;`*data*

If you output data to port C or D (including ports E, G and H) when they are specified as input ports, a warning message **CAUTION:WRONG I/O PORT DIRECTION** will be displayed.

When data is outputted, a negative pulse will be outputted to the write strobe output (pin 31) of the output port. Monitor this output to synchronize the transfer when you are reading the data outputted to the I/O port from an external device.

Input Port (Pin 21 to 28)

As described in the previous section, input ports can also function as output ports. If you want to input data, you must set ports C and D to the input mode before processing the data. If ports C and D are set to the input mode, their I/O status line (pins 29 and 30) will be set to LOW. Be sure to check the I/O status lines (pins 29 and 30) of ports C and D to make sure they are set to the input mode when you want to import data from an outside source.

There are 2 ways to read data from an I/O port.

- Use the **READIO** statement of Instrument BASIC.
- Use **OUTPINPCIO?** to **OUTPINPEIO?** of the GPIB command.

Data Input

Use the following procedure to input data from an external device:

1. Set ports C and D as input ports.

Ports Used	GPIB Commands
Port C (4 Bit)	CIN
Port D (4 Bit)	DIN
Port E (8 Bit)	CIN,DIN

2. Monitor the I/O status lines (pins 29 and 30) of ports C and D to make sure they are set to LOW. (If positive logic is specified by the **POSL** command, the input mode will be HIGH.)
3. Input data to the input ports.
4. Execute the following command from the controller to read the data inputted to the I/O ports.

Ports Used	GPIB Commands	Instrument BASIC
Port C (4 Bit)	`OUTPINPCIO?` *data*	`Data=READIO(15,2)`
Port D (4 Bit)	`OUTPINPDIO?` *data*	`Data=READIO(15,3)`
Port E (8 Bit)	`OUTPINPEIO?` *data*	`Data=READIO(15,4)`

The INPUT1 Input (Pin 2) is provided to inform the E5100A/B of a data input. See the "INPUT1 Input (Pin 2), OUTPUT1 Output (Pin 3), OUTPUT2 Output (Pin 4)" commands for additional information.

INPUT1 Input (Pin 2), OUTPUT1 Output (Pin 3), OUTPUT2 Output (Pin 4)

To synchronize the E5100A/B and an external device at a measurement, the status lines are assigned to pins 2 to 4. Their functions are described in the following paragraphs.

INPUT1 Input (Pin 2) — Used to send timing information from an outside source. If there is input to this pin, OUTPUT1 and OUTPUT2 are set to the specified status.

OUTPUT1 Output (Pin 3), OUTPUT2 Output (Pin 4) — Used to send the status of the E5100A/B to an external device. This is set to the status specified at the INPUT1 input. You can also set this to any status using an GPIB command.

The status of OUTPUT1 and OUTPUT2 at the INPUT1 input is set by the following commands.

Commands to Set the Status of OUTPUT1 and OUTPUT2 at the INPUT1 Input

	LOW	HIGH
OUTPUT1	`OUT1ENVL`	`OUT1ENVH`
OUTPUT2	`OUT2ENVL`	`OUT2ENVH`

You can also use the following commands to set OUTPUT1 and OUTPUT2 to any status.

	LOW	HIGH
OUTPUT1	OUT1L	OUT1H
OUTPUT2	OUT2L	OUT2H

Examples of Data I/O

Let's look at an example of data I/O that uses the status line. OUTPUT1 and OUTPUT2 are used as the status indicators.

OUTPUT1 Set to HIGH when the E5100A/B completes a measurement and data process and is waiting for a trigger.

OUTPUT2 Set to HIGH while the E5100A/B is performing a measurement.

The following is a timing chart to describe the status.

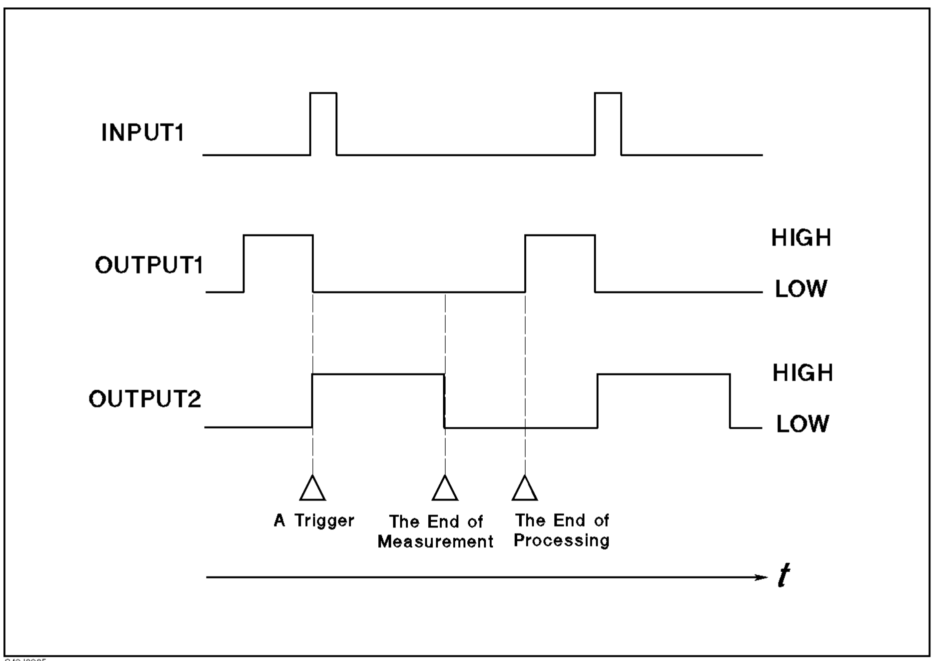


Figure 4-2. INPUT1, OUTPUT1, OUTPUT2 Timing Chart

1. Specify the status of OUTPUT1 and OUTPUT2 at the INPUT1 input. See Figure 4-2 to set OUTPUT1 to LOW and set OUTPUT2 to HIGH.

```
OUTPUT @E5100;"OUT1ENVL"
OUTPUT @E5100;"OUT2ENVH"
```

2. Start a measurement. Set a trigger at the INPUT1 input to perform a measurement. HIGH is set to OUTPUT1 to go back to the beginning of the loop to wait for a trigger when the data process is completed. You can check the timing of data input to the INPUT1 port by using the **INPT?** query.

```
ON ERROR GOTO Finished
LOOP
OUTPUT @E5100;"OUT1H"     Set OUTPUT1 to HIGH.
REPEAT
OUTPUT @E5100;"INPT?"     Wait for an INPUT1 input.
ENTER @E5100;Inpt
UNTIL Inpt=1              OUTPUT1 is set to LOW and OUTPUT2 is set to HIGH by the INPUT1 input.
:
:                         Set a trigger and perform a measurement.
:
OUTPUT @E5100;"OUT2L"     Set OUTPUT2 to LOW at the completion of measurement.
:
:                         Process the measured data.
:
END LOOP
Finished:!
```

Module 5-1. Using Data I/O

The Parallel I/O Mode A (Option 005 Only)

The E5100A/B Option 005 has an Input-4 Bit/Output-8 Bit I/O port. The following is a description of how to use the Option 005 I/O port from a BASIC program.

Data Output (4 Bit)

You can use the following methods to output data:

- For Instrument BASIC, use the **WRITEIO** statement.

  `WRITEIO 15,2;`*data*

- For HP BASIC use an GPIB command **OUT8IO**.

  `OUTPUT @E5100;"OUT8IO ";`*data*

Data Input (8 Bit)

You can use the following methods to read inputted data:

- For Instrument BASIC, use the **READIO** statement.

  `Data=READIO(15,2)`

- For HP BASIC use the GPIB command **INP8IO?** or **OUTPINP8IO?**.

```
OUTPUT @E5100;"INP8IO?"     Output data from the I/O port directly.
ENTER @E5100;Data           The return value is 4 bit data.
```

Module 5-2. Reading Input Data

Or

```
OUTPUT @E5100;"INP8IO"        Read data from the I/O port to the memory and output it from the
                              memory.
OUTPUT @E5100;"OUTPINP8IO?"
ENTER @E5100;Data             The return value is 8 bit data with 0s in its upper 4 bits.
```

Module 5-3. Reading Input Data

5

Using Status Reporting Function

The analyzer has status registers that report system conditions. The register contents are changed depending on the particular condition of the analyzer. By reading this register, you can determine the specific analyzer status.

This chapter provides the following information:

- General status register model.
- Status register structure of the E5100A/B.
- How to use status register in a program.
- Sample program: performing calibration.

General Status Register Model

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

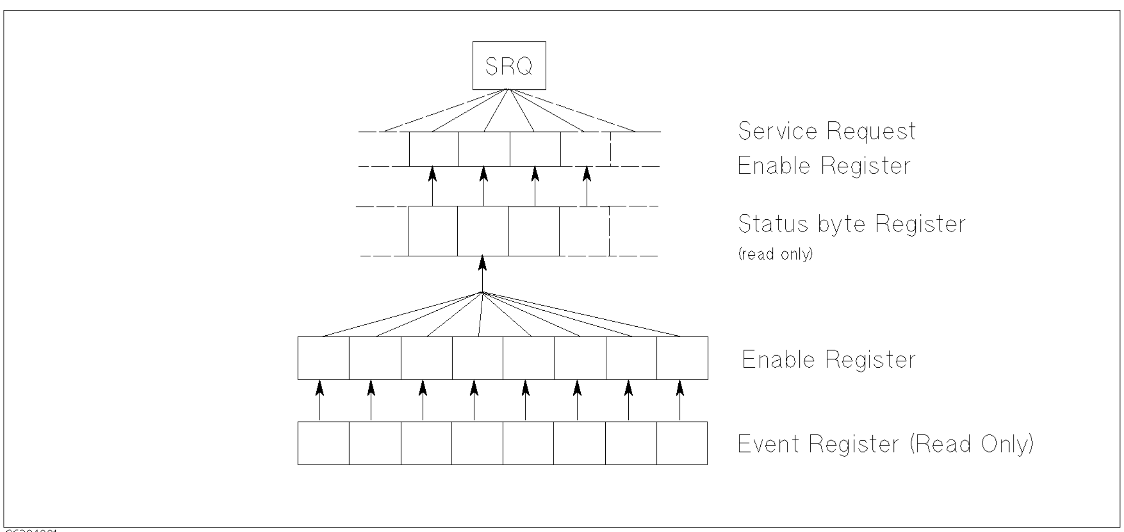


Figure 5-1. General Status Register Model

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

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

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

Event Register

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

You cannot change bit status by GPIB command.

The analyzer has the following event registers:

- Instrument Event Status Register.
- Standard Event Status Register.
- Operation Status Event Register.

Enable Register

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

Use this register to select which event register bits generate the SRQ.

All event registers have a corresponding enable register for each bit.

Status Byte Register

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

The value of the status byte register can be read by using the **SPOLL** statement or the ***STB?** query from the controller. **SPOLL** reads the status byte register value directly without the analyzer being set to remote. Therefore, you can continue to operate front panel keys while a controller is reading the status byte register. On the other hand, the ***STB?** query sets the analyzer to remote mode. Reading the status byte register by either command does not affect the contents of the status byte register, except that **SPOLL** clears the RQS bit.

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

SRQ (Service Request) can be generated by the status byte register by setting the service request enable register. For more information about SRQ, see “SRQ and Interrupt” in this chapter.

Transition Filter and Condition Register

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

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

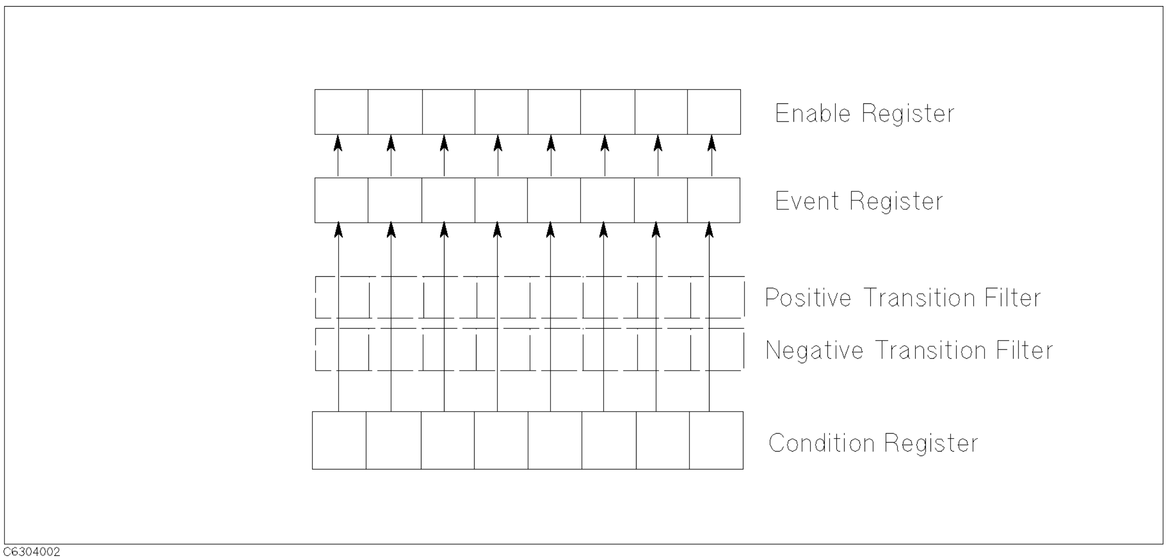


Figure 5-2. Transition Filter and Condition Register

For the E5100A/B, only the "Program Running" bit of the operation status register has a transition filter. By using the transition filter, you can generate an SRQ either at the start or the end of the program execution.

Status Register Structure

Status byte totals three status registers which indicate the internal condition of an instrument. Figure 5-3 shows the status reporting structure of the E5100A/B.

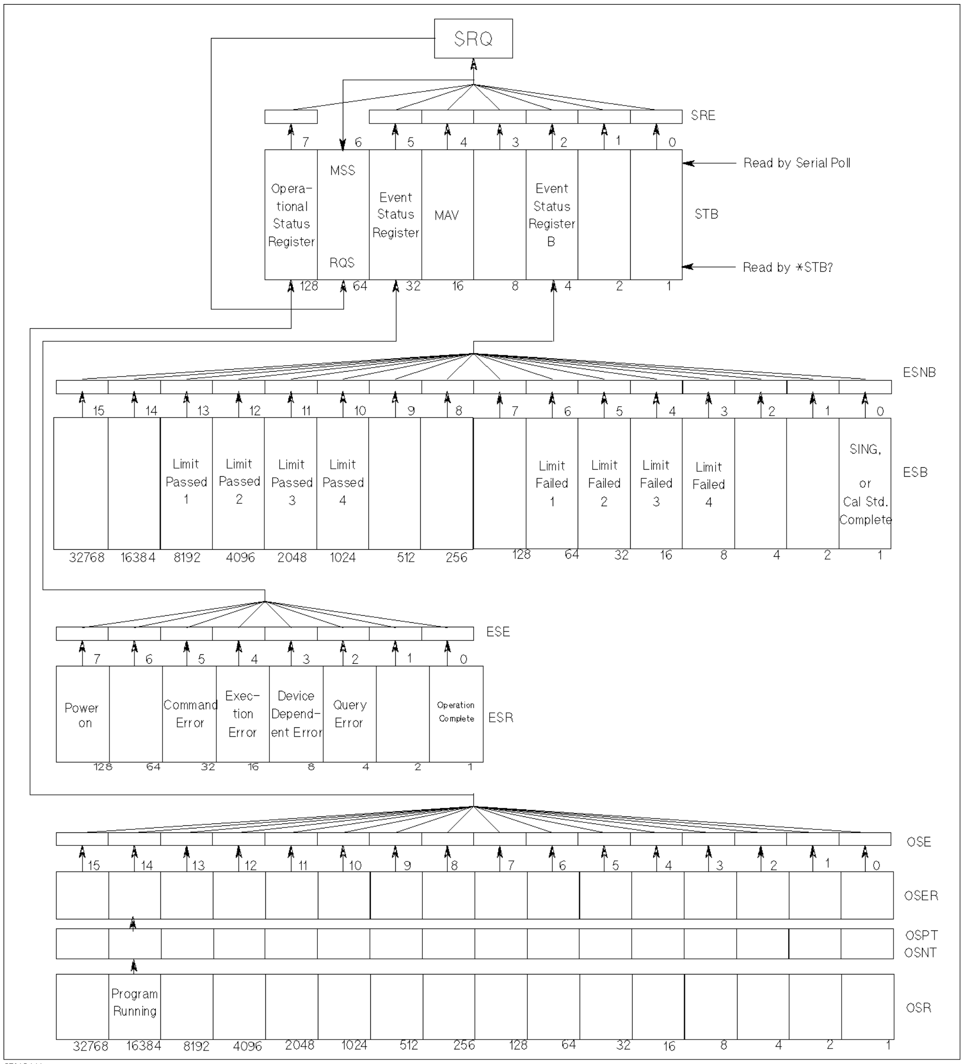


Figure 5-3. Status Reporting Structure

The E5100A/B has a status reporting system to report the condition of the instrument. Status bytes consists of 8-bit registers, each bit represents specific instrument conditions. The value of the status byte can be read by using `SPOLL(717)` statement from an external controller. This command reads value directly from the E5100A/B without being set to remote. So, you can operate front panel keys while a controller is reading the status byte. Contents of the status byte can also be read by using the `*STB?` command. Reading the status byte has no effect on the contents of the status byte. Table 5-1 shows contents of status byte.

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

Bit	Name	Description
2	Check event status register B	One of the enabled bits in event status register B has been set.
4	Message Available	"1" is set when Output Queue has data and "0" is set when Output Queue has no data.
5	Summary bit of event status register	One of the enabled bits in the event status register has been set.
6	MSS (Master Summary Status Bit)	One of the enabled status byte bits is causing an SRQ.
7	Operational status summary bit	One of the enabled bits in the operational status register has been set.

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

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

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

or,

```
10  ASSIGN @E5100; TO 717
20  OUTPUT @E5100;"*STB?"
30  ENTER @E5100;Stat
40  Stb4=BIT(Stat,4)
50  PRINT Stb4
60  END
```

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

The Event Status Register (ESR), Event Status register B (ESB), and Operational Status Register (OSR) are subordinate to the status byte. Each register is set a bit with condition which is watched by status bit. Status bit is cleared when is read by query or *CLS command is executed.

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

Bit	Name	Description
0	Operation complete	A command for which OPC has been enabled, and completed an operation.
2	Query error	1. The E5100A/B has been addressed to talk, but there is nothing in the output queue to transmit. 2. Data in the Output Queue has been lost.
3	Device dependent error	An error, other than a command error, a query error, and an execution error has occurred.
4	Execution error	1. A program data element following a header exceeded its input range, or is inconsistent with the E5100A/B's capabilities. 2. A valid program message could not be properly executed due to some instrument condition.
5	Command error	1. An IEEE 488.2 syntax error has been occurred. Possible violations include, a data element violated the E5100A/B listening formats or a data element type is unacceptable to the E5100A/B. 2. A semantic error which indicates that an unrecognized header was received has occurred. Unrecognized headers include incorrect device-specific headers and incorrect or unimplemented IEEE 488.2 common commands.
7	Power on	A power-on sequence has occurred.

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

Bit	Name	Description
0	Sweep or group complete, or cal std. complete	A single sweep or group has been completed since the last read of the register. Operates in conjunction with SING or NUMG.
3	Limit failed, 4	Limit test failed on trace 4.
4	Limit failed, 3	Limit test failed on trace 3.
5	Limit failed, 2	Limit test failed on trace 2.
6	Limit failed, 1	Limit test failed on trace 1.
10	Limit Passed, 4	Limit test passed on trace 4.
11	Limit Passed, 3	Limit test passed on trace 3.
12	Limit Passed, 2	Limit test passed on trace 2.
13	Limit Passed, 1	Limit test passed on trace 1.

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

Bit	Name	Description
14	Program running	An HP Instrument BASIC program is running.

Each status register has a register mask which enables generating Service ReQuest (SRQ) with condition of a status bit. For instance, to generate an SRQ when the E5100A/B completes the specified number of sweep, enable ESNB bit 1 which is the mask register for ESB 0 ("SING, NUMG, or Cal Std. Complete") which shows sweep completion and SRE bit 2. This makes a path from ESB bit 0 to an SRQ. This example is listed as a program listing:

```
10  ASSIGN @E5100 TO 717
20  !
30  OUTPUT @E5100;"CLES"   ! Clears status registers
40  OUTPUT @E5100;"ESNB 1" ! Enables mask register of "SING. NUMG. or
50  !                        ! Cal Std. Complete" of ESB
60  OUTPUT @E5100;"*SRE 4" ! Enables mask register of "Event Status
70  !                        ! Register B" of STB
80  !
90  ON INTR 7 GOTO End       ! Declare SRQ interrupt
100 ENABLE INTR 7;2
100 OUTPUT @E5100;"SING"   ! Execute single sweep
110 GOTO 110                 ! Endless loop
120 !
130 End:                     ! Exit from loop when sweep is completed
140 END
```

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

OSPT, OSNT

OSPT (Operational Status Positive Transition Filter)

Sets the positive transition filter. Setting a bit in OSPT will cause a 0 to 1 transition in the corresponding bit of the associated operational status register (OSR) to cause a 1 to be written in the associated bit of corresponding operational status event register (OSER).

Because only bit 17 of the E5100A/B's OSR is used to show program status, when bit 17 of OSPT is set to 1, starting a program causes a 1 to be written in bit 17 of OSER. (And then a 1 is written in bit 7 of STB.)

OSNT (Operational Status Negative Transition Filter)

Sets the negative transition filter. Setting a bit in the negative transition filter will cause a 1 to 0 transition in the corresponding bit of the associated operational status register to cause a 1 to be written in the associated bit of corresponding operational status event register.

Because only bit 17 of the E5100A/B's OSR the is used to show program status, when bit 17 of OSNT is set to 1, stopping a program causes a 1 to be written in bit 17 of OSER. (And then a 1 is written in bit 7 of STB.)

How to Use the Status Registers in a Program

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

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

Reading an Event Register Directly

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

1. Query the event register contents.
2. Retrieve a return value.
3. Check the bit condition using the BASIC **BIT** function.

```
OUTPUT @E5100;"ESB?"             Queries instrument event status register contents.
ENTER @E5100;Esb                 Retrieve return value.
IF BIT(Esb,4) THEN               If bit 4 of the instrument event status register is set to 1, the limit
  DISP "LIMIT TEST FAILED AT Ch 1."  test failed on channel 1.
END IF
```

Module 4-1. Reading an Event Register

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

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

SRQ and Interrupt

You can initialize your program to enable interrupt processing by the Service Request (SRQ) from the analyzer. The analyzer generates an SRQ when the specified condition is satisfied.

The SRQ itself does not contain information on the SRQ source. However, the Request Service (RQS) bit in the Status Byte Register of the SRQ source device is set to 1. If multiple devices are connected on the bus, you can check the RQS bit (bit 6) of the analyzer by using a serial poll, **SPOLL**.

Use the SRQ interrupt if you want to determine when the condition changes. The following procedure is used to set the SRQ interrupt:

1. Define the branch for the interruption. (Use **ON INTR** statement.)
2. Set the enable register for the corespondent event register bit to 1.
3. Set the service request enable register bit for the correspondent status byte register bit to 1.
4. Clear the status register before enabling the SRQ interruption.
5. Enable the SRQ interruption. (Use **ENABLE INTR** statement.)
6. Start the event.
7. Wait for the SRQ. Usually, the program waits within an endless loop.
8. If multiple devices that can generate an SRQ exist on the GPIB, you should check bit 6 of the status byte register of the target device. If the SRQ is generated from the target device, the status byte register bit 6 is set to 1.

The following example uses an SRQ interruption for detecting the end of sweep. Bit 0 of the instrument event status register is used for this purpose.

`ON INTR Scode GOTO Sweep_end`	*When the SRQ has occurred, jumps to label, "Sweep_end."*
`OUTPUT @E5100;"ESNB 1"`	*Set bit 0 of the instrument event status enable register to 1.* ($2^0 = 1$)
`OUTPUT @E5100;"*SRE 4"`	*Set bit 2 of the service request enable register to 1.* ($2^2 = 4$)
`OUTPUT @E5100;"*CLS"`	*Clears the event register.*
`OUTPUT @E5100;"*OPC?"`	*Confirms the clear operation is completed.*
`ENTER @E5100;Opc`	
`!`	
`OUTPUT @E5100;"HOLD"`	*Sets the trigger mode HOLD.*
`ENABLE INTR Scode;2`	*Enables the SRQ interruption just before triggering.*
`OUTPUT @E5100;"SING"`	*Trigger a measurement.*
`Waiting: GOTO Waiting`	*Waits until SRQ is generated.*
`Sweep_end:!`	*When on SRQ, program jumps to this label.*
`IF NOT BIT(SPOLL(@E5100),6) THEN`	*Check the SRQ is generated from the target device by checking status byte register bit 6.*
`  ENABLE INTR Scode;2`	*If not, enable the SRQ again, then*
`  GOTO Waiting`	*returns to the endless looping.*
`END IF`	

Module 4-2. Detecting Sweep End Using SRQ and Interrupt

Note

***CLS** clears only the event registers and the status byte register. The enable register and transition filter settings are not altered by executing the ***CLS** command. To clear the enable register and transition filter, use the **PRES** command.

Figure 5-7 shows the SRQ generation sequence of the example above.

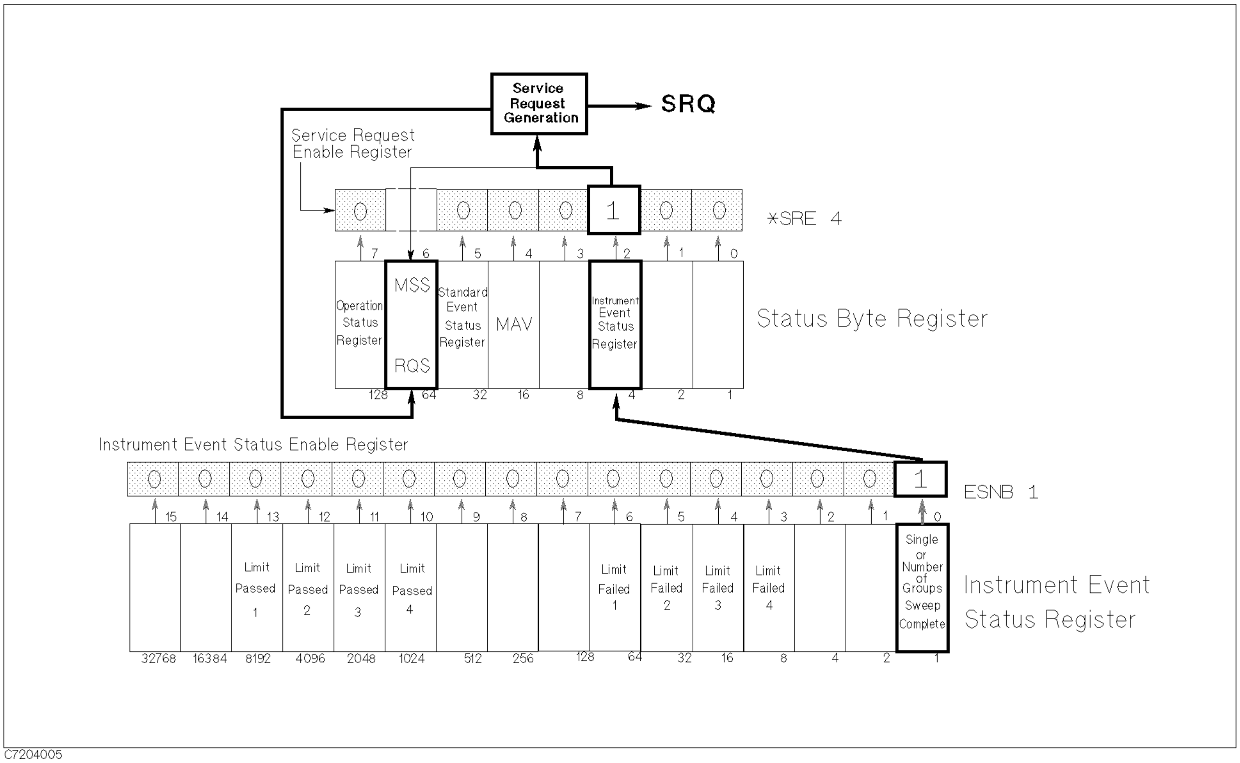


Figure 5-7. SRQ Generation Sequence

- Related GPIB Commands

The following GPIB commands are used for setting the SRQ generation:

*SRE *decimal*	Sets the service request enable register.
*ESE *decimal*	Sets the enable register for event status register.
ESNB *decimal*	Sets the enable register for instrument event status register.
OSE *decimal*	Sets the enable register for operation status register.
OSPT *decimal*	Sets the transition filter to positive for operation status register.
OSNT *decimal*	Sets the transition filter to negative for operation status register.

6

Programming Miscellaneous

Introduction

This chapter describes methods to use the E5100A/B, Instrument BASIC and an external controller together and some notes on the operation. The later sections introduce techniques for programming and hints to increase process speed.

This chapter describes the following topics:

- Using an External Controller and the E5100A/B Together
- Controlling Instrument BASIC from an External Controller
- Programming Techniques

Using an External Controller and the E5100A/B Together

The E5100A/B has the ability to function as a controller to control other GPIB devices. If you want to use an external controller, you can also use both of them on the same bus. This section explains how to use the two controllers on the same bus.

Locking Out Local Operation (LOCAL LOCKOUT)

If you are controlling the E5100A/B with an external controller, the E5100A/B is in the remote mode. In the remote mode, the front panel key operations are not accepted. You can, however, press the LOCAL key to cancel the remote mode. To prevent the operator from pressing the LOCAL key by mistake and changing the device setup, you can make all keys (including the LOCAL key itself) unavailable. This status is called Local Lockout.

You can set local lockout as follows.

```
ASSIGN Scode TO 7
LOCAL LOCKOUT Scode
```

Module 6-1. Setting Local Lockout

You can cancel local lockout by sending the **LOCAL** command.

```
LOCAL Scode     Cancel the all devices on the bus.
LOCAL @E5100    Cancel any one device.
```

Module 6-2. Canceling Local Lockout

How To Pass Control (PASS CONTROL)

Though more than one controller can be on the same bus, only one controller can actually control the bus at any one time. The controller that has the right to control the bus is called the Active Controller.

The controller that has control at system startup is called the System Controller. The system controller is specified physically and it cannot be controlled and changed by a program. If you want to use more than one controller, you must pass control to the other controller. The **PASS CONTROL** statement of BASIC is used to transfer control to another controller.

```
PASS CONTROL @E5100
```

Module 6-3. Pass Control

Only the active controller can execute this statement.

The system controller can become the active controller at any time by executing the ABORT statement.

```
ABORT 7
```

Module 6-4. Abort

Only the system controller can execute this statement.

Controlling Instrument BASIC From an External Controller

The E5100A/B has a command called the Program Subsystem command that enables an external controller to control Instrument BASIC. The program subsystem command enables you to execute a program written in Instrument BASIC or refer to the variables used by Instrument BASIC from an external controller. This section will describe the following topics related to the program subsystem command usage.

- Referring to and Transferring an Array
- Remote Processing a BASIC Command
- Transferring a Program

Note The following commands and programs are all executed on an external controller.

Note The program subsystem command is based on SCPI (Standard Commands for Programmable Instrument).

Referring to and Transferring an Array

You may need to exchange data between programs when you are using an external controller and Instrument BASIC together. This section describes how to use the program subsystem command to exchange data.

Referring to a Numeric Variable

Execute the following statements to refer to numeric variable data in Instrument BASIC from an external controller:

```
OUTPUT @E5100;"PROG:NUMB? 'Dat'"
ENTER @E5100;Dat
```

Module 6-5. Referring to a Numeric Variable

Referring to a String Character Variable

Similarly, refer to a string character variable as follows:

```
OUTPUT @E5100;"PROG:STR? 'String$'"
ENTER @E5100;String$
```

Module 6-6. Referring to a String Character Variable

Transferring a Numeric Variable

Execute a command with a variable name and data as a parameter to transfer numeric data to a variable in Instrument BASIC.

```
OUTPUT @E5100;"PROG:NUMB 'Center',100000000"
INPUT "ENTER CENTER FREQUENCY",Center
OUTPUT @E5100;"PROG:NUMB 'Center',";Center
```

Module 6-7. Transferring a Numeric Variable

Transferring a String Character Variable

Execute a command with a variable name and data as a parameter to transfer string character data to a variable in Instrument BASIC.

```
OUTPUT @E5100;"PROG:STR 'File$','TEST1'"

File$="TEST1"
OUTPUT @E5100;"PROG:STR 'File$','";File$;"'"
```

Module 6-8. Transferring a String Character Variable

Referring to and Transferring an Array

Similarly, you can refer to the contents of an array.

`DIM Dat_array(1:201,1:2)`	*Define an array of the same size as the array you are referring to.*
`OUTPUT @E5100;"PROG:NUMB? 'Dat_array'"`	
`ENTER @E5100;Dat_array(*)`	

Module 6-9. Transferring Array

You cannot refer to an array by each element (Ex. **PROG:NUMB? 'Dat_array(1,1)'**).

Execute a command with an array name and data as a parameter to transfer array data.

```
OUTPUT @E5100;"PROG:NUMB ""Dat_array"",";Dat_array(*)
```

Module 6-10. Referring to Array

In fact, the program is more complex because you have to synchronize two programs. See "Sample Program -3: Controlling Instrument BASIC from an External Controller".

Remote Processing a BASIC Command (PROG:EXEC)

You can remote process a BASIC command supported by the E5100A/B's BASIC command lines.

Following is an example of how to use the **PROG:EXEC** command.

```
OUTPUT @E5100;"PROG:EXEC 'EDIT'"                    Start up the BASIC editor.
OUTPUT @E5100;"PROG:EXEC 'MSI "":INTERNAL,4""'"     Select the internal disk drive.
INPUT "Enter File Name:",File$
OUTPUT @E5100;"PROG:EXEC 'GET """;File$;"""'"       Read a file into the editor.
```

Module 6-11. PROG:EXEC Command

Controlling the Process Status

You can check or set the process status of a BASIC program from an external controller.

Controlling the Process Status of Instrument BASIC

You can use the **PROG:STAT** command to control the process status of Instrument BASIC from an external controller.

```
OUTPUT @E5100;"PROG:STAT RUN"   Execute a program.
```

Module 6-12. PROG:STAT Command

See the *GPIB Command Reference* for detail.

Checking the Process Status of an Instrument BASIC Program

The following are examples of how to remote check the process status of Instrument BASIC.

- Use the **PROG:STAT?**Query.
  Depending on the return value, you can see if the process is in the **RUN**, **PAUSE** or **STOP** status.

```
OUTPUT @E5100;"PROG:STAT?"   Use Query to get the process status.
ENTER @E5100;Status$         The return value is character string.
DISP Status$
```

Module 6-13. PROG:STAT? Query

- Check the bit 14 of OSR (Operation Status Register).

  B 14 of the OSR is set depending on the process status of Instrument BASIC. A **1** is set if the process is running, or a **0** is set if it is stopped (including the PAUSE status).

```
OUTPUT @E5100;"OSR?"
ENTER @E5100;Osr
BIT(Osr,14)
```

Module 6-14. Checking Operation Status Register

When you want to generate an SRQ according to the process status of Instrument BASIC, you can select whether you want to generate it at the start of program or at the completion of program by specifying OSPT or OSNT.

```
OUTPUT @E5100;"OSPT 16384"          Generate an SRQ at the start of program.
OUTPUT @E5100;"OSE 16384; *SRE 32"
```

Module 6-15. Generating SRQ

Change **OSPT** to **OSNT** to generate an SRQ at the completion of program.

Note **CLES** or ***CLS** cannot clear **OSPT** or **OSNT**. Execute **OSPT 0** or **OSNT 0** to clear the setup.

- Use the **PROG:WAIT?** command to pause an external controller until a program completes. **1** is returned at the completion of running Instrument BASIC if you execute the **PROG:WAIT?**Query. If you read the return value by the **ENTER** statement, the program will pause until the value is returned.

```
OUTPUT @E5100;"PROG:WAIT?"   Wait for a program to complete.
ENTER @E5100;Wt              1 is returned if the program is completed.
```

Module 6-16. PROG:WAIT Command

- Use Instrument BASIC to set the variable to indicate the program status and refer to this variable by the **PROG:NUMB?** or **PROG:STR?**Query from an external controller. This is useful when you want to know the actual process status of an Instrument BASIC program. Ex. Check if a calibration is completed. You can have controllers to cooperate and communicate with each other in detail if you use the **CASE** statement in an external controller to branch to appropriate process depending on the contents of this variable. A sample program that uses this method is in “Sample Program -3: Controlling Instrument BASIC from an External Controller”.

Transferring a Program

You can transfer a program between an external controller and Instrument BASIC.

Transferring from an External Controller to Instrument BASIC

The following example transfers a program on the disk in an external controller to the Instrument BASIC editor through GPIB.

`OUTPUT @E5100;"PROG:DEL:ALL"`	*Clear the editor before a transfer.*
`ASSING @File TO Filename$`	
`ON ERROR GOTO Done`	
`DIM LINE$[100]`	
`OUTPUT @E5100;"PROG:DEF #0"`	*Send the header.*
`LOOP`	
`  Line$=""`	
`  ENTER @File;Line$`	*Read and transfer a program line by line.*
`  OUTPUT @E5100;Line$`	
`END LOOP`	
`DONE:!`	*If the control gets to the end of file, it jumps to this label by an error interrupt.*
`OFF ERROR`	
`OUTPUT @E5100;""END`	*Send a terminator and end the transfer.*

Module 6-17. Transferring a Program to the Instrument BASIC Editor

Instrument BASIC checks the transferred program and comments out lines that have grammatical errors.

Transferring from Instrument BASIC to an External Controller

You can also transfer a program from the Instrument BASIC editor to an external controller.

The following example stores each Instrument BASIC line to a character string array **PROG$**.

`DIM PROG$(1:3000)[100]`	*Up to 3000 line program can be stored.*
`!`	
`ON ERROR GOTO Finished`	
`ENABLE INTR Scode;2`	
`OUTPUT @E5100;"PROG:DEF?"`	
`ENTER @E5100 USING "#,8A";Heade$`	
`ENTER @E5100;PROG$(*)`	
`!`	
`Finished: OFF ERROR`	*In case a program does not have exactly 3000 lines, use an error interrupt to prevent the program from abnormally exiting.*

Module 6-18. Transferring a Program From the Instrument BASIC Editor

Sample Program -3: Controlling Instrument BASIC from an External Controller

This program runs an external controller and Instrument BASIC at the same time to synchronize the function of both sides. To check the process status of Instrument BASIC from an external controller, you must set up a variable (**Stat$**) to indicate the program status of Instrument BASIC. In the external controller, use the **SELECT** . . . **CASE** statement to process data according to the contents of **Stat$**.

Actually, Instrument BASIC handles processes such as setups, calibrations, measurements and data analysis and then transfers the results to the external controller. The external controller requests the process at each step.

Note

The **EXECUTE** process may not complete successfully if an external controller sends an GPIB command while Instrument BASIC is processing another command using the **EXECUTE** statement. The E5100A/B may even stop in certain circumstances. To prevent this, do not send a new command from an external controller while the E5100A/B is processing a command. See for an actual method.

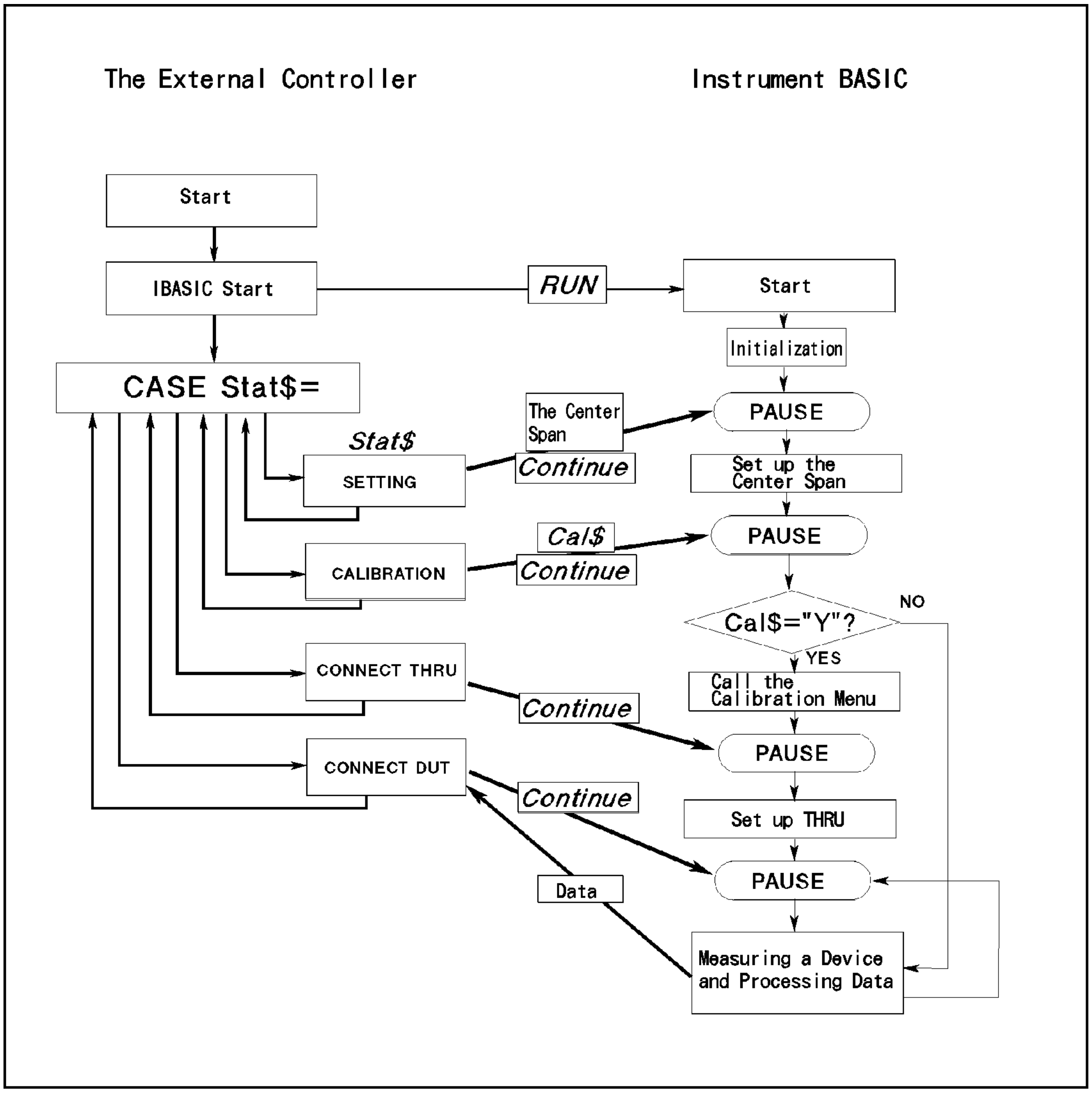


Figure 6-1.
Controlling Instrument BASIC from an External Controller (The Control Flow)

The Program for Instrument BASIC

Disk This program is contained in the attached sample disk with the file name **IB_CTRL**.

```
110   ASSIGN @E5100 TO 800          ! IBASIC INITIALIZATION
120   Scode=8                       !
130   CLEAR @E5100                  !
140   !
150   Stat$="SETTING"               !
160   Cal$=""                       !
170   Center=0                      !
180   Span=0                        !
```

```
190    !
200    OUTPUT @E5100;"DISAHIHB"     ! MEASUREMENT SETUP
210    OUTPUT @E5100;"PRES"         !
220    OUTPUT @E5100;"CHAN1;ANAMODE GAINP;MEAS AR;FMT LOGM"
230    OUTPUT @E5100;"HOLD"         !
240    !
250    EXECUTE "ANAOCH1"            ! SETUP WAVEFORM ANALYSIS
260    EXECUTE "ANARFULL"           !
270    EXECUTE "ANAODATA"           !
280    !
290    PAUSE                        !
300    !
310    OUTPUT @E5100;"CENS ";Center,Span
320    !
330    Stat$="CALIBRATION"          !
340    PAUSE                        !
350    !
360    IF Cal$="Y" OR Cal$="y" THEN GOSUB R_cal
370    !
380    LOOP                         !
390      Stat$="CONNECT DEVICE"     !
400      DISP "CONNECT DEVICE"      !
410      PAUSE                      !
420    !
430      Stat$="MEASUREMENT"        !
440      DISP "NOW MEASURING..."    !
450      EXECUTE "SING"             !
460    !
470      WRITEIO 8,0;-3             !
480      EXECUTE "OUTPFILT?"        !
490      DIM Ret(5)                 !
500      FOR I=0 TO 5               !
510        Ret(I)=READIO(8,I)       !
520      NEXT I                     !
530    !
540      PAUSE                      !
550    END LOOP                     !
560    !
570    STOP                         !
580    !
590 R_cal:                          !
600    Stat$="CONNECT R"            !
610    DISP "Connect THRU"          !
620    PAUSE                        !
630    DISP                         !
640    OUTPUT @E5100;"CALI RESP"    !
670    OUTPUT @E5100;"STANC?"       !
680    ENTER @E5100;Tmp             !
690    !
730    OUTPUT @E5100;"RESPDONE?"     !
740    ENTER @E5100;Tmp             !
760    RETURN                       !
770    !
780    END                          ! END OF PROGRAM
```

The Program for the External Controller

Disk

This program is contained in the attached sample disk with the file name **IBCTRL_E**.

```
110    ASSIGN @E5100 TO 717         ! EXT CONTROLLER INITIALIZATION
120    Scode=7                      !
130    ABORT Scode                  !
140    !
150    INPUT "Center Frequency (MHz) ?",Center ! SWEEP CONDITION ENTRY
160    Center=Center*1.E+6          !
170    INPUT "SPAN (kHz) ?",Span    !
180    Span=Span*1000.              !
190    !
200    OUTPUT @E5100;"CLES"         !
210    OUTPUT @E5100;"*OPC?"        !
220    ENTER @E5100;Opc             !
230    !
240    OUTPUT @E5100;"PROG:STAT RUN"
250    !
260    OUTPUT @E5100;"OSNT 16384;OSPT 0" !
270    OUTPUT @E5100;"OSE 16384;*SRE 128"
280    ON INTR Scode GOTO Paused    !
290    !
300 Begin:                          !
310    WAIT .5                      !
320    ENABLE INTR Scode;2          !
330    LOOP                         !
340    END LOOP                     !
350 Paused:                         !
360    !
370    OUTPUT @E5100;"PROG:STR? 'Stat$'"
380    ENTER @E5100;Stat$           !
390    !
400    SELECT Stat$                 !
410    CASE """SETTING"""           !
420      OUTPUT @E5100;"PROG:NUMB'CENTER',";Center
430      OUTPUT @E5100;"PROG:NUMB'SPAN',";Span
440      GOSUB Continue             !
450    !
460    CASE """CALIBRATION"""       !
470      INPUT "CAL?",Cal$          !
480      OUTPUT @E5100;"PROG:STR 'Cal$','";Cal$;"'"
490      GOSUB Continue             !
500      !
510    CASE """CONNECT R"""         !
520      INPUT "Connect THRU, Then Press [ENTER]",A$
530      GOSUB Continue             !
540      !
550    CASE """CONNECT DEVICE"""    !
560      INPUT "Connect Device, Then Press [ENTER]",A$
570      GOSUB Continue             !
```

```
580      !
590    CASE """MEASUREMENT"""       !
600      DIM Ret(5)                 !
610      OUTPUT @E5100;"PROG:NUMB? 'RET'"
620      ENTER @E5100;Ret(*)        !
630      CLEAR SCREEN               !
640      PRINT "LOSS:",Ret(0),"[dB],BW:",Ret(1),"[Hz]"
650      PRINT "CENT Freq:",Ret(2),"[Hz],Q:",Ret(3)
660      PRINT "D.LF:",Ret(4),"[Hz],D.RF:",Ret(5),"[Hz]"
670      GOSUB Continue             !
680    !
690    END SELECT                   !
700    GOTO Begin                   !
710    !
720 Continue:                       !
730    OUTPUT @E5100;"CLES"         !
740    OUTPUT @E5100;"*OPC?"        !
750    ENTER @E5100;Opc             !
760    OUTPUT @E5100;"PROG:STAT CONT"
770    RETURN                       !
780    END                          ! END OF PROGRAM
```

Programming Techniques

This section provides information that you may find useful when you write a program to control the E5100A/B using BASIC.

This section provides the following information.

- Using a Disk
- Using a Softkey Label
- Measuring the Processing Time
- Checking GPIB Errors
- Tips for Increasing Speed

Using a Disk

The E5100A/B has 2 storage devices: a disk drive and a RAM disk drive. There are 2 ways to access these storage devices from a controller.

- Accessing Via the Measurement Module
  Use this method to save or read information, such as the setup of the E5100A/B, by using the GPIB command.

  The following program saves measurement setup data to the internal disk with a specified file name.

```
10   ASSIGN @E5100 TO 717
20   INPUT "Enter File Name (without extension)",File_name$
30   OUTPUT @E5100;"STODDISK"                          Use STODMEMO for the RAM disk.
40   OUTPUT @E5100;"SAVDSTA """;File_name$;""""
50   END
```

Module 6-19. Saving Measurement Setup Data

The following program loads the saved setup data to the E5100A/B.

```
10   ASSIGN @E5100 TO 717
20   INPUT "Enter File Name (with extension)",File_name$
30   File_name$=UPC$(File_name$)                    You must specify a file name in
                                                    capital letters.
40   OUTPUT @E5100;"STODDISK"
50   OUTPUT @E5100;"RECD """;File_name$;""""
60   END
```

Module 6-20. Loading Measurement Setup Data

- Accessing Directly from Instrument BASIC
  This method accesses a file directly without going through the E5100A/B. You use this method when you are handling file data itself. See Chapter 7 “Data Storage and Retrieval” of *HP Instrument BASIC Programming Technique* for more information.

  The following program stores trace data on the internal disk into an array **Dat** in the program.

```
10   ASSIGN @E5100 TO 717
20   INPUT "Enter File Name (without extension)",File_name$
30   MSI ":INTERNAL,4"                                     Use MSI ":MEMORY,0" for the
                                                           RAM disk.
40   DIM Dat(1:201,1:2)
50   File_name$=UPC$(File_name$)&"_D"
60   !
70   ASSIGN @File TO File_name$
80   ENTER @FIle USING "17X,#"                             Read the header.
90   ENTER @File;Nop
100  ENTER @File USING "4X,#"                              Store the data into an array.
110  ENTER @File;Dat(*)
120  ASSIGN @File TO *
130  PRINT Dat(*)
140  END
```

Module 6-21. Storing Trace Data

Note

The setups **STODDISK/STODMEMO** and **MSI** are stored independently. This causes the RAM disk to be accessed by an HP disk access command, as long as **STODMEMO** is executed, even though the internal disk drive is selected by **MSI**.

Using a Softkey Label (ON KEY LABEL)

You can use the **ON KEY LABEL** statement to display your own softkey label.

```
ON KEY 1 LABEL "ORIGINAL    LABEL" GOSUB Jump1
!
LOOP
END LOOP
```

Module 6-22. Using a Softkey Label

The label is displayed only when the program is running. You must, therefore, use an infinite loop to keep the programming running.

Up to a 20-character softkey label can be displayed.

Displaying the Softkey Label

The softkey label displayed by **ON KEY LABEL** is displayed when SYSTEM I-BASIC ON KEY LABELS are pressed.

You can use the GPIB command **BASL** to display the softkey label.

```
OUTPUT @E5100;"BASL"     Display the softkey label.
```

Module 6-23. Using Softkey Label

See "Sample Program -4: Reading and Running a Program by a Softkey Operation" which uses the softkey label.

Measuring the Processing Time (TIMEDATE)

You can use the real time clock of the E5100A/B to time the processing of a program.

You can time a process as follows:

```
Start_time=TIMEDATE                    The start time.
  :
[The processes to time.]
  :
End_time=TIMEDATE                      The end time.
Total=End_time-Start_time              Calculate the elapsed time.
PRINT Total,"seconds of processing"
```

Module 6-24. Measuring Processing Time

The real time clock measures in seconds [s].

Checking GPIB Errors (OUTPERRO?)

You can check if an GPIB error occurred in a program by using the **OUTPERRO?**Query. You can use this function to easily find bugs when you are debugging a program. You can check for an GPIB error as follows:

```
DIM Err$[50]
!
OUTPUT @E5100;"OUTPERRO?"
ENTER @E5100;Err$
DISP Err$
```

Module 6-25. Checking GPIB Errors

For example, the following message will be displayed if you use the above procedure to check an error occurred when you send a wrong GPIB command.

```
-113,"Undefined header"
```

You can look at “Error Messages” in the *GPIB Command Reference* to see if the command sent is supported by the E5100A/B.

An error message is stored in the error queue. Execute ***CLS** or **CLES** to clear the error queue.

Tips for Increasing Speed

This section describes how to increase the processing speed of a program. Please modify your program if you find that you can use some of these suggestions.

- Try not to use markers for a data analysis. If you have to use them, use the **MARD OFF** command to erase the marker display.
- Use Fast Instrument BASIC function.
- For Instrument BASIC, do not use the **ON INTR** interrupt.
  Do not use an interrupt process but refer to the status register directly.

```
REPEAT
  OUTPUT @E5100;"*ESB?"
  ENTER @E5100;Esb
UNTIL BIT(Esb,0)           Monitor the completion of sweep.
```

Module 6-26. Refering to Status Register

- Try not to switch active channels. Turn dual channels on if you need switch channels.

  The following commands are valid for inactive channels.

OUTPIFORM?	Output the format array data on the inactive channel.
OUTPIRFORM?	Output the actual part of format array data on the inactive channel.
OUTPITMEM?	Output the sub trace data on the inactive channel.
OUTPIRTMEM?	Output the actual part of sub trace data on the inactive channel.

  The trigger command is valid for the active channel. You can, however, use the **REST** command (MEASUREMENT RESTART of (MENU)) to remeasure both the active channel and the inactive channel without switching.

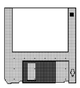
- Use subroutines instead of subprograms.
- Comment out or delete unnecessary lines.

Sample Program -4: Reading and Running a Program by a Softkey Operation

This program displays the file names contained in the inserted disk on the softkey labels and executes the selected file as a program. With this program, you need only softkey operation to easily execute a program.

If you rename this file as **AUTOST**, it will be automatically executed at the startup. This is convenient when no keyboard is connected to the instrument.

Disk



This program is contained in the attached sample disk with the file name **LABEL** (For Instrument BASIC). The program for an external controller is not provided.

```
110  ASSIGN @E5100 TO 800             ! INITIALIZATION
120  DIM Dir$(1:200)[80],File$(1:200)[10]
130  !
140  CAT TO Dir$(*)                   ! STORE FILENAMES ON DISK
150  !
160  File_end=0                       ! READ FILENAMES
170  File_number=1                    !
180    WHILE File_end=0 AND File_number<200
190     File$(File_number)=Dir$(File_number+7)[1,10]
200      IF File$(File_number)="" THEN
210        File_end=1                 !
220        File_number=File_number-1  !
230      ELSE                         !
240        File_number=File_number+1  !
250      END IF                       !
260    END WHILE                      !
270  !
280  Max_page=INT(File_number/6)+1    ! SET ON-KEY-LABEL DISPLAY
290  Npage=1                          !
300  OUTPUT @E5100;"BASL"             !
310  !
320  Head:                            !
330  Page=(Npage-1)*6
340  !
350    ON KEY 1 LABEL File$(Page+1) GOSUB Jump1 ! DISPLAY FILENAMES ON LABELS
360    ON KEY 2 LABEL File$(Page+2) GOSUB Jump2
370    ON KEY 3 LABEL File$(Page+3) GOSUB Jump3
380    ON KEY 4 LABEL File$(Page+4) GOSUB Jump4
390    ON KEY 5 LABEL File$(Page+5) GOSUB Jump5
400    ON KEY 6 LABEL File$(Page+6) GOSUB Jump6
410    ON KEY 7 LABEL "NEXT PAGE" GOTO Jump7
420    ON KEY 8 LABEL "PREV.PAGE" GOTO Jump8
430  !
440    LOOP                           ! WAIT KEY ENTRY
450    END LOOP                       !
460  !
470  Jump1: GET File$(Page+1)          ! GET SELECTED FILE
480  Jump2: GET File$(Page+2)          !
490  Jump3: GET File$(Page+3)          !
```

```
500  Jump4: GET File$(Page+4)          !
510  Jump5: GET File$(Page+5)          !
520  Jump6: GET File$(Page+6)          !
530  Jump7: Npage=Npage+1              !
540    IF Npage>Max_page THEN Npage=Max_page
550    GOTO Head                      !
560  Jump8: Npage=Npage-1              !
570    IF Npage<=1 THEN Npage=1       !
580    GOTO Head                      !
590  !
600  END                              ! END OF PROGRAM
```

Using the Parallel Processing

Because the 87510A performs measurement and analysis sequentially, the total throughput time is summary of the measurement time plus the time required to analyze the measurement data. However, the E5100A/B can perform the measurement and analyze the data simultaneously. This parallel processing improves the total throughput as shown in the following figure:

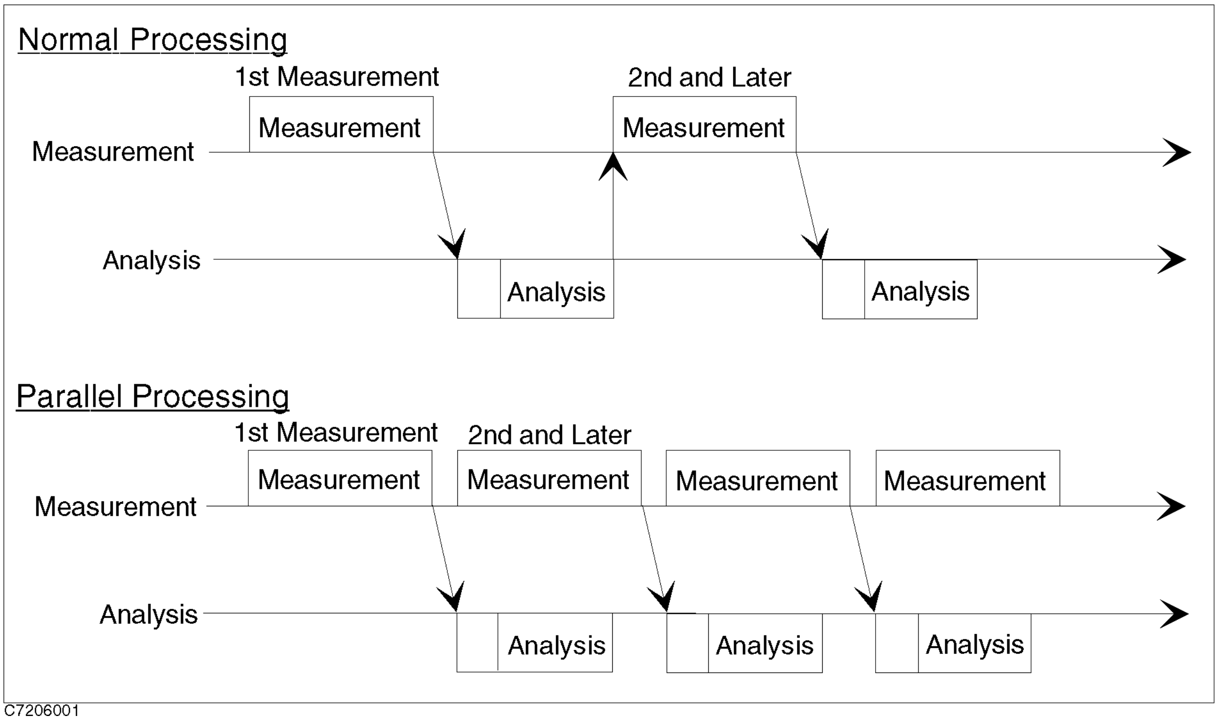


Figure 6-2. Parallel Processing

The parallel processing flow is as follows: The digital signal processor (DSP) measures the DUT, and at the same time the CPU processes the previous measurement's data in parallel with the current measurement. Parallel processing is useful for a production line that successively measures and analyzes many devices and performs GO/NO-GO testing. It cannot be used to make a single measurement and analyze the data at the same time.

Programming for Parallel Processing

The following commands are provided for parallel processing:

`TRIGMEAS` : Triggers the measurement for parallel processing. The DSP starts measuring the DUT.
`MOVADARY` : Enters measurement data triggered by the `TRIGMEAS` command to a data array. The CPU then analyzes this data in parallel with the next measurement. This command must be executed for each measurement.
`ADTOTRAC` : Starts data process into the data trace array. This command must be executed for each measurement.

The following program example shows a parallel processing program that uses these commands:

```
OUTPUT @E5100;"TRIGMEAS"  !    1st. measurement
LOOP
OUTPUT @E5100;"MOVADARY"  !    Data transfer
OUTPUT @E5100;"TRIGMEAS"  !    Start measuring
OUTPUT @E5100;"ADTOTRAC"  !    Start parallel processing
!
OUTPUT @E5100;"PEAK?"     !    Analysis command in parallel processing
ENTER @E5100;F_pk,G_pk    !    Detects two peaks
!
OUTPUT @E5100;"NXPEAK?"   !
ENTER @E5100;F_nxpk,G_nxpk!
!
END LOOP
```

Module 6-27. Parallel Processing

The following commands must be programmed as one group.

```
OUTPUT @E5100;"MOVADARY"  !    Data transfer
OUTPUT @E5100;"TRIGMEAS"  !    Start measuring
OUTPUT @E5100;"ADTOTRAC"  !    Start parallel processing
```

Commands that are executed during parallel processing should be put after ADTOTRAC.

Sample Program -4a :Parallel Processing

The following is a program listing that shows how to use parallel processing. This program executes the analysis commands during the time the DSP measures the DUT. As a result, the total throughput time is improved.

Disk

This program is contained in the attached sample disk with the file name **PARA** (for Instrument BASIC). The program for an external controller has the name **PARA_E**.

```
110    ! CONCURRENT PROCESS
120    !
130    !
140    ASSIGN @E5100 TO 800
150    CLEAR @E5100
160    DIM Pnext(1:10),Fnext(1:10)
170    L1=5
180    !
190    GOSUB Setting
200   GOSUB Thrucal
210   GOSUB Measure
220   STOP
230   !
240 Setting:       !
250   Fc=7.E+7     !70MHZ
260   Fspan=100000 !100kHz
270   Ifbw=8000    !8kHz
280   Nop=201
```

```
290   !
300   OUTPUT @E5100;"PRES"
310   OUTPUT @E5100;"POIN ";Nop
320   OUTPUT @E5100;"CHAN1;ANAMODE GAINP; MEAS AR;FMT LOGM"
330   OUTPUT @E5100;"HOLD"
340   OUTPUT @E5100;"CENT ";Fc;";SPAN ";Fspan
350   OUTPUT @E5100;"IFBW ";Ifbw
360   RETURN
370   !
380 Thrucal:  !
390   OUTPUT @E5100;"CALI RESP"  !Select and execute THRU CAL
391   INPUT "CONNECT THRU, THEN PRESS [RETURN].",Dum$
400   OUTPUT @E5100;"STANC?"
410   ENTER @E5100;Tmp
420   OUTPUT @E5100;"RESPDONE?"
430   ENTER @E5100;Tmp
431   INPUT "CONNECT DUT, THEN PRESS [RETURN].",Dum$
440   RETURN
450   !
460 Measure:  !
470   EXECUTE "ANAOCH1"  ! Select channel for analysis and set analysis range
480   EXECUTE "ANARFULL" !
490   EXECUTE "ANAODATA" !
500   !
510   N=100
520   INPUT "Enter loop N :",N
530   Tstart=TIMEDATE
540   EXECUTE "TRIGMEAS"    !1st. measurement
550   I=1
560   WHILE I<=N            !Repeat measurement and analysis
570   EXECUTE "MOVADARY"    !Enters measurement data for parallel processing
580   EXECUTE "TRIGMEAS"    ! Triggers measurement for parallel processing
590   EXECUTE "ADTOTRAC"    !Starts parallel processing
600   EXECUTE "PEAK?"       ! Commands executed in parallel processing
610   Pmax=READIO(8,0)      !
620   Fmax=READIO(8,1)      !
630   FOR K=1 TO L1         !
640   EXECUTE "NEXPK?"      !
650   Pnext(K)=READIO(8,0)  !
660   Fnext(K)=READIO(8,1)  !
670   NEXT K
680   I=I+1
690   END WHILE
700   Tstop=TIMEDATE
710   !
720   PRINT "PEAK(MAX) :";Pmax;"[dB] @";Fmax/1.E+6;"[MHz]"
730   FOR K=1 TO L1
740   PRINT "PEAK(NEXT[";K;"]):";Pnext(K);"[dB] @";Fnext(K)/1.E+6;"[MHz]"
750   NEXT K
760   PRINT "LOOP :";N;" NEXT[K] :";L1
770   PRINT "TIME(AVE.) : ";(Tstop-Tstart)/N*1000;"[msec]"
780   RETURN
790   !
800   END
```

Notes for Parallel Processing

- Use the `TRIGMEAS` command to trigger the measurement in parallel processing.
- The following command sequence is required:
  1. `TRIGMEAS` (Trigger 1st. measurement)
  2. `LOOP`
  3. `MOVADARY` (Enter data)
  4. `TRIGMEAS` (Trigger measurement)
  5. `ADTOTRAC` (Start parallel processing)
  6. Analysis commands
- The parallel processing command sequence is used from the program.
- There is no advantage in using parallel processing when the analyzer is doing very high speed measurements. This is true because the parallel processing is processed during the measurement. Also, there is no advantage if very complex analysis is done in parallel processing because it takes long time to analyze the data. Parallel processing is not recommended under the following conditions:
  1. When using a short measurement time (less than 0.2msec/point).
  2. When using filter and resonator analysis commands (such as `OUTPXFIL?`, `OUTPRESO?`).
  3. When data is transferred through GPIB.
  4. When a single measurement and analysis is done.

7

Application Sample Programs

This chapter provides you with sample programs that you can use for actual measurements.

- Setting Up and Performing the List Sweep
- Saving and Setting Up Calibration Data
- Analyzing a Ceramic Resonator
- Analyzing a Crystal Resonator

Some of the sample programs listed in this chapter do not include subroutines. In these cases, the required subroutines and their function are listed at the beginning of each sample program.

The programs in the sample program disk contain all the necessary subroutines.

Sample Program -5: Setting Up and Performing the List Sweep

This program sets up the list sweep table and performs a list sweep.

When you run this program, the program will ask you the number of segments, the segment range and the number of displayed points so that you can easily set up the list sweep table.

Disk This program is contained in the attached sample disk with the file name **LIST_SWP** (for Instrument BASIC). The program for an external controller has the name **LISTSWP_E**.

```
110  ASSIGN @E5100 TO 800             ! IBASIC INITIALIZATION
120  CLEAR @E5100                     !
130  OUTPUT @E5100;"DISAHIHB"         !
140  !
150  DIM Table(1:31,1:3)              ! VARIABLE  DECLARATION
160  !
170  INPUT "Number of segments?",Numb ! ENTER AND DISPLAY NUMBER OF SEGMENTS
180  !
190  PRINT USING "10A,11A,11A,20A";"Segment","Start(MHz)","Stop(MHz)"
     ,"Number of points"              !
200  !
210  FOR I=1 TO Numb                  ! GOTO TABLE SETUP ROUTINE
220    GOSUB Loadpoin                 !
230  NEXT I                           !
240  !
250  LOOP                             ! ASK IF  EDIT NEEDED
260    INPUT "Do you want to edit? (Y/N)",An$
270  EXIT IF An$="N" OR An$="n"       !
280    INPUT "Segment Number?",I      !
290    GOSUB Loadpoin                 !
300  END LOOP                         !
310  !
320  OUTPUT @E5100;"PRES"             ! MEASUREMENT SETUP
330  OUTPUT @E5100;"CHAN1;ANAMODE GAINP;MEAS AR;FMT LOGM" !
340  !
350  OUTPUT @E5100;"EDITLIST"         ! TABLE SETUP
360  OUTPUT @E5100;"CLEL"             !
370  FOR I=1 TO Numb                  !
380    OUTPUT @E5100;"SADD"           !
390    OUTPUT @E5100;"STAR ";Table(I,1)*1.E+6                    !
400    OUTPUT @E5100;"STOP ";Table(I,2)*1.E+6                    !
410    OUTPUT @E5100;"POIN ";Table(I,3)                          !
420    OUTPUT @E5100;"SDON"           !
430  NEXT I                           !
440  OUTPUT @E5100;"EDITDONE"         !
450  OUTPUT @E5100;"SWPT LIST"          !
460  OUTPUT @E5100;"LISDOBASE"        !
470  !
480  OUTPUT @E5100;"SING?"            ! MAKE A MEASUREMENT
490  ENTER @E5100;Tmp
```

```
500  OUTPUT @E5100;"AUTO"              ! AUTO SCALING
510  STOP                              ! STOP PROGRAM
520  !
530  Loadpoin:                         ! TABLE PARAMETER SETUP SUBROUTINE
540    INPUT "Enter start frequency (MHz)",Table(I,1)
550    INPUT "Enter stop frequency (MHz)",Table(I,2)
560    INPUT "Enter number of points",Table(I,3)
570    IF Table(I,3)=1 THEN Table(I,2)=Table(I,1)
580    PRINT TABXY(0,I+1);RPT$(" ",58)
590    PRINT TABXY(0,I+1);I;TAB(11);Table(I,1);TAB(22);Table(I,2);
TAB(35);Table(I,3)
600    RETURN                          !
610    !
620    END                             ! END OF PROGRAM
```

Saving and Resetting Calibration Data

This program saves and resets calibration data.

This program transfers error correction data in the calibration coefficient array memory of the E5100A/B to an external controller, saves the data and uses that data in the file as calibration data.

Calibration coefficient arrays corresponding to each error item are stored in three memories as complex numbers. This sample transfers the error data of a frequency response calibration (stored in the array 1) and resets it. See Appendix D of the *GPIB Command Reference* for each error item and its corresponding calibration coefficient array.

You can use either of the following two ways to save calibration data:

1. Use the **SAVDSTA** command to save the data with the measurement setup data to the E5100A/B's internal disk or the RAM disk memory.
2. Transfer calibration data to a controller and save it to the disk in the controller.

This section lists the program for example 2. See "Using a Disk" of Chapter 3 for example 1.

Sample Program -6: Saving Calibration Data in an External Controller

This program transfers calibration data to an external controller and saves it to a file.

Disk

This program is contained in the attached sample disk with the file name **CALSTR_E** (for the external controller). The program for Instrument BASIC is not provided.

```
110  ASSIGN @E5100 TO 717;FORMAT ON            ! INITIALIZATION
120  ASSIGN @Dt TO 717;FORMAT OFF              !
130  Scode=7                                   !
140  CLEAR @E5100                              !
150  CLEAR @Dt                                 !
160  !
170  DIM Dat(1:201,1:2)                        !
180  Nop=201                                   !
190  !
200  GOSUB Setup                               ! CALL SUBROUTINES
210  GOSUB Cal                                 !
220  GOSUB Get_cal_data                        !
230  GOSUB Save_cal_data                       !
240  GOTO Ending                               !
250  !
260 Setup:                                     ! MEASUREMENT SETUP
270  INPUT "CENTER? [MHz]",F_cent              !
280  F_cent=F_cent*1.E+6                       !
290  INPUT "SPAN? [kHz]",F_span                !
300  F_span=F_span*1000                        !
310  OUTPUT @E5100;"PRES"                      !
320  OUTPUT @E5100;"CHAN1;ANAMODE GAINP;MEAS AR;FMT LOGM"
330  OUTPUT @E5100;"HOLD"                      !
340  OUTPUT @E5100;"POIN ";Nop                 !
```

```
350  OUTPUT @E5100;"CENS ";F_cent,F_span         !
360  RETURN                                     !
370  !
380 Cal:                                        ! CALIBRATION
390  INPUT "CONNECT THRU,THEN PRESS [Return]",Dum$
400  OUTPUT @E5100;"CALI RESP"                  !
410  OUTPUT @E5100;"STANC?"                     !
420  ENTER @E5100;Tmp                           !
430                                             !
440 Sweep_end:DISP "SWEEP COMPLETED"            !
450  OUTPUT @E5100;"RESPDONE?"                  !
460  ENTER @E5100;Tmp                           !
470                                             !
480  DISP "THRU CAL. COMPLETED."                !
490  RETURN                                     !
500  !
510 Get_cal_data:                               ! GETTING CALIBRATION DATA
520  INPUT "PRESS [RETURN] TO GET CAL DATA FROM
     HP E5100A/B.",Dum$                         !
530  OUTPUT @E5100;"FORM3"                      !
540  OUTPUT @E5100;"OUTPCALC01?"                !
550  ENTER @E5100 USING "#,8A";Header$          !
560  ENTER @Dt;Dat(*)                           !
570  ENTER @E5100 USING "#,A";End$              !
580  RETURN                                     !
590  !
600 Save_cal_data:                              ! SAVING CALIBRATION DATA
610  INPUT "Enter File Name (with extension)",File$
620  CREATE BDAT File$,Nop,16                   !
630  ASSIGN @File TO File$                      !
640  OUTPUT @File;Dat(*)                        !
650  ASSIGN @File TO *                          !
660  RETURN                                     !
670  !
680 Ending:                                     ! ENDING PROGRAM
690  ASSIGN @Dt TO *                            !
700  !
710  END                                        !
```

Sample Program -7: Setting Up Calibration Data from an External Controller

This program transfers the calibration data saved in an external controller by **CALSTOR_E** to the E5100A/B and sets it up again.

To set the calibration function on again, you must measure the standard to calculate the calibration coefficient. You cannot, however, set the calibration function on if you transferred calibration data only. This means that a measurement was not performed. This sample program, therefore, performs a dummy calibration measurement and transfers calibration data to set the calibration function on. (The subroutine **Dummy_cal**)

If you change the setup (for example, the frequency range) of the E5100A/B after transferring calibration coefficient, the transferred calibration coefficient becomes invalid. You must, therefore, set up the E5100A/B before transferring calibration data.

Disk This program is contained in the attached sample disk with the file name **CALSET_E** (for the external controller). The program for Instrument BASIC is not provided.

```
110  ASSIGN @E5100 TO 717;FORMAT ON              ! INITIALIZATION
120  ASSIGN @Dt TO 717;FORMAT OFF                !
130  Scode=7                                     !
140  CLEAR @E5100                                !
150  CLEAR @Dt                                   !
160  !
170  DIM Dat(1:201,1:2)                          !
180  Nop=201                                     !
190  !
200  GOSUB Setup                                 ! CALL SUBROUTINES
210  GOSUB Load_cal_data                         !
220  GOSUB Dummy_cal                             !
230  GOSUB Send_cal_data                         !
240  GOTO Ending                                 !
250  !
260 Setup:                                       ! ANALYZER SETUP
270  INPUT "CENTER? [MHz]",F_cent                !
280  F_cent=F_cent*1.E+6                         !
290  INPUT "SPAN? [kHz]",F_span                  !
300  F_span=F_span*1000                          !
310  OUTPUT @E5100;"PRES"                        !
320  OUTPUT @E5100;"CHAN1;ANAMODE GAINP;MEAS AR;FMT LOGM"      !
330  OUTPUT @E5100;"HOLD"                        !
340  OUTPUT @E5100;"POIN ";Nop                   !
350  OUTPUT @E5100;"CENS ";F_cent,F_span         !
360  RETURN                                      !
370  !
380 Dummy_cal:                                   ! DUMMY CALIBRATION
390  OUTPUT @E5100;"CALI RESP"                   !
400  OUTPUT @E5100;"STANC?"                      !
420  ENTER @E5100;Tmp                            !
430  !
440 Cal_end:                                     ! CALIBRATION END
450  OUTPUT @E5100;"RESPDONE?"                    !
470  ENTER @E5100;Tmp                            !
480  RETURN                                      !
490  !
500 Load_cal_data:                               ! LOADING CAL DATA
510  INPUT "Enter File Name (with extension)",File$
520  ASSIGN @File TO File$                       !
530  ENTER @File;Dat(*)                          !
540  ASSIGN @File TO *                           !
550  RETURN                                      !
560  !
570 Send_cal_data:                               ! SENDING CAL DATA
580  INPUT "PRESS [RETURN] TO SEND CAL DATA TO HP E5100A/B.",Dum$
590  OUTPUT @E5100;"FORM4"                       !
600  OUTPUT @E5100;"INPUCALC01 ";Dat(*)          !
```

```
610  RETURN                                        !
620  !
630 Ending:                                        ! ENDING PROGRAM
640  ASSIGN @Dt TO *                               !
650  !
660  END                                           !
```

Sample Program -8: Analyzing a Ceramic Resonator

This program analyses the resonant impedance and ripple of a ceramic resonator.

A wave form analysis command **OUTPCERR?** is used for the measurement. The parameters to be measured are Z_r, f_r, Z_a, f_a, Rpl_1, Rpl_2, *and* Rpl_3. See "OUTPCERR?" of the *GPIB Command Reference* for more information.

Measurement Conditions:

- The Log Amplitude & The Phase Format
- Impedance Conversion ON
- Frequency Response Calibration

Items to Prepare:

- A device (a ceramic resonator)
- A cable (or a fixture) to connect the device and the E5100A/B
- THRU calibration standard

Disk

This program is contained in the attached sample disk with the file name **CER_RES** (for Instrument BASIC). The program for an external controller is not provided.

```
110  ASSIGN @E5100 TO 800                       ! IBASIC INITIALIZATION
120  CLEAR @E5100                               !
130  !
140  OUTPUT @E5100;"PRES"                       ! MEASUREMENT SETUP
150  OUTPUT @E5100;"CHAN1;HOLD"                 !
160  OUTPUT @E5100;"ANAMODE ZTRAN;MEAS AR;FMT MAGZP;SCAY 1"
170  OUTPUT @E5100;"COUC OFF"                   !
180  OUTPUT @E5100;"DUAC OFF"                   !
190  OUTPUT @E5100;"POIN 201"                   !
200  OUTPUT @E5100;"IFBW 4KHZ"                  !
210  OUTPUT @E5100;"POWE 0"                     !
240  OUTPUT @E5100;"DISAHIHB"                   !
250  !
260  INPUT "Center Frequency (kHz)",Center      ! SWEEP CONDITION ENTRY
270  Center=Center*1000.                        !
280  INPUT "Frequency Span (kHz)",Span          !
290  Span=Span*1000.                            !
300  OUTPUT @E5100;"CENS ";Center,Span          !
310  !
320  GOSUB Thru_cal                             ! GOTO CAL ROUTINE
330  !
340  EXECUTE "ANAOCH1"                          ! WAVEFORM ANA SETUP
350  EXECUTE "ANAODATA"                         !
360  EXECUTE "ANARFULL"                         !
370  !
380  WRITEIO 8,0;.05                            ! MEASUREMENT
390  EXECUTE "THRR"                             !
400  LOOP                                       !
```

```
410    DISP "Connect Device and Press Continue."
420    PAUSE                                   !
430    DISP "MEASURING"                        !
440    EXECUTE "SING"                          !
450    EXECUTE "OUTPCERR?"                     !
460    Rr=READIO(8,0)                          !
470    Fr=READIO(8,1)                          !
480    Ra=READIO(8,2)                          !
490    Fa=READIO(8,3)                          !
500    Rpl_left=READIO(8,4)                    !
510    Rpl_center=READIO(8,5)                  !
520    Rpl_right=READIO(8,6)                   !
530    GOSUB Printing                          !
540  END LOOP                                  !
550  STOP                                      !
560  !
570 Printing:                                  ! DISPLAY PARAMETERS
580  PRINT USING "5A,6D.3D,6A";"Fr ";Fr/1000.;"(kHz)"
590  PRINT USING "5A,6D.3D,6A";"Rr ";Rr;" (ohm)"
600  PRINT USING "5A,6D.3D,6A";"Fa ";Fa/1000.;"(kHz)"
610  PRINT USING "5A,6D.3D,7A";"Ra ";Ra/1000;" (kohm)"
620  PRINT USING "12A,3D.3D,5A";"Ripple LEFT ";Rpl_left;" (dB)"
630  PRINT USING "14A,3D.3D,5A";"Ripple CENTER ";Rpl_center;" (dB)"
640  PRINT USING "13A,3D.3D,5A";"Ripple RIGHT ";Rpl_right;" (dB)"
650  RETURN                                    !
660  !
670 Thru_cal:                                  ! THRU CALIBRATION
680  DISP "Connect THRU standard and Press Continue."
690  PAUSE                                     !
700  DISP                                      !
710  OUTPUT @E5100;"CALI RESP"                 !
750  OUTPUT @E5100;"STANC?"                    !
780  ENTER @E5100;Tmp                          !
800  OUTPUT @E5100;"RESPDONE?"                 !
810  ENTER @E5100;Tmp                          !
830  RETURN                                    !
840  !
850  END                                       ! END OF PROGRAM
```

Sample Program -9: Analyzing a Crystal Resonator

This program measures the resonant frequency (Phase 0°) , the resonant impedance (Phase 0°) and the equalizing circuit constant of a crystal resonator.

The measured parameters are as follows:

The resonance parameter (**OUTPRESO?** is used.)	*Zr*,*Fr*
The equalizing circuit constant (**EQUCO?** and **EQUCPARA?** are used.)	C_0, C_1, L_1, R_1, G_0, R_0

Measurement conditions are as follows:

- Uses a PI-network test fixture.
- Sets the frequency range (center value[MHz] and span value[kHz]), the power level [μW] applied to a crystal at resonance that are specified by the user.
- Uses the list sweep to perform two types of measurements: the parallel capacity measurement point and the measurement near the resonant point.

Required Subroutines

Power_set:	The drive level [μW] of a crystal at resonance and the resonant impedance [Ω] are used to calculate the power level [dBm].
List_set:	Set up the list sweep table.
Pi_cal:	A calibration subroutine for a π fixture.
Printing:	Display analysis data on the screen.

Disk

This program is contained in the attached sample disk with the file name **FRCI_EQV** (for Instrument BASIC). The program for an external controller is not provided.

```
110  DIM Startf(1:2),Stopf(1:2),Ifbw(1:2),Nop(1:2),Power(1:2)
120 !
130 ASSIGN @E5100 TO 800
140 CLEAR @E5100
150 !
160 OUTPUT @E5100;"PRES"
170 OUTPUT @E5100;"CHAN1;ANAMODE ZTRAN;MEAS AR;FMT MAGZP;SCAY 1"
180 OUTPUT @E5100;"POWU WATT;PICIRC ON"
190 OUTPUT @E5100;"HOLD"
200 OUTPUT @E5100;"DUAC OFF"
210 OUTPUT @E5100;"COUC OFF"
220 OUTPUT @E5100;"DISG OFF"
230 OUTPUT @E5100;"DISAHIHB"
240 !
250 INPUT "Center frequency ? (MHz)",Center
260 Center=Center*1.E+6
270 INPUT "Span frequency ? (kHz)",Span
280 Span=Span*1000.
290 !
300 GOSUB List_set
310 !
320 GOSUB Pi_cal
330 !
340 EXECUTE "ANAOCH1"
350 EXECUTE "ANARFULL"
360 EXECUTE "ANAODATA"
370 !
380 DISP "CONNECT DEVICE, and PRESS CONTINUE."
390 PAUSE
400 DISP
410 !
420 !      MEASUREMENT
430 !
440 LOOP
450 !
460     EXECUTE "SING"
470 !
480     EXECUTE "OUTPRESO?"
490     Ci=READIO(8,0)
500     Fr=READIO(8,1)
510 !
520     WRITEIO 8,0;Center*.9
530     EXECUTE "EQUCO?"
540     CO=READIO(8,0)
550 !
560     EXECUTE "EQUCPARS4?"
570     C1=READIO(8,1)
580     L1=READIO(8,2)
590     R1=READIO(8,3)
600 !
610 GOSUB Printing
620 !
630 DISP "CONNECT NEXT DEVICE and Press Continue."
640 PAUSE
```

```
650  !
660  END LOOP
670  !
680  STOP
690  !
700  !
710 Pi_cal:  !
720   OUTPUT @E5100;"CALI ONEP"
730   !
740   DISP "Connect SHORT, and press [Continue]"
750   PAUSE
760   OUTPUT @E5100;"CLASS11B?"
770   ENTER @E5100;Tmp
780   !
790   DISP "Leave the ternimal OPEN, and press [Continue]"
800   PAUSE
810   OUTPUT @E5100;"CLASS11A?"
820   ENTER @E5100;Tmp
830   !
840   DISP "Connect LOAD, and press [Continue]"
850   PAUSE
860   OUTPUT @E5100;"CLASS11C?"
870   ENTER @E5100;Tmp
880   !
890   OUTPUT @E5100;"SAV1?"
900   ENTER @E5100;Tmp
910   !
920   RETURN
930   !
940   !
950 Printing:   !
960   !
970   CLEAR SCREEN
980   !
990   PRINT TABXY(1,1);"RESULT"
1000  PRINT USING "25A,3D.8D,6A";"RESONANCE FREQUENCY (Fr) ";Fr/1.E+6;" (MHz)"
1010  PRINT USING "14A,15X,5D.2D,6A";"CI (Zr) VALUE ";Ci;" (ohm)"
1020  PRINT
1030  PRINT USING "10A,5X,5D.3D,5A";"C0 VALUE ";C0*1.E+12;" (pF)"
1040  PRINT USING "10A,5X,5D.5D,5A";"C1 VALUE ";C1*1.E+12;" (pF)"
1050  PRINT USING "10A,5X,5D.3D,5A";"L1 VALUE ";L1*1000;" (mH)"
1060  PRINT USING "10A,5X,5D.3D,6A";"R1 VALUE ";R1;" (ohm)"
1070  !
1080  RETURN
1090  !
1100  !
1110 List_set:!
1120  Startf(1)=Center*.9
1130  Stopf(1)=Center*.9
1140  Nop(1)=1
1150  Ifbw(1)=1000
1160  Startf(2)=Center-Span/2
1170  Stopf(2)=Center+Span/2
1180  Nop(2)=200
1190  Ifbw(2)=1000
```

```
1200 !
1210 INPUT "POWER (uW)",W
1220 !
1230 OUTPUT @E5100;";EDITLIST;EDITLIS1;CLEL"
1240 FOR I=1 TO 2
1250   Power(I)=W/1.E+6
1260   OUTPUT @E5100;"SADD"
1270   OUTPUT @E5100;"STAR ";Startf(I)
1280   OUTPUT @E5100;"STOP ";Stopf(I)
1290   OUTPUT @E5100;"POIN ";Nop(I)
1300   OUTPUT @E5100;"IFBW ";Ifbw(I)
1310   OUTPUT @E5100;"POWE ";Power(I)
1320   OUTPUT @E5100;"SDON"
1330 NEXT I
1340 OUTPUT @E5100;"EDITDONE"
1350 OUTPUT @E5100;"SWPT LIST"
1360 OUTPUT @E5100;";LISDOBASE"
1370 RETURN
1380 !
1390 END
```

8

Command Reference

Common Command

*CLS

Clears the error queue, the status byte register, the event register of the standard operation status register structure, and the standard event status register. (no query)

Examples	`OUTPUT @E5100;"*CLS"`

*ESE <*value*>

Sets the enable bits of the standard status register.

Parameter Range	0 ~ 255 (decimal expression of enable bits of the operation status register)
Query Response	<*value*>

Examples	`OUTPUT @E5100;"*ESE 1"`
	
	`OUTPUT @E5100;"*ESE?"`
	`ENTER @E5100;A`

*ESR?

Outputs the contents of the standard event status register. (Query only)

Query Response	<*value*>
Examples	`OUTPUT @E5100;"*ESR?"` `ENTER @E5100;A`

*IDN?

Outputs the E5100A/B ID. (Query only)

*IDN?

Query Response	<*manufacturer*> <*model*> <*serial no.*> <*firmware rev.*> <*manufacturer*> : `Agilent Technologies` <*model*> : `E5100A` <*serial no.*> : serial number, such as `JP5KC00101` <*firmware rev.*> : firmware revison number, such as `REV3.00`
Examples	`OUTPUT @E5100;"*IDN?"` `ENTER @E5100;A$`

*OPC

Tells the analyzer to set bit 0 (Operation Complete bit) in the Standard Event Status Register when it completes all pending operations.

`*OPC?` query places an ASCII character 1 into the analyzer's output queue when all pending operations have been completed.

Query Response	{1}

Examples	`OUTPUT 717;"*OPC"`
	
	`OUTPUT 717;"*OPC?"`
	`ENTER 717;A`

*PCB <*value*>

Specifies the address of a controller that is temporarily passing GPIB control to the E5100A/B.

Parameter Range	0 to 30
Query Response	<*value*>
Examples	`OUTPUT @E5100;"*PCB 0"`

*RST

Resets the E5100A/B to its initial settings. See *Function Reference* for more information on default settings. (No Query)

*SRE <*value*>

Sets the enable bits of the status byte register.

Parameter Range	0 to 255 (decimal expression of enable bits of the status byte register)
Query Response	<*value*>

Examples	`OUTPUT @E5100;"*SRE 1"`
	
	`OUTPUT @E5100;"*SRE?"`
	`ENTER @E5100;A`

*STB?

Reads the status byte by reading the master summary status bit.

Query Response	*<value>*
Examples	`OUTPUT @E5100;"*STB?"` `ENTER @E5100;A`

*TRG

Triggers the E5100A/B when the trigger mode is set to EXTERNAL trigger. (No Query)

Examples	To start measurement `OUTPUT @E5100;"*TRG"`

*TST?

Executes a power-on self test and returns the test result. (No Query)

Query Response	`{0|1}` 0 : No error
Examples	`OUTPUT @E5100;"*TST?"` `ENTER @E5100;A`

*WAI

Makes the E5100A/B wait until all previously sent commands are completed. (No Query)

ADTOTRAC

Starts data process. This command is only used for the parallel processing using with TRIGMEAS and MOVADARY. (No Query)

ANAMODE {GAINP|ZREFL|ZTRAN}

Select measurement function.

Equivalent Key Sequence	(Meas/Format) FUNCTION [] GAIN-PHASE, IMPEDANCE:Refl, Trans
Parameter Description	GAINP : Gain-Phase measurement ZREFL : Reflection impedance measurement ZTRAN : Transmission impedance measurement
Query Response	{ GAINP | ZREFL | ZTRAN }

Examples	`OUTPUT @E5100;"ANAMODE GAINP"`
	
	`OUTPUT @E5100;"ANAMODE?"`
	`ENTER @E5100;A$`

ANAOCH{1-4}

Selects channel for waveform analysis. For details, refer to Appendix D. (No Query)

Examples	`OUTPUT 717;"ANAOCH1"`

ANAODATA

Selects a data trace for waveform analysis. For details, refer to Appendix D. (No Query)

Examples	`OUTPUT 717;"ANAODATA"`
	
	`OUTPUT 717;"ANAODATA?"`
	`ENTER 717;A`

ANAOMEMO

Selects a sub-trace for waveform analysis. For details, refer to Appendix D. (No Query)

ANAPOINS?␣<*value*>

Exchange the measurement point(1 to NOP) to the stimulus value (query only). See `POIN`.

Parameter Description	<*value*> : Measurement Point (1 to NOP)
Parameter Range	<*value*> : 1 to NOP
Query Response	{*value*}
Examples	`OUTPUT @E5100;"ANAPOINS? 10"` `ENTER @E5100;A`

ANARANG␣<*value1*>,<*value2*>

Sets the waveform analysis stimulus range by entering the START (*value1*) and STOP (*value2*) values. For details, refer to Appendix D.

Parameter Range	<*value1*> : 10×10^3 (=10k) to 300×10^6 (=300M) Hz <*value2*> : 10×10^3 (=10k) to 300×10^6 (=300M) Hz
Query Response	{*value1*}{*value2*}
Examples	`OUTPUT 717;"ANARANG 1000,10000"`

ANARANGP␣<*value1*>,<*value2*>

Sets the waveform analysis stimulus range by entering the point number of START and point number of STOP values.

Parameter Description	<*value1*> : The point number of START (1 to NOP) <*value2*> : The point number of STOP (1 to NOP)
Query Response	{*value1*}{*value2*}

Examples	`OUTPUT @E5100;"ANARANGP 10,100"`
	
	`OUTPUT @E5100;"ANARANGP?"`
	`ENTER @E5100;A, B`

ANARFULL

Sets the analysis range equal to the full stimulus range. For details, refer to Appendix D. (No Query)

ANASTIMP?␣<*value*>

Exchange the stimulus value to the nearest measurement point on the channel specified by ANAOCH (query only). See POIN.

Parameter Description	<*value*> : Measurement Point (1 to NOP)
Parameter Range	<*value*> Frequency Sweep : 10 x 10^3 1 to NOP Power Sweep : Power Range
Query Response	{*value*}
Examples	`OUTPUT @E5100;"ANASTIMP? 100E6"` `ENTER @E5100:A`

ATRC␣{1|2}

Selects the active trace for the marker function.

Equivalent Key Sequence	(Marker) ACTIVE TRC [ ]
Parameter Description	1: Main trace 2: Sub trace
Query Response	{1|2}

Examples	`OUTPUT @E5100;"ATRC"`
	
	`OUTPUT @E5100;"ATRC?"`
	`ENTER @E5100;A`

ATTI{A|B|C|R|}␣{0|25}

Selects the attenuator value at an input port.

Equivalent Key Sequence	(System) ATTENUATOR, PORT:A, PORT:B, PORT:C, PORT:R 0dB, 25dB
Parameter Description	0 : 0 dB 25 : 25 dB
Query Response	{0|25}

Examples	`OUTPUT @E5100;"ATTIA 25"`
	
	`OUTPUT @E5100;"ATTIA?"`
	`ENTER @E5100;A`

ATTIAAUTO␣{OFF|ON|0|1}

Select the attenuator auto mode at the port A ON or OFF. The auto mode automatically selects the proper attenuator value.

Equivalent Key Sequence	(System) ATTENUATOR PORT:A AUTO
Parameter Description	OFF or 0 : AUTO OFF ON or 1 : AUTO ON
Query Response	{0|1}
Examples	OUTPUT @E5100;"ATTIAAUTO ON" OUTPUT @E5100;"ATTIAAUTO?" ENTER @E5100;A

ATTIBAUTO␣{OFF|ON|0|1}

Select the attenuator auto mode at the port B ON or OFF. The auto mode automatically selects the proper attenuator value.

Equivalent Key Sequence	(System) ATTENUATOR PORT:B AUTO
Parameter Description	OFF or 0 : AUTO OFF ON or 1 : AUTO ON
Query Response	{0|1}

ATTICAUTO␣{OFF|ON|0|1}

Select the attenuator auto mode at the port C ON or OFF. The auto mode automatically selects the proper attenuator value.

Equivalent Key Sequence	(System) ATTENUATOR PORT:C AUTO
Parameter Description	OFF or 0 : AUTO OFF ON or 1 : AUTO ON
Query Response	{0|1}

ATTIRAUTO␣{OFF|ON|0|1}

Select the attenuator auto mode at the port R ON or OFF.

Equivalent Key Sequence	(System) ATTENUATOR PORT:R AUTO
Parameter Description	OFF or 0 : AUTO OFF ON or 1 : AUTO ON
Query Response	{0|1}

ATTW␣<*value*>

Sets the waiting time when the attenuator switch is changed at the power sweep mode. (Option 010 only, No warning will be shown even if this command is executed with the E5100A/B with no option 010.) The power-on default setting is 1 ms.

Parameter Range	0 to 100 sec.
Query Response	{*value*} (sec)

Examples	`OUTPUT 717;"ATTW 10"`
	
	`OUTPUT 717;"ATTW?"`
	`ENTER 717;A`

AUTO

Selects the scale/div value automatically to fit the trace data to the display.

Equivalent Key Sequence	(Display) `AUTO SCALE`
Examples	`OUTPUT 717;"AUTO"`

BASL

Displays the softkey lable defined by HP Instrument BASIC. (No Query)

Equivalent Key Sequence	(System) `I-BASIC` `ON KEY LABEL`
Examples	`OUTPUT @E5100;"BASL"`

BEEPDONE␣{OFF|ON|0|1}

Sets the operation completion beeper ON or OFF.

Equivalent Key Sequence	(System) `MORE` `BEEP DONE ON off`
Parameter Description	`OFF` or `0` : operation completion beeper OFF `ON` or `1` : operation completion beeper ON
Query Response	{0|1}

Examples	`OUTPUT 717;"BEEPDONE ON"`
	
	`OUTPUT 717;"BEEPDONE?"`
	`ENTER 717;A`

BEEPFAIL␣{OFF|ON|0|1}

Sets the limit fail beeper ON or OFF.

Equivalent Key Sequence	(System) LIMIT MENU BEEP FAIL ON off
Parameter Description	OFF or 0 : limit fail beeper OFF ON or 1 : limit fail beeper ON
Query Response	{0|1}

Examples	`OUTPUT 717;"BEEPFAIL ON"`
	
	`OUTPUT 717;"BEEPFAIL?"`
	`ENTER 717;A`

BEEPWARN␣{OFF|ON|0|1}

Sets the warning beeper ON or OFF.

Equivalent Key Sequence	(System) MORE BEEP WARN ON off
Parameter Description	OFF or 0 : warning beeper OFF ON or 1 : warning beeper ON
Query Response	{0|1}

Examples	`OUTPUT 717;"BEEPWARN ON"`
	
	`OUTPUT 717;"BEEPWARN?"`
	`ENTER 717;A`

BINSIZE␣<*value*>

Specify the number of continuous data outputted to the I/O port by `INPUTRACB`.

Specify the maximum value in the second parameter of `INPUTRACB`. (Option 022 only)

Parameter Range	0 to 6
Query Response	{*value*}

Examples	`OUTPUT @E5100;"BINSIZE 2"`
	
	`OUTPUT @E5100;"BINSIZE?"`
	`ENTER @E5100;A`

BLIGHT␣{OFF|ON|0|1}

Sets backlighting the LCD screeen ON or OFF.

Parameter Description	`OFF` : Backlighting OFF `ON` : Backlighting ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"BLIGHT OFF"`
	
	`OUTPUT @E5100;"BLIGHT?"`
	`ENTER @E5100;A`

BOTV␣*<value>*

Sets the value at the bottom line of the graticule.

Equivalent Key Sequence	(Display) `BOTTOM VALUE`
Parameter Range	-10^9 to 10^9
Query Response	{*value*}

Examples	`OUTPUT @E5100;"BOTV 100"`
	
	`OUTPUT @E5100;"BOTV?"`
	`ENTER @E5100;A`

CALCOPY␣*<value1>*, *<value2>*

Copy CAL data and CAL settings between two channels. (No Query)

Parameter Description	*<value1>* : Source channel *<value2>* : Destination channel
Examples	`OUTPUT @E5100;"CALCOPY 1,2"`

CALI␣{NONE|RESP|RAI|ONEP}

Selects the measurement calibration type. (Query)

Equivalent Key Sequence	(Cal) CALIBRATE:NONE, RESPONSE, RESPONSE & ISOL'N, 3 TERM
Parameter Description	NONE : No calibration RESP : Response measurement calibration RAI : Response and isolation measurement calibration ONEP : 1-port 3-term measurement calibration
Query Response	{NONE|RESP|RAI|ONEP}

Examples	`OUTPUT @E5100;"CALI NONE"`
	
	`OUTPUT @E5100;"CALI?"`
	`ENTER @E5100;A$`

CALK{O|S|L}{LS|RS|CP}␣<*value*>

Sets the value of calibration kit of the acitive measurement mode.

Equivalent Key Sequence	(Cal) MODIFY CAL KIT OPEN STD, SHORT STD, LOAD STD, Rs, Ls, Cp
Parameter Range	O : OPEN S : SHORT L : LOAD LS : L_s RS : R_s CP : C_p
Query Response	{*value*}

Examples	`OUTPUT @E5100;"CALKLRS 49.9"`
	
	`OUTPUT @E5100;"CALKLRS?"`
	`ENTER @E5100;A`

CENS␣<*value1*>,<*value2*>

Sets center and span stimulus value.

Parameter Description	<*value1*> : Center value <*value2*> : Span value
Parameter Range	<*value1*> : 10×10^3 (=10k) to 300×10^6 (=300M) Hz <*value2*> : 0 to 2.9999×10^8 (=299.99M) [Hz]
Examples	`OUTPUT @E5100;"CENS 199.95MHZ,100KHZ"`

CENT␣<*value*>

Sets the center stimulus value.

Equivalent Key Sequence	(Center)
Parameter Range	10×10^3 (=10k) to 300×10^6 (=300M) Hz
Query Response	{*value*}

Examples	`OUTPUT @E5100;"CENT 99.95MHZ"`
	
	`OUTPUT @E5100;"CENT?"`
	`ENTER @E5100;A`

CHAD␣<*string*>

Changes the current directory.

Equivalent Key Sequence	(Save/Recall) FILE UTILITY CHANGE DIRECTORY
Parameter Description	<*string*> :Directory path name
Examples	`OUTPUT @E5100;"CHAD "".."""`

CHAN␣{1|2|3|4}

Selects the active measurement channel. (No Query)

Equivalent Key Sequence	(Meas/Format) ACTIVE CH [ ]
Examples	`OUTPUT 717;"CHAN 1"`

CIN

Sets port C of the 24-bit I/O port to be an input port.

Examples	`OUTPUT 717;"CIN"`

CIVAL␣<*value*>

Enters the CI value.

Equivalent Key Sequence	(MENU2) USER CI
Parameter Range	10^{-3} to 1 G [Ω]
Query Response	{*value*}

Examples	`OUTPUT @E5100;"CIVAL 10"`
	
	`OUTPUT @E5100;"CIVAL?"`
	`ENTER @E5100;A`

CLACT␣<*Value*>

Uses this command to specify the load capacitance in [F]. This load capacitance must be the actual capacitance connected to the crystal resonator when you measure through search the load resonance frequency (FL) of a resonator using the SRCHTRFL? command. CL boards that are used respectively for the 41900A π network test fixture and the 41901A SMD π network test fixture come with a precisely measured fixed C. Be sure to enter these C values.

See the sample programs in Appendix K for more information.

Parameter Range	0 to 3.402824E+36 Unit: F Default: 0
Query Response	{*Value*} {*Value*} : Load capacitance in [F] to be specified (capacitance actually connected to the resonator when you make search using the SRCHTRFL? command)
Example	`OUTPUT @E5100;"CLACT 1.95E-11"`

CLASS11{A|B|C}

Selects the calibration standard class (S11A (OPEN), S11B (SHORT), or S11C (LOAD)) for the S11 1-port calibration and measures the calibration data.

Equivalent Key Sequence	(Cal) [S11] : OPEN , SHORT , LOAD
Parameter Description	A: OPEN B: SHORT C: LOAD
Examples	`OUTPUT @E5100;"CLASS11A"`

Returns 1 when the measurement for calibration data is completed by using the CLASS11{A|B|C}? command.

CLEL

Clears the current frequency list. (No Query)

Equivalent Key Sequence	(Sweep)
Examples	`OUTPUT @E5100;"CLEL"`

CLEM{1-4}

Clears the marker. (No Query)

Equivalent Key Sequence	(Marker) CLEAR MARKER
Examples	`OUTPUT @E5100;"CLEM1"`

CLEMNU3

Clears the softkey definitions of (Menu3). (No Query)

Examples	`OUTPUT @E5100;"CLEMNU3"`

CLES

Clears the status byte, the event status register, the event status register B, and the operational status register.

(No Query)

Examples	`OUTPUT @E5100;"CLES"`

CLOSE

Returns a file, which has been read/write-enabled using the `ROPEN` command or `WOPEN` command, to access-disabled status. If this command is executed before reading process using the `READ?` command completes, an error occurs.

Generally, this command is used in combination with the `ROPEN` command and `READ?` command or the `WOPEN` command and the `WRITE` command, as shown in Figure J-1. (No query)

CLTGT␣<*Value*>

Use this command to specify the target load capacitance in [F]. This capacitance must be the desired target capacitance when you measure the load resonance frequency (FL) of a resonator through search using the `SRCHTRFL?` command. The `SRCHTRFL?` command determines the load resonance frequency for the specified load capacitance.

See the sample programs in Appendix K for more information.

Parameter Range	0 to 3.402824E+36 Unit: F Default: 0
Query Response	{*Value*} {*Value*} : Target load capacitance in [F] to be specified (capacitance used when you make search using the `SRCHTRFL?` command)
Example	`OUTPUT @E5100;"CLTGT 2.E-11"`

CONT

Continuous trigger. (No Query)

Equivalent Key Sequence	(Trigger) `CONTINUOUS`
Examples	`OUTPUT @E5100;"CONT"`

CONV␣{OFF|ZTRA|YTRA|ZREF|YREF}

Selects the measurement data conversion setting. This command is available when the `ANAMODE` is set to Gain-Phase mode.

Equivalent Key Sequence	(Meas/Format) `FUNCTION [ ]`
Parameter Description	OFF : Conversion OFF ZTRA : Z:transmission YTRA : Y:transmission ZREF : Z:Reflection YREF : Y:Reflection
Query Response	{OFF|ZTRA|YTRA|ZREF|YREF}

Examples	`OUTPUT @E5100;"CONV ZTRA"`
	
	`OUTPUT @E5100;"CONV?"`
	`ENTER @E5100;A$`

COPYRIGHT␣{OFF|ON|0|1}

Turn ON/OFF the copyright notice. (No Query)

Parameter Range	OFF or 0 : Copyright notice OFF ON or 1 : Copyright notice ON
Examples	`OUTPUT @E5100;"COPYRIGHT OFF"`

CORR␣{OFF|ON|0|1}

Sets the error correction function ON or OFF.

Equivalent Key Sequence	(Cal) CORRECTION ON off
Parameter Range	OFF or 0 : error correction function OFF ON or 1 : error correction function ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"CORR OFF"`
	
	`OUTPUT @E5100;"CORR?"`
	`ENTER @E5100;A`

CORRS?

Outputs the status of the error correction function on the active channel. (Query only)

Query Response	{OFF|Cor|C?|C!}
Examples	`OUTPUT @E5100;"CORRS?"` `ENTER @E5100;A$`

COUC␣{OFF|ON|0|1}

Sets the channel coupling of stimulus values ON or OFF.

Equivalent Key Sequence	(Sweep) COUPLED CH ON off
Parameter Description	OFF or 0 : channel coupling OFF ON or 1 : channel coupling ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"COUC OFF"`
	
	`OUTPUT @E5100;"COUC?"`
	`ENTER @E5100;A`

COUT

Sets port C of the 24-bit I/O port to be an output port.

Examples	`OUTPUT @E5100;"COUT"`

CRED␣<*string*>

Create a directory (only MS-DOS format). (No Query)

Equivalent Key Sequence	(Save/Recall) `FILE UTILITY` `CREATE DIRECTORY`
Parameter Description	<*string*> :Up to 8 characters for directory name and up to 3 characters for extension
Examples	`OUTPUT @E5100;"CRED ""DATA"""`

CURD?

Outputs current directory. (Query only)

Equivalent Key Sequence	(Save/Recall) `FILE UTILITY` `CURRENT DIRECTORY`
Query Response	{*string*}
Examples	`OUTPUT @E5100;"CURD?"` `ENTER @E5100;A$`

CURMPOIN?

Outputs the latest measurement point number and measurement data. (Query only)

Query Response	{*numeric1*}{*numeric2*} {*numeric1*} : measurement point number {*numeric2*} : Formatted data (complex value , data format: real, imaginary)
Examples	`OUTPUT @E5100;"CURMPOIN?"` `ENTER @E5100;A,B,C`

CWFREQ␣<*value*>

Sets the frequency for power sweep.

CWFREQ␣<*value*>

Equivalent Key Sequence	(Sweep) CWFREQ
Parameter Range	10×10^3 (=10k) to 300×10^6 (=300M) Hz
Query Response	{*value*}

Examples	`OUTPUT @E5100;"CWFREQ 100KHZ"`
	
	`OUTPUT @E5100;"CWFREQ?"`
	`ENTER @E5100;A`

DATAM␣{NONE|RESP|RAI|ONEP}

Select a type for 1 point correction on the active channel. See CORR command.

Parameter Description	NONE : No RESP : Response calibration RAI : Response & isolation calibration ONEP : 3-term calibration
Query Response	{NONE|RESP|RAI|ONEP}

Examples	`OUTPUT @E5100;"DATAM NONE"`
	
	`OUTPUT @E5100;"DATAM?"`
	`ENTER @E5100;A$`

DATAMN␣{1-10}

Set an index of data space for 1 point correction on the active channel.

Query Response	{value}

Examples	`OUTPUT @E5100;"DATAMN 1"`
	
	`OUTPUT @E5100;"DATAMN?"`
	`ENTER @E5100;A`

DATI

Saves the data trace to the memory trace. (No Query)

Equivalent Key Sequence	(Display) DEFINE TRACE DATA->MEM
Examples	`OUTPUT @E5100;"DATI"`

DAYMYEAR

Sets the displayed date mode to day/month/year order. (No Query)

Equivalent Key Sequence	(System) MORE SET CLOCK DayMonYear
Examples	`OUTPUT @E5100;"DAYMYEAR"`

DELA

This command equals `FMT DELA`. See `FMT`.

DELO

Sets the delta marker mode OFF. (No Query)

Equivalent Key Sequence	(Marker) Δ MODE MENU Δ MODE OFF
Examples	`OUTPUT @E5100;"DELO"`

DELR{1-4}

Selects the delta reference marker.

Equivalent Key Sequence	ΔMODE MENU ΔREF MKR ΔREF=1 to ΔREF=4
Query Response	{*value*}

Examples	`OUTPUT @E5100;"DELR1"`
	
	`OUTPUT @E5100;"DELR1?"`
	`ENTER @E5100;A`

DELRFIXM

Sets the user-specified fixed reference marker. (No Query)

Equivalent Key Sequence	(Marker) ΔMODE MENU Δ REF=Δ FIXED MKR
Examples	`OUTPUT @E5100;"DELRFIXM"`

DIN

Set port D of the 24-bit I/O port to be an input port. (No Query)

Examples	`OUTPUT @E5100;"DIN"`

DISA␣{ALLI|HIHB|ALLB|BASS}

Selects the display allocation mode.

Equivalent Key Sequence	(Display) DISPLAY ALLOCATION
Parameter Description	ALLI : All instrument HIHB : Half instrument half BASIC ALLB : All BASIC BASS : BASIC status
Query Response	{ALLI|HIHB|ALLB|BASS}

Examples	`OUTPUT @E5100;"DISA HIHB"`
	
	`OUTPUT @E5100;"DISA?"`
	`ENTER @E5100;A$`

DISG␣{OFF|ON|0|1}

Sets the graticule display on or off.

Equivalent Key Sequence	(Display) MORE GRATICLE ON off
Parameter Description	OFF or 0 : Graticule display OFF ON or 1 : Graticule display ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"DISG ON"`
	
	`OUTPUT @E5100;"DISG?"`
	`ENTER @E5100;A`

DISP␣{DATA|MEMO|DATM|DDM|DMM}

Selects the display trace type.

Equivalent Key Sequence	(Display) MORE MORE TRACE:DATA, MEMORY, DATA and MEMORY, DATA-MEM, DATA/MEM
Parameter Range	DATA : Data only MEMO : Memory only DATM : Data and memory DDM : Data divided by memory DMM : Data minus memory
Query Response	{DATA|MEMO|DATM|DDM|DMM}

Examples	`OUTPUT @E5100;"DISP DATA"`
	
	`OUTPUT @E5100;"DISP?"`
	`ENTER @E5100;A$`

DIST␣{OFF|ON|0|1}

Sets the trace display on or off.

(TRACE on off under (DISPLAY); Query)

Parameter Description	OFF or 0 : trace display OFF ON or 1 : trace display ON
Query Response	{0|1}
Examples	`OUTPUT @E5100;"DIST ON"`

DONE

Completes the measurement of the selected standard calibration. (No Query)

Equivalent Key Sequence	(Cal) DONE
Examples	`OUTPUT @E5100;"DONE"`

DOUT

Sets D port of the 24-bit I/O port to output port. (No Query)

Examples	`OUTPUT @E5100;"DOUT"`

DSKEY

Disables the front panel keys and the rotary knob. To enable the keys and knob again, send the ENKEY command.

Examples	`OUTPUT @E5100;"DSKEY"`

DUAC␣{OFF|ON|0|1}

Selects the dual (ON) or single (OFF) channels display.

Equivalent Key Sequence	(Display) MULTI CHAN ON off
Parameter Description	OFF or 0 : Active channel only ON or 1 : Dual channel
Query Response	{0|1}

Examples	`OUTPUT @E5100;"DUAC ON"`
	
	`OUTPUT @E5100;"DUAC?"`
	`ENTER @E5100;A`

EDITDONE

Completes editing the frequency list for the list sweep. (No Query)

Equivalent Key Sequence	(Sweep) LIST DONE
Examples	`OUTPUT 717;"EDITDONE"`

EDITLIS1

Selects list 1 for editing. (No Query)

Equivalent Key Sequence	(Sweep) LIST NO [ ] EDIT LIST
Examples	`OUTPUT @E5100;"EDITLIS1"`

EDITLIS2

Selects list 2 for editing. (No Query)

Equivalent Key Sequence	(Sweep) LIST NO [ ] EDIT LIST
Examples	`OUTPUT @E5100;"EDITLIS2"`

EDITLIST

Begins editing the frequency list. (No Query)

Equivalent Key Sequence	(Sweep) EDIT LIST
Examples	`OUTPUT @E5100;"EDITLIST"`

ELED␣<*value*>

Sets the electrical delay.

Equivalent Key Sequence	(Display) MORE ELECTRICAL DELAY
Parameter Range	−10 to 10 [sec.]
Query Response	{*value*}

Examples	`OUTPUT @E5100;"ELED 0"`
	
	`OUTPUT @E5100;"ELED?"`
	`ENTER @E5100;A`

ENKEY

Re-enables the front panel keys and the rotary knob which have been disabled by the DSKEY command. (No Query)

EQUC0?␣<*value*>

Returns C_0 of the equivalent circuit of the resonator at specified frequency. C_0 is calculated by using the following equation:

$$C_0 = \frac{B_s}{\omega_s}$$

Where,

B_s	Imaginary part on f_s
ω_s	$2 \times \pi \times f_s$
f_s	Frequency which is specified as command parameter

If Z-conversion is selected, C_0 is calculated by using following equations:

EQUC0?␣<*value*>

$$C_0 = \frac{-1}{B_s \times \omega_s}$$

This command is only available when LOG MAG & Phase format is selected. If another format is selected, `0` will be returned. If the specified frequency is out of analysis range, `0` will be returned.

If B_s is `0` when the Z-conversion is activated, `EQUC0?` returns `0`.

(Query only)

Parameter Range	1000 (=1k) to 3.0×10^8 (=300M) [Hz]	
Query Response	{*value*(C_0)}	

Examples	`OUTPUT @E5100;"EQUC0? 100MHZ"`
	`ENTER @E5100;C0`
	`PRINT "C0=",C0`
		Query C_0 *at 100 MHz.*
		Receive the returned C_0.
		Display C_0 *on the CRT.*

EQUCPARA5?

Executes the equivalent circuit analysis for a resonator using the same equivalent circuit of `EQUCPARA?`, but does not output G0. `EQUCPARA5?` does not display the warning even an anti-resonance frequency is not in the analysis range. (Query only)

Query Response	{*value1*},{*value2*},{*value3*},{*value4*},{*value5*} {*value1*}:C0 {*value2*}:C1 {*value3*}:L1 {*value4*}:R1 {*value5*}:R0
Examples	`OUTPUT @E5100;"EQUCPARA5?"` `ENTER @E5100;A,B,C,D,E`

EQUCPARA?

Executes four element analysis of a crystal resonator, and outputs parameters, C_0, C_1, L_1, and R_1. For more information, refer to “EQUCPARA?” in Appendix D. (Query only)

Equivalent Key Sequence	(Menu 2) `EQUIVALENT CKT 6`
Query Response	{*numeric1*}, {*numeric2*}, {*numeric3*}, {*numeric4*}, {*numeric5*}, {*numeric6*} <*numeric1*> : C_0 <*numeric2*> : C_1 <*numeric3*> : L_1 <*numeric4*> : R_1 <*numeric5*> : G_0 <*numeric6*> : R_0
Examples	`OUTPUT @E5100;"EQUCPARA?"` `ENTER @E5100;A,B,C,D`

EQUCPARS4?

Executes four element analysis of a crystal resonator, and outputs parameters, C_0, C_1, L_1, R_1, f_s, f_a, f_r, f_1, f_2. (Query only)

Equivalent Key Sequence	(Menu 2) EQUIVALENT CKT 4
Query Response	{*numeric1*}, {*numeric2*}, {*numeric3*}, {*numeric4*}, {*numeric5*}, {*numeric6*}, {*numeric7*}, {*numeric8*}, {*numeric9*} <*numeric1*> : C_0 <*numeric2*> : C_1 <*numeric3*> : L_1 <*numeric4*> : R_1 <*numeric5*> : f_s <*numeric6*> : f_a <*numeric7*> : f_r <*numeric8*> : f_1[1] <*numeric9*> : f_2[1]
Examples	`OUTPUT @E5100;"EQUCPARS4?"` `ENTER @E5100;A,B,C,D,E,F,G,H,I`

1 $f_1 < f_2$

EQUCPARS?

Executes four elements analysis of a crystal resonator, and outputs parameters, C_0, C_1, L_1, R_1, f_s, f_a, f_r, f_1, and f_2. For more information, refer to "EQUCPARS?" in Appendix D. (Query only)

Equivalent Key Sequence	(SPCL FNCT) EQUIVALENT CKT
Query Response	{*value1*}, {*value2*}, {*value3*}, {*value4*}, {*value5*}, {*value6*}, {*value7*}, {*value8*}, {*value9*}, {*value10*}, {*value11*} <*value1*> : C_0 <*value2*> : C_1 <*value3*> : L_1 <*value4*> : R_1 <*value5*> : f_s <*value6*> : f_a <*value7*> : f_r <*value8*> : f_1[1] <*value9*> : f_2[1] <*value10*> : G_0 <*value11*> : R_0
Examples	`OUTPUT 717;"EQUCPARS?"` `ENTER 717;A,B,C,D,E,F,G,H,I`

1 $f_1 < f_2$

EQUM␣<*value*>

Specifies how many points are used for an approximation of a circle for `EQUCPARA?` and `EQUCPARS?` command. The default value is 8. For a detail information about `EQUM`, refer to "EQUCPARA?" in Appendix D.

Parameter Range	2 to 801
Query Response	{*value*}

Examples	`OUTPUT @E5100;"EQUM 30"`
	
	`OUTPUT @E5100;"EQUM?"`
	`ENTER @E5100;A`

ESB?

Outputs the event status register B value. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"ESB?"` `ENTER @E5100;A`

ESNB␣<*value*>

Specifies the bits of event status register B.

Parameter Range	0 to 32767 ($=2^{15}-1$) (decimal expression of the event status register)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"ESNB 0"`
	
	`OUTPUT @E5100;"ESNB?"`
	`ENTER @E5100;A`

EXET␣<*value*>

Executes the service test. (No Query)

Equivalent Key Sequence	(System) EXECUTE TEST
Parameter Description	{*value*} : test number
Examples	`OUTPUT @E5100;"EXET 11"`

EXPC␣{OFF|ON|0|1}

Sets the interpolated error correction ON or OFF.

Parameter Description	OFF or 0 : interpolation OFF ON or 1 : interpolation ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"EXPC OFF"`
	
	`OUTPUT @E5100;"EXPC?"`
	`ENTER @E5100;A`

EXPZP␣{OFF|ON|0|1}

Selects the expanded phase format for the impedance measurement.

Equivalent Key Sequence	(Meas/Format) FORMAT EXP PHASE ON OFF
Parameter Description	ON or 1 : expanded phase format ON OFF or 0 : expanded phase format OFF
Query Response	{1|0}

Examples	`OUTPUT @E5100;"EXPZP ON"`
	
	`OUTPUT @E5100;"EXPZP"`
	`ENTER @E5100;A`

EXTRLOCK?

Outputs the state of the external reference (locked or unlocked). (Query only)

Parameter Range	0 : unlocked 1 : locked
Query Response	{0|1}
Examples	`OUTPUT @E5100;"EXTRLOCK?"` `ENTER @E5100;A`

EXTT␣{OFF|ONSWEE}

Selects the external trigger mode.

EXTT␣{OFF|ONSWEE}

Equivalent Key Sequence	(Trigger) TRIG EVENT [ ], INT. TRIG ON SWEEP
Parameter Range	OFF : External trigger OFF (internal trigger mode ON) ONSWEE : External trigger on sweep
Query Response	{OFF|ONSWEE}

Examples	`OUTPUT @E5100;"EXTT OFF"`
	
	`OUTPUT @E5100;"EXTT?"`
	`ENTER @E5100;A$`

FILC␣*<string1>*,*<string2>*,*<string3>*,*<string4>*

Copies file on flexible and RAM disks. (No Query)

Equivalent Key Sequence	(SAVE) FILE UTILITY COPY FILE
Parameter Description	*<string1>* : Source file name. String. *<string2>* : Source device name. "MEMORY" or "DISK"[1] *<string3>* : Destination file name. *<string4>* : Destination device name.[1]
Examples	`OUTPUT 717;"FILC ""DAT1.TXT"",""MEMORY"",""DAT1.TXT"",""DISK"""`

1 Internal flexible disk drive:DISK; Internal RAM disk drive:MEMORY

FMT␣{LOGM|PHAS|DELA| ... |ADMG|ADMB|MAGZDF}

Selects the display format.

Equivalent Key Sequence	(Meas/Format) FORMAT, LOG MAG & PHASE, LOG MAG & DELAY, LIN MAG & PHASE, LIN MAG & DELAY, REAL & IMAGINARY, LOG MAG, LIN MAG, PHASE, DELAY, REAL, IMAGINARY, EXPANDED PHASE, |Z| & PHASE z, |Y| & PHASE y, R-X, G-B, |Z|, |Y|, PHASE z, PHASE y, R, G, B, X
Parameter Description	LOGM : Log magnitude format PHAS : Phase format DELA : Delay format LINM : Linear magnitude format REAL : Real format IMAG : Imaginary format LOGMP : Log magnitude and phase format LOGMD : Log magnitude and delay format LINMP : Linear magnitude and phase format LINMD : Linear magnitude and delay format EXPP : Expanded pahse format MAGZP : |Z| & phase format MAGYP : |Y| & phase format IMPRX : R-X format ADMGB : G-B format MAGZ : |Z| format MAGY : |Y| format PHAZ : Phase_z format PHAY : Phase_y format IMPR : R format IMPX : X format ADMG : G format ADMB : B format MAGZDF : Z-Δ format for tracking measurement (option 023 only)
Query Response	{LOGM | PHAS | DELA | LINM | REAL | IMAG | LOGMP | LOGMD | LINMP | LIMMD | EXPP | MAGZP | MAGYP | IMPRX | ADMGP | MAGZ | MAGY | PHAZ | PHAY | IMPR | IMPX | ADMG | ADMB | MAGZDF}

Examples	`OUTPUT 717;"FMT LOGM"`
	
	`OUTPUT 717;"FMT?"`
	`ENTER 717;A$`

FNAME?␣<*value*>

Returns the file name corresponding to a specified number in the current directory. To each file, a number is assigned from 1 to "the number of the files" in alphabetical order. Use the FNUM? command to verify the number of the files in the current directory. (Query only)

Parameter	Description	Range
<*value*>	Specified file No.	1 to "the number of the files in the current directory"

■ Query Response

{*string*} <new line><^END>

FNAME?␣<*value*>

FNUM?

Returns the number of the files in the current directory. (Query only)

- Query Response

{*value*} <new line><^END>

FSIZE?␣<*string*>

Returns the size of a specified file in bytes. If the file does not exist, this command returns -1. (Query only)

Parameter	Description
<*string*>	File name of up to 12 characters including its extension

- Query Response

{*value*} <new line><^END>

GRAPCOL␣{FIXC|VARC|MONO}

Selects the color of the display image file to be saved on the disk drive.

Parameter Description	{FIXC|VARC|MONO} FIXC : Fixed color VARC : Variable color (Same color as the display) MONO : Monocrome
Query Response	{FIXC|VARC|MONO}

Examples	`OUTPUT @E5100;"GRAPCOL FIXC"`
	
	`OUTPUT @E5100;"GRAPCOL?"`
	`ENTER @E5100;A$`

GRAPFORM␣{PCL|TIFF}

Selects the graphic format to be save the display image on the disk drive.

Equivalent Key Sequence	(Save/Recall) GRAPH FMT [ ]
Parameter Description	{PCL|TIFF} PCL : PCL TIFF : TIFF
Query Response	{PCL|TIFF}

Examples	`OUTPUT @E5100;"GRAPFORM TIFF"`
	
	`OUTPUT @E5100;"GRAPFORM?"`
	`ENTER @E5100;A$`

GRODAPER <*value*>

Sets the group delay aperture.

Equivalent Key Sequence	(Meas/Format) GROUP DELY APERTURE
Parameter Range	1 to 200 (% of span) [%]
Query Response	{*value*}

Examples	`OUTPUT @E5100;"GRODAPER 30"`
	
	`OUTPUT @E5100;"GRODAPER?"`
	`ENTER @E5100;A`

HOLD

Holds the present measurement.

Equivalent Key Sequence	(Trigger) HOLD
Query Response	{1|0} 0 : Sweep mode 1 : Hold mode

Examples	`OUTPUT @E5100;"HOLD"`
	
	`OUTPUT @E5100;"HOLD?"`
	`ENTER @E5100;A`

IDN?

Outputs the analyzer ID.

Query Response	`Agilent Technologies,E5100A,JPXKCmmmmm,n.nn` mmmmm: serial number n.nn: revision number
Examples	`OUTPUT @E5100;"IDN?"` `ENTER @E5100;A$`

IFBW␣<*value*>

Sets the bandwidth value for IF bandwidth reduction.

IFBW␣<*value*>

Equivalent Key Sequence	(Sweep) IF BW
Parameter Range	2, 20, 200, 1000, 4000, 8000 [Hz]
Query Response	{*value*}

Examples	`OUTPUT @E5100;"IFBW 200HZ"`
	
	`OUTPUT @E5100;"IFBW?"`
	`ENTER @E5100;A`

IFBWAUTO

Automatically selects the proper IF bandwidth for each measurement point. (No Query)

Equivalent Key Sequence	(Sweep) IF BW [ ] IF BW AUTO
Examples	`OUTPUT @E5100;"IFBWAUTO"`

INID

Initializes the disk in the built-in flexible disk drive. (No Query)

Equivalent Key Sequence	(Save/Recall) FILE UTILITY INITIALIZE DISK
Examples	`OUTPUT @E5100;"STODDISK"` `OUTPUT @E5100;"INID"`

INP8IO

Inputs data from the 4-bit parallel input port to the E5100A/B. (option 005 only)

Query Response	{*value*}

Examples	`OUTPUT @E5100;"INP8IO"`
	`OUTPUT @E5100;"OUTPINP8IO?"`
	`ENTER @E5100;A`
	
	`OUTPUT @E5100;"INP8IO?"`
	`ENTER @E5100;A`

INPT?

Outputs value which tells whether there is pulse input at the Input1 port of the 24-bit I/O port. (Query only)

Query Response	{1|0} 0 : There is not pulse input at the Input1. 1 : there is pulse input at the Input1. Once `INPT?` returns 1, next `INPT?` query returns 0 until the next pulse input has occurred at Input1.
Examples	`OUTPUT @E5100;"INPT?"` `ENTER @E5100;A`

INPUCALC{01-03}␣<*value1*>,<*value2*>, ... ,<*valuen*>

Stores the measurement calibration error coefficient set real/imaginary pairs input via GPIB into instrument memory. The data transfer format must be used ASCII format when this command. (No Query)

Parameter Range	<*value*> : Complex number (Data format: real, imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`!Set calibration error coefficient`
	`OUTPUT @E5100;"INPUCALC01 ";A(*)`

INPUDATA␣<*value1*>,<*value2*>, ... ,<*valuen*>

Enters data to the data array of the active channel. (No Query)

Parameter Range	<*value*> : Complex value (data format:Real, Imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`! Set data to enter memory array`
	`OUTPUT @E5100;"INPUDATA";A(*)`

INPUDATAM{01-03}␣<*value1*>,<*value2*>, ... ,<*valuen*>

Input 1 point correction value to the analyzer; E5100A/B gets data separately as real part and imaginary part. Refer to DATAMN command. (No Query)

Parameter Range	<*value*> : Complex value (Data format: real, imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`! Set data to enter memory array`
	`OUTPUT @E5100;"INPUDATAM";A(*)`

INPUFORM␣<*value1*>,<*value2*>, ... ,<*valuen*>

Inputs formatted data. (No Query)

Parameter Range	<*value*> : Complex value (Data format: real, imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`! Set formatted data`
	`OUTPUT @E5100;"INPUFORM ";A(*)`

INPUIFORM␣<*value1*>,<*value2*>, ... ,<*valuen*>

Enters data to the formatted data of the inactive channel. (No Query)

When the number of cannel is 3 or 4, this command enters data of the following channel:

Active Channel	Channel the INPUIFORM? command enters data
1	2
2	1
3	4
4	3

Parameter Range	<*value*> : Complex value (data format:Real, Imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`! Set data to enter memory array`
	`OUTPUT @E5100;"INPUIFORM";A(*)`

INPUMEMO␣<*value1*>,<*value2*>, ... ,<*valuen*>

Enters data to the memory array of the active channel. (No Query)

Parameter Range	<*value*> : Complex value (data format:Real, Imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`! Set data to enter memory array`
	`OUTPUT @E5100;"INPUMEMO";A(*)`

INPURAW␣<*value1*>,<*value2*>, ... ,<*valuen*>

Enters data to the raw data array of the active channel. (No Query)

Parameter Range	<*value*> : Complex value (data format:real, imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`! Set data to enter to raw array`
	`OUTPUT @E5100;"INPURAW";A(*)`

INPURFORM␣<value1>,<value2>, ... ,<valuen>

Enters data to the format data array of the active channel. (No Query)

Parameter Description	<*value*> : real value

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`! Set data to enter to format array`
	`OUTPUT @E5100;"INPURFORM ";A(*)`

INPURTMEM␣<value1>,<value2>, ... ,<valuen>

Enters data to the sub-trace array of the active channle. The sub-trace is available when the number of channels is set to 2. (No Query)

Parameter Description	<*value*> : real value

Examples	`DIM A(1:201)` *NOP: 201*
	`! Set data to enter to sub-trace`
	`OUTPUT @E5100;"INPURTMEM";A(*)`

INPUSTIM␣<value1>,<value2>, ... ,<valuen>

Enters data to the stimulus data array of the active channel. (No Query)

Parameter Range	10×10^3 (=10k) to 300×10^6 (=300M) Hz (frequency sweep) −9 to +11 dBm (−48 to +22 dBm option 010 only) (power sweep)

Examples	`DIM A(1:201)` *NOP: 201*
	`! Set stimulus data`
	`OUTPUT @E5100;"INPUSTIM";A(*)`

INPUTRAC␣<value1>,<value2>,<value3>

Pass the phase value to trap at a measured point and specify whether data is outputted to the I/O port or not when the measured value reaches the phase value. The data to be outputted to the I/O port is specified by `INPUTRACB`. (No Query,Option 022 only)

Parameter Description	<*value1*> : The measured point number <*value2*> : The trap phase value <*value3*> : I/O output ON/OFF
Examples	`OUTPUT @E5100;"INPUTRAC 10, 0, ON"`

INPUTRACB␣<*value1*>,<*value2*>,<*value3*>

Pass the data outputted to the I/O port when the measured value of phase reaches the limit value specified by the `INPUTRAC` command. Multiple data up to 6 data can be outputted continuously to the I/O port. (No Query,Option 022 only)

Parameter Description	<*value1*> : The measured point number <*value2*> : The order of output I/O data (1 to 6) <*value3*> : Data Query Response

Examples	`OUTPUT @E5100;"INPUTRACB 5, 1, ";DVAL("1010",2)`
	`OUTPUT @E5100;"INPUTRACB 5, 2, ";DVAL("0100",2)`
	`OUTPUT @E5100;"INPUTRACB 5, 3, ";DVAL("0000",2)`

If E5100A/B OPT. 022 is set up by the above commands, the data described in the following figure will be outputted to the I/O port when the measured phase reaches the target value at the measured point 5.

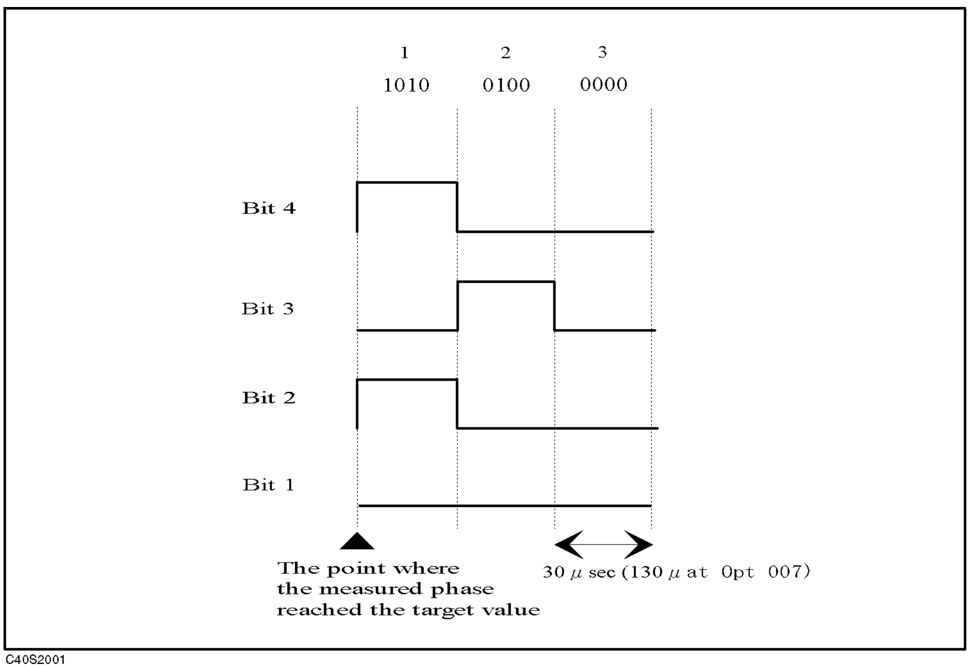


Figure 8-1. The Data Output Timing Chart

IOPO?

Returns the installed option number of the I/O port of the rear panel. If not option is installed, `IOPO?` returns string, “STD”. If Option 005 is installed, `IOPO?` returns “005”, and Option 006 is installed, `IOPO?` returns “006”. (Query only)

Query Response	{STD|005|006} STD : Standard I/O port 005 : Option 005 I/O port 006 : Option 006 I/O port
Examples	`OUTPUT @E5100;"IOPO?"` `ENTER @E5100;A$`

LIMILINE␣{OFF|ON|0|1}

Sets limit line display ON or OFF.

Equivalent Key Sequence	(SYSTEM) LIMIT LINE ON off
Parameter Description	OFF or 0 : Limit lines OFF ON or 1 : Limit lines ON
Query Response	{1|0}

Examples	`OUTPUT 717;"LIMILINE ON"`
	
	`OUTPUT 717;"LIMILINE?"`
	`ENTER 717;A`

LIMITEST␣{OFF|ON|0|1}

Sets the limit testing ON or OFF.

Equivalent Key Sequence	(SYSTEM) LIMIT TEST ON off
Parameter Description	OFF or 0 : Limit testing OFF ON or 1 : Limit testing ON
Query Response	{1|0}

Examples	`OUTPUT @E5100;"LIMITEST ON"`
	
	`OUTPUT @E5100;"LIMITEST?"`
	`ENTER @E5100;A`

LISDFBASE

Display trace on frequency base when the frequency list sweep is used. (No Query)

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU MORE LIST DISP: FREQ BASE
Examples	`OUTPUT @E5100;"LISDFBASE"`

LISDOBASE

Displays the measured data on order base.

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU MORE ORDER BASE
Examples	`OUTPUT @E5100;"LISDOBASE"`

LISFREQ

This command equals `SWPT LIST`. See `SWPT`.

LISSLIS1

Activates LIST 1 for the list sweep. (No Query)

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU MORE LIST [ ]
Examples	`OUTPUT @E5100;"LISSLIS1"`

LISSLIS2

Activates LIST 2 for the list sweep. (No Query)

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU MORE LIST [ ]
Examples	`OUTPUT @E5100;"LISSLIS2"`

LMAX?␣<*value*>

Outputs the nth local maximum value from the left of range which is set by the `ANARANG` command. (Query only)

Parameter Range	1 to
Query Response	{*value*} 3.40282347E+38 will be output when no appropriate points are found.

Examples	`OUTPUT @E5100;"LMAX 1"`
	
	`OUTPUT @E5100;"LMAX?"`
	`ENTER @E5100;A`

LMAXS?␣<*value*>

Outputs the *n*th,from the left of range, local maximum value and stimulus value. The range is set with the ANARANG command. Refer to ANAOCH command, also. (Query only)

Parameter Range	1 to
Query Response	{*value1*} {*value2*} {*value1*} *n*th local maximum value {*value2*} Stimulus value from the left of range 3.40282347E+38 will be output when no appropriate points are found.

Examples	`OUTPUT @E5100;"LMAXS 1"`
	
	`OUTPUT @E5100;"LMAXS?"`
	`ENTER @E5100;A`

LMIN?␣<*value*>

Outputs the *n*th local minimum value from the left of range which is set by the ANARANG command. (Query only)

Parameter Range	1 to
Query Response	{*value*} 3.40282347E+38 will be output when no appropriate points are found.

Examples	`OUTPUT @E5100;"LMIN 1"`
	
	`OUTPUT @E5100;"LMIN?"`
	`ENTER @E5100;A`

LMINS?␣<*value*>

Outputs the *n*th, from the left range, local minimum value and stimulus value. This is set with the ANARANG command. Refer to ANAOCHn command, also. (Query only)

Parameter Range	1 to
Query Response	{*value1*} {*value2*} {*value1*} *n*th local minimum value {*value2*} Stimulus value from the left of range 3.40282347E+38 will be output when no appropriate points are found.

Examples	`OUTPUT @E5100;"LMINS 1"`
	
	`OUTPUT @E5100;"LMINS?"`
	`ENTER @E5100;A`

LOWELIMI␣<*value1*><*value2*><*value3*>, ... ,<*valuen*>

Sets the lower limit values of the limit line. The lower value can be set at each measurement point.

Parameter Description	n = Number of points <value> : Lower value of limit line
Query Response	{*value1*} {*value2*} {*value3*} ... {*valuen*}

Examples	`DIM A(1:201)` *NOP: 201*
	Set lower value of limit line
	`OUTPUT @E5100;"LOWELIMI";A(*)`
	
	`OUTPUT @E5100;"LOWELIMI?"`
	`ENTER @E5100;A(*)`

MARD␣{OFF|ON|0|1}

Displays (ON) or does not display (OFF) markers and the marker information on the screen.

Parameter Description	`OFF` or 0 : Marker display OFF `ON` or 1 : Marker display ON
Query Response	{1|0}

Examples	`OUTPUT @E5100;"MARD OFF"`
	
	`OUTPUT @E5100;"MARD?"`
	`ENTER @E5100;A`

MARK{1-4}␣<*value*>

Selects the active marker and sets the marker stimulus value.

Equivalent Key Sequence	(Marker) `ACTIVE MARKER` `1` to `4`
Parameter Range	<*value*> : 10×10^3 (=10k) to 300×10^6 (=300M) Hz <*value*> : −9 to +11 dBm (−48 to +22 dBm option 010 only) (Option 010, frequency sweep only)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"MARK1 20MHZ"`
	
	`OUTPUT @E5100;"MARK1?"`
	`ENTER @E5100;A`

MARKBUCK␣<*value*>

Moves the active marker to specified data point number. (No Query)

Parameter Range	1 to "number of points"
Examples	`OUTPUT @E5100;"MARKBUCK 20"`

MARKCENT

Changes the stimulus center value to the active marker value. (No Query)

Equivalent Key Sequence	(Center), (Span), (Start), (Stop), MARKER→CENTER
Examples	`OUTPUT @E5100;"MARKCENT"`

MARKCONT

Interpolates between measured points to allow the markers to be placed at any point on the trace.

Equivalent Key Sequence	(Marker) MKR MODE MENU CONTINUOUS
Examples	`OUTPUT @E5100;"MARKCONT"`

MARKCOUP

Couples the marker stimulus values for the two display channels.

Equivalent Key Sequence	(Marker) MKR MODE MENU MARKERS: COUPLED
Examples	`OUTPUT @E5100;"MARKCOUP"`

MARKDISC

Places markers only on measured trace points determined by the stimulus settings.

Equivalent Key Sequence	(Marker) MKR MODE MENU MARKERS: DISCRETE
Examples	`OUTPUT @E5100;"MARKDISC"`

MARKFSTI␣<*value*>

Sets the fixed marker stimulus value offset.

Equivalent Key Sequence	(Marker) ΔMODE MENU FIXED MKR POSITION FIXED MKR STIMULUS
Parameter Range	<*value*> : 10 k to 300 MHz <*value*> : −64 to 18 dBm
Query Response	{*value*}
Examples	OUTPUT @E5100;"MARKFSTI" OUTPUT @E5100;"MARKFSTI?" ENTER @E5100;A

MARKFVAL␣<*value*>

Sets the fixed marker position value offset.

Equivalent Key Sequence	(Marker) ΔMODE MENU FIXED MKR POSITION FIXED MKR VALUE
Query Response	{*value*}
Examples	OUTPUT @E5100;"MARKFVAL" OUTPUT @E5100;"MARKFVAL?" ENTER @E5100;A

MARKL␣{OFF|ON|0|1}

Displays (ON) or does not display (OFF) the list of stimulus values and response values of all markers.

Equivalent Key Sequence	(MKR) MKR LIST on OFF
Parameter Description	OFF or 0 : Marker list OFF ON or 1 : Marker list ON
Query Response	{0|1}
Examples	OUTPUT @E5100;"MARKL OFF" OUTPUT @E5100;"MARKL?" ENTER @E5100;A

MARKODATA

Enables the marker to move on the measurement data trace.

Examples	`OUTPUT @E5100;"MARKODATA"`

MARKOFF

Turns off all the markers and the delta reference marker.

Equivalent Key Sequence	(Marker) ALL MKR OFF
Examples	`OUTPUT @E5100;"MARKOFF"`

MARKOMEMO

Enables the marker to move on the memory data trace.

Examples	`OUTPUT @E5100;"MARKOMEMO"`

MARKREF

Changes the reference value to the active marker's response value, without changing the reference position. (No Query)

Examples	`OUTPUT @E5100;"MARKREF"`

MARKSPAN

Changes the start and stop values of the stimulus span to the active marker and the delta reference marker.

Equivalent Key Sequence	(Center), (Span), (Start), (Stop), MARKER→SPAN
Examples	`OUTPUT 717;"MARKSPAN"`

MARKSTAR

Changes the start value of the stimulus to the active marker value. (No Query)

Equivalent Key Sequence	(Center), (Span), (Start), (Stop), MARKER → START
Examples	`OUTPUT 717;"MARKSTAR"`

MARKSTOP

Changes the stop value of the stimulus to the active marker value. (No Query)

Equivalent Key Sequence	(Center), (Span), (Start), (Stop), MARKER → STOP
Examples	`OUTPUT 717;"MARKSTOP"`

MARKTIME␣{OFF|ON|0|1}

Sets the x-axis marker readout to the sweep time (ON), or cancels the setting (OFF).

Equivalent Key Sequence	(Mkr) MKR TIME on OFF
Parameter Description	OFF or 0: Marker time OFF ON or 1 : Marker time ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"MARKTIME ON"`
	
	`OUTPUT @E5100;"MARKTIME?"`
	`ENTER @E5100;A`

MARKUNCO

Allows the marker stimulus values to be controlled independently on each channel.

Equivalent Key Sequence	(Marker) MKR MODE MENU UNCOUPLE
Examples	`OUTPUT @E5100;"MARKUNCO"`

MARKZERO

Puts a fixed reference marker at the present active marker position, and makes the fixed marker stimulus and response values at that position equal to zero.

Equivalent Key Sequence	(Marker) ΔMODE MENU MKR ZERO ΔREF=Δ
Examples	`OUTPUT @E5100;"MARKZERO"`

MAXPOIN?

Outputs the maximum number of the measurement points. (Query only)

Query Response	{401|801|1601}
Examples	`OUTPUT @E5100;"MAXPOIN?"` `ENTER @E5100;A`

MAXPORT?

Outputs the number of ports (receivers). (Query only)

Query Response	{1|2|3|4}
Examples	`OUTPUT @E5100;"MAXPORT?"` `ENTER @E5100;A`

MEAS1PT?␣<*value*>

Outputs measurement value at the point number specified by the parameter.

When the trap function is turn on (`TRAP ON`), `MEAS1PT` waits to start a measurement until the phase value is in the condition specified by `INPUTRAC` command, and returns the query response.

Parameter Range	1 to NOP
Query Response	{*value*}

Examples	`OUTPUT @E5100;"MEAS1PT? 10000000"`
	
	`OUTPUT @E5100;"MEAS1PT?"`
	`ENTER @E5100;A`

MEAS␣{AR|BR|CR|RA|BA|CA|RB|AB|CB|RC|AC|BC|R|A|B|C}

Selects the parameters or inputs to be measured.

Equivalent Key Sequence	(MEAS) A/R , R , A
Parameter Description	AR : A/R measurement BR : A/R measurement CR : A/R measurement RA : R/A measurement BA : B/A measurement CA : C/A measurement RB : R/B measurement AB : A/B measurement CB : C/B measurement RC : R/C measurement AC : A/C measurement BC : B/C measurement R : R measurement A : A measurement B : B measurement C : C measurement
Query Response	{AR|BR|CR|RA|BA|CA|RB|AB|CB|RC|AC|BC|R|A|B|C}

Examples	`OUTPUT @E5100;"MEAS AR"`
	
	`OUTPUT @E5100;"MEAS?"`
	`ENTER @E5100;A$`

MEASA

This command equals `MEAS A`. See `MEAS`.

MEASR

This command equals `MEAS R`. See `MEAS`.

MEASTAT␣{OFF|ON|0|1}

Calculates and displays the mean, standard deviation, and peak-to-peak values among the search range (ON), or does not display them (OFF).

Equivalent Key Sequence	(Marker) UTILITY MENU STATISTICS ON OFF
Parameter Description	OFF or 0 : Marker statistic OFF ON or 1 : Marker statistic ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"MEASTAT ON"`
	
	`OUTPUT @E5100;"MEASTAT?"`
	`ENTER @E5100;A`

MENU3␣<*value*>,<*string1*>,<*string2*>,<*string3*>

Defines the user softkey mene of (Menu3). (No Query)

Parameter Description	<*value*> : Softkey position (1 to 8) <*string1*> : Upper softkey lable (Up to ten characters) <*string2*> : Lower softkey lable (Up to ten characters) <*string3*> : GPIB command to be executed
Examples	`OUTPUT @E5100;"MENU3 1,""START"",""100 MHz"",""STAR 100MA"""`

MOHMSW␣{A|B},{ON|OFF}

Sets 1 MΩ input ON or OFF. (option 101 only)

Equivalent Key Sequence	(System) INPUT Z PORT A [] , PROT B []
Parameter Description	A : port A B : port B ON : 1 MΩ input OFF : 50 Ω input
Query Response	{ON|OFF}

Examples	`OUTPUT @E5100;"MOHMSW A,ON"`
	
	`OUTPUT @E5100;"MOHMSW? A"`
	`ENTER @E5100;A$`

MONDYEAR

Changes the displayed date to the "month:day:year" format. (No Query)

Equivalent Key Sequence	(System) MORE SET CLOCK DATE MODE: MonDayYear
Examples	`OUTPUT @E5100;"MONDYEAR"`

MOVADARY

Enters measurement data triggered by `TRIGMEAS` command. This command is only used for the parallel processing using with `TRIGMEAS` and `MOVADARY`. (No Query)

MULC␣{OFF|ON|0|1}

Selects the multi (ON) or single (OFF) channels display.

Equivalent Key Sequence	(Display) MULTI CH ON OFF
Parameter Description	OFF or 0 : Active channel only ON or 1 : Multi-channel
Query Response	{0|1}

Examples	`OUTPUT @E5100;"MULC ON"`
	
	`OUTPUT @E5100;"MULC?"`
	`ENTER @E5100;A`

NEGL

Sets the output of the 24-bit I/O port to the negative logic. (No Query)

Examples	`OUTPUT @E5100;"NEGL"`

NEXPK?

Outputs the maximum local maximum value and its stimulus next to the value last found by the PEAK?, or NEXPK? commands. For more information, refer to Appendix D. (Query only)

Query Response	{*value1*}{*value2*} <*value1*> : Maximum local maxmum value <*value2*> : Stimulus value
Examples	`OUTPUT @E5100;"NEXPK?"` `ENTER @E5100;A,B`

NEXNPK?

Output the negative peak value and stimulus which found nextly from the last one which found by 'NPEAK?'' or NEXNPK? (Query only).

Query Response	{*value1*}{*value2*} <*value1*> : Negative peak <*value2*> : Stimulus value
Examples	`OUTPUT @E5100;"NEXNPK?"` `ENTER @E5100;A,B`

NOMF␣<*Value*>

Use this command to specify the nominal load resonance frequency in [Hz]. This frequency must be the frequency of a resonator used when you measure the load resonance frequency through search using the `SRCHTRFL?` command.

Note that this frequency is internally linked to the frequency at which the search starts to follow the series resonance frequency (Fr) that is specified using the `PTFR` command (option 023). Therefore, if you specify one of these frequencies, the other frequency will be automatically set at the same value.

See the sample programs in Appendix K for more information.

Parameter Range	10000 to 3.E+8 (= 10 kHz to 300 MHz) Unit: Hz Default: 10000
Query Response	{*Value*} {*Value*} : Nominal load resonance frequency in [Hz] to be specified (frequency used when you make search using the `SRCHTRFL?` command)
Example	`OUTPUT @E5100;"NOMF 1.E+7"`

NPEAK?

Output minimus negative peak value in the range which specified by `ANARANG` (query only).

Query Response	{*value1*}{*value2*} <*value1*> : Minimus negative peak <*value2*> : Stimulus value
Examples	`OUTPUT @E5100;"NPEAK?"` `ENTER @E5100;A,B`

NPEAKLIST?

Output Nth smallest negative peaks value (Query only).

Parameter Range	1 to NOP
Query Response	{*value1*}, {*value2*}, ...

Examples	`DIM A(3)`
	`OUTPUT @E5100;"NPEAKLIST? 3"`
	`ENTER @E5100;A(*)`

NPEAKSORT?␣<*value*>

Searches n peaks starting from the smallest peak value within the range specified by the ANARANG commands. Outputs the n stimulus (frequency or level) values in increasing order.

Parameter Description	The number of searched negative peak
Parameter Range	1 to 100
Query Response	{*value1*},{*value2*}, . . .

Examples	`DIM A(1:5,1:2)`
	`OUTPUT @E5100;"NPEAKSORT? 5"`
	`ENTER @E5100;A(*)`

NUMC␣<*value*>

Sets the number of channel.

Equivalent Key Sequence	(Meas/Format) `NUM of CH [ ]`
Parameter Range	1 to 4
Query Response	{*value*}

Examples	`OUTPUT @E5100;"NUMC 3"`
	
	`OUTPUT @E5100;"NUMC?"`
	`ENTER @E5100;A`

NUMG␣<*value*>

Triggers a user-specified number of sweeps, and returns to the HOLD mode. (No Query)

Parameter Range	Greater than 0, integer only
Examples	`OUTPUT @E5100;"NUMG 10"`

NUMLMAX?

Outputs the number of local maximums within the range set by the `ANARANG` command. (Query only)

Query Response	{*value*} (number of local maximums)
Examples	`OUTPUT @E5100;"NUMLMAX?"` `ENTER @E5100;A`

NUMLMIN?

Outputs the number of local minimum within the range set by the `ANARANG` command. (Query only)

Query Response	{*value*} (number of local minimums)
Examples	`OUTPUT @E5100;"NUMLMIN?"` `ENTER @E5100;A`

NUMLMINMAX?

Outputs the total number of the local maxmum and local miminum points within the range specified by the `ANARANG` command. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"NUMLMINMAX?"` `ENTER @E5100;A`

OSE␣<*value*>

Enables the operational status register.

Parameter Range	0 to 65535 ($=2^{16}-1$, decimal expression of enable bits of the the operational status register)
Query Response	{*value*}
Examples	`OUTPUT @E5100;"OSE 1"`

OSER?

Outputs the current value in the event register of an operational status register. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"OSER?"` `ENTER @E5100;A`

OSNT␣<*value*>

Sets the negative transition filter of an operational status register.

OSNT␣<*value*>

Parameter Range	0 to 65535 ($=2^{16}-1$, decimal expression of the operational status register)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"OSNT 1"`
	
	`OUTPUT @E5100;"OSNT?"`
	`ENTER @E5100;A`

OSPT␣<*value*>

Sets the positive transition filter of an operational status register.

Parameter Range	0 to 65535 ($=2^{16}-1$, decimal expression of the operational status register)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"OSPT 1"`
	
	`OUTPUT @E5100;"OSPT?"`
	`ENTER @E5100;A`

OSR?

Outputs the operational status register value. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"OSR?"` `ENTER @E5100;A`

OUT1ENVH

Sets OUTPUT1 set to HIGH when a pulse input has occurred at INPUT1. (No Query)

Examples	`OUTPUT @E5100;"OUT1ENVH"`

OUT1ENVL

Sets OUTPUT1 set to LOW when a pulse input has occurred at INPUT1. (No Query)

Examples	`OUTPUT @E5100;"OUT1ENVL"`

OUT1H

Sets OUTPUT1 to HIGH. (No Query)

Examples	`OUTPUT @E5100;"OUT1H"`

OUT1L

Sets OUTPUT1 to LOW. (No Query)

Examples	`OUTPUT @E5100;"OUT1L"`

OUT2ENVH

Sets OUTPUT2 set to HIGH when a pulse input has occurred at INPUT1 of the 24 bit I/O port. (No Query)

Examples	`OUTPUT @E5100;"OUT2ENVH"`

OUT2ENVL

Sets OUTPUT2 set to LOW when a pulse input has occurred at INPUT1 of the 24 bit I/O port. (No Query)

Examples	`OUTPUT @E5100;"OUT2ENVL"`

OUT2H

Sets OUTPUT2 to HIGH. (No Query)

Examples	`OUTPUT @E5100;"OUT2ENVL"`

OUT2L

Sets OUTPUT2 to LOW. (No Query)

Examples	`OUTPUT @E5100;"OUT2L"`

OUT8IO␣<*value*>

Outputs the data to the 8-bit parallel output port. (Option 005 only, No Query)

Parameter Range	0 to 255
Examples	`OUTPUT @E5100;"OUT8IO 0"`

OUTAIO␣<*value*>

Output decimal data specified as the parameter to port A (8 bit) of the 24-bit I/O port. (No Query)

Parameter Range	0 to 255
Examples	`OUTPUT @E5100;"OUTAIO 0"`

OUTBIO␣<*value*>

Output decimal data specified as the parameter to port B (8 bit) of the 24-bit I/O port. (No Query)

Parameter Range	0 to 255
Examples	`OUTPUT @E5100;"OUTBIO 0"`

OUTCIO␣<*value*>

Output decimal data specified as the parameter to port C (4 bit) of the 24-bit I/O port. (No Query)

Parameter Range	0 to 15
Examples	`OUTPUT @E5100;"OUTCIO 0"`

OUTDIO␣<*value*>

Output decimal data specified as the parameter to port D (4 bit) of the 24-bit I/O port. (No Query)

Parameter Range	0 to 15
Examples	`OUTPUT @E5100;"OUTDIO 0"`

OUTEIO␣<*value*>

Output decimal data specified as the parameter to port E (8 bit) of the 24-bit I/O port. (No Query)

Parameter Range	0 to 255
Examples	`OUTPUT @E5100;"OUTEIO 0"`

OUTFIO␣<*value*>

Output decimal data specified as the parameter to port F (16 bit) of the 24-bit I/O port. (No Query)

Parameter Range	0 to 65535
Examples	`OUTPUT @E5100;"OUTFIO 0"`

OUTGIO␣<*value*>

Output decimal data specified as the parameter to port G (20 bit) of the 24-bit I/O port. (No Query)

Parameter Range	0 to 1048575
Examples	`OUTPUT @E5100;"OUTGIO 0"`

OUTHIO␣<*value*>

Output decimal data specified as the parameter to port H (24 bit) of the 24-bit I/O port. (No Query)

Parameter Range	0 to 16777215
Examples	`OUTPUT @E5100;"OUTHIO 0"`

OUTPCALC{01-03}?

Outputs the active calibration set array of the active channel. (Query only)

Query Response	{*value (1)*} {*value (2)*} . . . {*value (n)*} n: Number of points *value*: Complex value (data format : real, imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`OUTPUT @E5100;"OUTCALC01?"`
	`ENTER @E5100;A(*)`

OUTPCERR?

Outputs the ceramic resonator parameters within the range specified by the `ANARANGE` command. (Query only)

Query Response	{*value1*}, {*value2*}, {*value3*}, {*value4*}, {*value5*}, {*value6*}, {*value7*} {*value1*} : Gain at f_r (G_r) {*value2*} : Resonant frequency (f_r) {*value3*} : Gain at (f_a) {*value4*} : Anti-resonant frequency f_a (G_a) {*value5*} : Maximum ripple height in left side of resonant point {*value6*} : Maximum ripple height between resonant and anti-resonant point {*value7*} : Maximum ripple height in right side of anti-resonant point If `OUTPCERR?` could not find any ripples, the query returns `0`. This command is only available when the "LOG MAG & Phase", "LOG MAG & Delay" or "LOG MAG" formats are selected. If another format is selected, the query returns `0`. If Z-conversion is selected, then the impedance at f_r (Z_r) instead of the G_r and the impedance at f_a (Z_a) instead of the G_a are returned.

Examples	`OUTPUT @E5100;"OUTPCERR?"`
	Query the cerramic resonator parameters.
	`ENTER @E5100;Gr,Fr,Ga,Fa,Rpl1,Rpl2,Rpl3`
	Recieve the all return value.

OUTPCF2?␣<*value1*>,<*value2*>,<*value3*>,<*value4*>,<*value5*> ... ,<*valuen + 4*>

Outputs filter parameters within the range specified by the `ANARANG` command. This command outputs the same parameters as `OUTPCFIL` and outputs up to 20 sets of frequency offsets from center frequency (f_c) to left and right cutoff points ($\Delta f_{left\ n}$, $\Delta f_{right\ n}$).

Where,

- If two cutoff points which are x_1 dB below the maximum peak are not found, zeros will be returned for all parameters.
- If two cutoff points which are X_n dB below the maximum value are not found, zeroes will be returned for $\Delta f_{left\ n}$ and $\Delta f_{right\ n}$.
- If no peaks are found, zeroes will be returned for $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, $Pole_{stim2}$.*

Parameter Description

<*value1*> : Nominal frequency (f_c)
<*value2*> : Difference vale from the maximum value (D)
<*value3*> : Stop frequency of rejection band (f_1)
<*value4*> : Start frequency of sprious level range (f_2)
<*value5*> : Relative offset value from maximum peak value to determine the cutoff point (x_1[dB])
<*value6*> : Relative offset value from maximum peak value to determine the cutoff point (x_2 [dB])
⋮
<*valuen+4*> : Relative offset value from maximum peak value to determine the cutoff point (x_n[dB])
($2 \leq n \leq 20$)

Query Response

{*value1*}{*value2*} ... {*value(2n+11)*}

{value1} : insertion loss
{value2} : constant loss ($Loss_c$)
{value3} :x_1dB bandwidth
{value4} :center freequency (f_{cent})
{value5} :Q
{value6} :Ripple value at the passband
{value7} :Rejection
{value8} :Spurious level
{value9} :Left pole (negative peak in the left side of the maximum value) ($Pole_{x1}$)
{value10} :Stimulus value of $Pole_{x1}$ ($Pole_{stim1}$)
{value11} :Right pole (negative peak in the right side of the maximum value) ($Pole_{x2}$)
{value12} :Stimulus value of $Pole_{x2}$ ($Pole_{stim2}$)
{value13} :difference between f_c and the left point of cutoff frequency (Δf_{left1})
{value14} :difference between f_c and the right point of cutoff frequency (Δf_{right1})
{value15} :Δf_{left2}
{value16} :Δf_{right2}
⋮
{value2n+10} :$\Delta f_{left\ n}$
{value2n+11} :$\Delta f_{right\ n}$

Examples

```
OUTPUT @E5100;"OUTPCF2? 90M,0.1,90.2,3,10,12,20"
ENTER @E5100;A(10),B(*)
```

OUTPCFIL?␣<*value1*>,<*value2*>, ... , <*value6*>

Outputs filter parameters within the range specified by the `ANARANG` command. Command parameter sets nominal frequency, the offset of *x1*dB and *x2*dB to the maximum peak value to determine the cutoff points, same parameter with `POLE?`, and f_1 and f_2 for determining rejection level and spurious level respectively. For details, refer to Appendix D. (Query only)

OUTPCFIL?␣<*value1*>,<*value2*>, . . . , <*value6*>

Parameter Description	<*value1*> : Nominal frequency (f_C) <*value2*> : Relative offset value from maximum peak value to determine the cutoff point (x_1) [dB] <*value3*> : Relative offset value from maximum peak value to determine the cutoff point (x_2) [dB] <*value4*> : `POLE?` parameter <*value5*> : Stop frequency of rejection band (f_1) <*value6*> : Start frequency of sprious level range (f_2)
Query Response	{*value1*}, {*value2*}, . . . , {*value16*} {*value1*} : insertion loss {*value2*} : constant loss {*value3*} : xdB bandwidth {*value4*} : center freequency {*value5*} : Q {*value6*} : difference between the middle point of the analysis range and the left point of cutoff frequency {*value7*} : difference between the middle point of the analysis range and the right point of cutoff frequency {*value8*} : difference between the middle point of the analysis range and the left point of cutoff frequency {*value9*} : difference between the middle point of the analysis range and the right point of cutoff frequency {*value10*} : Ripple value at the passband {*value11*} : Rejection {*value12*} : Spurious level {*value13*} : Left pole (negative peak in the left side of the maximum value) {*value14*} : Stimulus value of the left pole {*value15*} : Right pole (negative peak in the right side of the maximum value) {*value16*} : Stimulus value of the right pole

Examples	`DIM ANS(1:16)`
	`OUTPUT @E5100;"OUTPCFIL?";7.0E6,-10,-20,-40,1E3,1E3`
	`ENTER @E5100;ANS(*)`

OUTPDATA?

Outputs data trace value. (Query only)

Query Response	{*numeric (1)*} {*numeric (2)*} . . . {*numeric (n)*} n : Number of points *numeric* : Complex value (data format: real, imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`OUTPUT @E5100;"OUTPDATA?"`
	`ENTER @E5100;A(*)`

OUTPDATAM{01-03}?

Output the 1 point correction factor. Refer to `DATAMN` command, also. (Query only)

Query Response	{*numeric*} *numeric* : Complex value (data format: real, imaginary)
Examples	`OUTPUT @E5100;"OUTPDATAM?"` `ENTER @E5100;Real,Imaginary`

OUTPDATAP␣<*value*>

Outputs the data at the specified point from the data array.

Parameter Range	<value>:number of points :(1 to NOP)
Query Response	{*value*} *value*: Complex data (data format: real, imaginary)
Examples	`OUTPUT @E5100;"OUTPDATAP? 1"` `ENTER @E5100;A`

OUTPDATAT?

Outputs data trace value on 16 points stimulus which is set by the `STIDROUT` command. If there are points that is not set by the `STIDROUT` command, the `OUTPDATAT?` returns the value at 100 kHz. (Query only)

Query Response	{*value1*} {*value2*} . . . {*value16*}

Examples	`DIM A(1:16)`
	`OUTPUT @E5100;"STIDROUT1";Freq`
	`OUTPUT @E5100;"OUTPDATAT?"`
	`ENTER @E5100;A(*)`

OUTPDATTP?␣<*value*>

Outputs the data-trace data at the specified point. (Query only)

Parameter Range	*value* : point number n:(1 to Number of Points)
Query Response	{*numeric*}
Examples	`OUTPUT @E5100;"OUTPDATTP ";1` `ENTER @E5100;A`

OUTPERRO?

Outputs the error message in the error queue. (Query only)

Query Response	{*value*}, {*string*} <*value*> : Error number <*string*> : Error string
Examples	`OUTPUT @E5100;"OUTPERRO?"` `ENTER @E5100;A,ERR$`

OUTPFILT?␣<*value*>

Outputs filter parameters within the range specified by the `ANARANG` command. Command parameter sets the offset of xdB to the maximum peak value to determine the cutoff points. For details, refer to Appendix D. (Query only)

Parameter Description	<*value*> : Relative offset value from maximum x[dB]
Query Response	{*value1*}, {*value2*}, . . . , {*value6*} {*value1*} : Insertion loss {*value2*} : xdBbandwidth {*value3*} : Center frequency {*value4*} : Q {*value5*} : Difference between the middle value of analysis range and the left cutoff frequency (ΔL.F.) {*value6*} : Difference between the middle value of analysis range and the right cutoff frequency (ΔR.F.)
Examples	`OUTPUT @E5100;"OUTPFILT? -3"` `ENTER @E5100;ANS(*)`

OUTPFORM?

Outputs the formatted trace data. (Query only)

Query Response	{*value (1)*} {*value (2)*} . . . {*value (n)*} n: Number of point value: complex value (data format : Real, Imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`OUTPUT @E5100;"OUTPFORM?"`
	`ENTER @E5100;A(*)`

OUTPFORMP?␣<*value*>

Outputs the formatted trace data at the specified point. (Query only)

Parameter Range	1 to "number of points"
Query Response	{*value*} value : complex value (data format : real, imaginary)
Examples	`OUTPUT @E5100;"OUTPFORMP?"` `ENTER @E5100;A,B`

OUTPIFORM?

Outputs the formatted data from the inactive channel. (Query only)

When the number of cannel is 3 or 4, this command outputs data from the following channel:

Active Channel	Channel the OUTPIFORM? command outputs
1	2
2	1
3	4
4	3

Query Response	{*value (1)*} {*value (2)*} . . . {*value (n)*} n:number of points *value* : Complex value (data format: real, imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`OUTPUT @E5100;"OUTIFORM?"`
	`ENTER @E5100;A(*)`

OUTPINP8IO?

Outputs the data entered from the 4-bit parallel input port. (option 005 only;Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"OUTPINP8IO?"` `ENTER @E5100;A`

OUTPINPCIO?

Outputs the data entered from port C (4 bit) of the 24-bit I/O port. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"OUTPINPCIO?"` `ENTER @E5100;A`

OUTPINPDIO?

Outputs the data entered from port D (4 bit) of the 24-bit I/O port. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"OUTPINPDIO?"` `ENTER @E5100;A`

OUTPINPEIO?

Outputs the data entered from port E (8 bit) of the 24-bit I/O port. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"OUTPINPEIO?"` `ENTER @E5100;A`

OUTPIRFORM?

Outputs the real part of the formatted data from the inactive channel. (Query only)

Query Response	{*value (1)*} {*value (2)*} . . . {*value (n)*} n:Number of points

Examples	`DIM A(1:201)` *NOP: 201*
	`OUTPUT @E5100;"OUTPIRFORM?"`
	`ENTER @E5100;A(*)`

OUTPIRTMEM?

Outputs the real part of the trace memory data from the inactive channel. (Query only)

When the number of cannel is 3 or 4, this command outputs data from the following channel:

Active Channel	Channel the OUTPIRFORM? command outputs data
1	2
2	1
3	4
4	3

Query Response	{*value (1)*} {*value (2)*} . . . {*value (n)*} n : Number of points

Examples	`DIM A(1:201)` *NOP: 201*
	`OUTPUT @E5100;"OUTPIRTMEM?"`
	`ENTER @E5100;A(*)`

OUTPMARK?

Outputs the active marker values. (Query only)

Query Response	{*numeric1*}, {*numeric2*}, {*numeric3*} {*numeric1*} : marker value {*numeric2*} : marker aux. value {*numeric3*} : stimulus
Examples	`OUTPUT @E5100;"OUTPMARK?"` `ENTER @E5100;A,B,C`

OUTPMAX?

Outputs the maximum value within the range specified with the `ANARANG` command. For details, refer to Appendix D. (Query only,)

Query Response	{*numeric1*}, {*numeric2*} {*numeric1*} : maximum value {*numeric2*} : stimulus value
Examples	`OUTPUT @E5100;"OUTPMAX?"` `ENTER @E5100;A,B`

OUTPMEAN?

Outputs the mean value within the range specified with the `ANARANG` command. For details, refer to Appendix D. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"OUTPMEAN?"` `ENTER @E5100;A`

OUTPMEMO?

Outputs memory trace value. (Query only)

Query Response	{*numeric (1)*} {*numeric (2)*} ... {*numeric (n)*} n : Number of points *numeric* : Complex value (data format: real, imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`OUTPUT @E5100;"OUTPMEMO?"`
	`ENTER @E5100;A(*)`

OUTPMEMOT?

Outputs memory trace value on 16 points stimulus which is set by the `STIMROUT` command. If there are points that is not set by `STIMROUT` command, `OUTPMEMOT?` returns the value at 100 kHz. (Query only)

Query Response	{*numeric1*} {*numeric2*} ... {*numeric16*}

Examples	`DIM A(1:16)`
	`OUTPUT @E5100;"STIMROUT1";Freq`
	`OUTPUT @E5100;"OUTPMEMOT?"`
	`ENTER @E5100;A(*)`

OUTPMEMTP?␣<*value*>

Outputs the memory data at a specified point. (Query only)

Parameter Description	*value* : point number n:(1 to Number of Points)
Query Response	{*numeric1*},{*numeric2*} {*numeric1*} : real part {*numeric2*} : imaginary part
Examples	`OUTPUT @E5100;"OUTPMEMTP? ";1` `ENTER @E5100;A,B`

OUTPMIN?

Outputs the minimum value within the range specified with the `ANARANG` command. For details, refer to Appendix D. (Query only)

Query Response	{*numeric1*}, {*numeric2*} {*numeric1*} : minimum value {*numeric2*} : stimulus value
Examples	`OUTPUT @E5100;"OUTPMIN?"` `ENTER @E5100;A,B`

OUTPMINMAX?

Outputs the maximum and minimum values within the range specified with the `ANARANG` command. For details, refer to Appendix D. (Query only)

Equivalent Key Sequence	(Memu2) `RESONANT (MINMAX)`
Query Response	{*numeric1*}, {*numeric2*}, {*numeric3*}, {*numeric4*} {*numeric1*} : minimum value {*numeric2*} : stimulus value {*numeric3*} : maximum value {*numeric4*} : stimulus value
Examples	`OUTPUT @E5100;"OUTPMINMAX?"` `ENTER @E5100;A,B,C,D`

OUTPMSTA?

Outputs the marker statistics within the specified range by the `ANARANG` command. (Query only)

Equivalent Key Sequence	(Marker) `UTILITY MENU` `STATISTICS ON off`
Query Response	{*numeric1*}, {*numeric2*}, {*numeric3*} {*numeric1*} : mean {*numeric2*} : standard deviation {*numeric3*} : peak to peak
Examples	`OUTPUT @E5100;"OUTPMSTA?"` `ENTER @E5100;A,B,C`

Note

(Marker) `UTILITY MENU` `STATISTICS ON` displays the maker statistics within the entire display range.

OUTPMWID?

Outputs the results of the bandwidth search. (Query only)

Equivalent Key Sequence	(Marker) `UTILITY MENU` `WIDTH ON OFF`
Query Response	{*numeric1*}, {*numeric2*}, {*numeric3*} {*numeric1*} : bandwidth {*numeric2*} : center {*numeric3*} : Q

Examples	`OUTPUT @E5100;"WIDV -5"`
	`OUTPUT @E5100;"WIDT ON"`
	`OUTPUT @E5100;"OUTPMWID?"`
	`ENTER @E5100;A,B,C`

OUTPMWIL?

Outputs the results of the bandwidth search with the bandwidth, center value, Q, and the insertion loss. (Query only)

Equivalent Key Sequence	(Marker) UTILITY MENU WIDTH ON OFF
Query Rewponse	{*value1*}, {*value2*}, {*value3*}, {*value4*} {*value1*} : Bandwidth {*value2*} : Center value {*value3*} : Q {*value4*} : Insertion loss

Examples	`OUTPUT @E5100;"WIDV -5"`
	`OUTPUT @E5100;"MARK1"`
	`OUTPUT @E5100;"SEAMAX"`
	`OUTPUT @E5100;"WIDT ON"`
	`OUTPUT @E5100;"OUTPMWIL?"`
	`ENTER @E5100;A,B,C,D`

OUTPMWLF?

Outputs the results of the bandwidth search with the insertion loss, the difference between the center frequency and the lower cutoff frequency (ΔL.F), and the difference between the center frequency and the upper cutoff frequency (ΔR.F) values. (Query only)

Equivalent Key Sequence	(Marker) UTILITY MENU WIDTH ON OFF
Query Response	{*numeric1*}, {*numeric2*}, . . . , {*numeric6*} {*numeric1*> : bandwidth {*numeric2*> : center {*numeric3*> : Q {*numeric4*> : insertion loss {*numeric5*> : ΔL.F {*numeric6*> : ΔR.F

Examples	`OUTPUT @E5100;"WIDV -5"`
	`OUTPUT @E5100;"WIDT ON"`
	`OUTPUT @E5100;"OUTPMWLF?"`
	`ENTER @E5100;A,B,C,D,E,F`

OUTPRAW?

Outputs the raw data array of the active channel. (Query only)

Query Response	{*value (1)*} {*value (2)*} . . . {*value (n)*} n = Number of points *value*:complex value(data format : Real, Imaginary)

Examples	`DIM A(1:201,1:2)` *NOP: 201*
	`OUTPUT @E5100;"OUTPRAW?"`
	`ENTER @E5100;A(*)`

OUTPRESF?␣<*value1*>,<*value2*>

Returns the stimulus of the maximum local-maximum and its $x1$dB below points of both sides, and the stimulus of minimum local-minimum and its $x2$dB above points of both sides. For more details, refer to Appendix D. (Query only)

Parameter Description	<*numeric1*> : down vale from the maximum peak x_1[dB] <*numeric2*> : down value from the minimum peak x_2[dB]
Query Response	{*numeric1*}, {*numeric2*}, ... , {*numeric6*} {*numeric1*} : center frequency between f_{s1} and f_{s2} {*numeric2*} : center frequency between f_{p1} and f_{p2} {*numeric3*} : left point of x_1dB down points from the maximum peak : f_{s1} {*numeric4*} : right point of x_1dB down points from the maximum peak : f_{s2} {*numeric5*} : left point of x_2dB down points from the maximum peak :f_{p1} {*numeric6*} : right point of x_2dB down points from the maximum peak :f_{p2}

Examples	`DIM ANS(1:6)`
	`OUTPUT @E5100;"OUTPRESF? ";-3,3`
	`ENTER @E5100;ANS(*)`

OUTPRESO?

Outputs the series resonant (Resonant) and parallel resonant (Anti-Resonant) parameters, 0° phase point frequency f_r (Resonant frequency) and f_a (Anti-Resonant frequency), and the corresponding gain values G_r and G_a. For details, refer to Appendix D. (Query only)

Equivalent Key Sequence	(Memu2) RESONANT (0° PHASE)
Query Response	{*numeric1*}, {*numeric2*}, ... , {*numeric4*} {*numeric1*} : resonance impedance (G_r) {*numeric2*} : resonance frequency (f_r) {*numeric3*} : anti-resonance impedance (G_a) {*numeric4*} : anti-resonance frequency (f_a)

Examples	`DIM ANS(1:4)`
	`OUTPUT @E5100;"OUTPRESO?"`
	`ENTER @E5100;ANS(*)`

OUTPRESR?

Outputs same parameter as `OUTPRESO?` and maximum difference, rpl1 of local maximum and its left local minimum on left of resonant point, maximum difference, rpl2 of local maximum and its right local minimum between resonant and anti-resonant points, and the maximum difference, rpl3 of the local maximum and its left local minimum on the right of the anti-resonant point. For details, refer to Appendix D. (Query only)

OUTPRESR?

Query Response	{*numeric1*}, {*numeric2*}, . . . , {*numeric7*} {*numeric1*} : resonance impedance (G_r) {*numeric2*} : resonance frequency (f_r) {*numeric3*} : anti-resonance impedance (G_a) {*numeric4*} : anti-resone frequency (f_a) {*numeric5*} : maximum difference of local maximum and its left local minimum on left of resonant point (rpl1) {*numeric6*} : maximum difference of local maximum and its right local minimum between resonant and anti-resonant points (rpl2) {*numeric7*} : maximum difference of the local maximum and its left local minimum on the right of the anti-resonant point (rpl3)

Examples	`DIM ANS(1:7)`
	`OUTPUT @E5100;"OUTPRESR?"`
	`ENTER @E5100;ANS(*)`

OUTPRFORM?

Outputs the main trace array of the active channel. (Query only)

Query Response	{*value (1)*} {*value (2)*} . . . {*value (n)*} n = Number of points *value* : Real value

Examples	`DIM A(1:201)` *NOP: 201*
	`OUTPUT @E5100;"OUTPRFORM?"`
	`ENTER @E5100;A(*)`

OUTPRFORMP?␣<*value*>

Outputs the specified points of the main trace array of the active channel. (This command equals `OUTPDATTP?`) (Query only)

Parameter Description	<*value*>: Point number to output : (1 to Number of points)
Query Response	{*value*}
Examples	`OUTPUT @E5100;"OUTPRFORMP? 2"` `ENTER @E5100;A`

OUTPRTMEM?

Outputs the sub-trace array of the active channel. (Query only)

Query Response	{*value (1)*} {*value (2)*} . . . {*value (n)*} n = Number of points *value* : Real value

Examples	`DIM A(1:201)` *NOP: 201*
	`OUTPUT @E5100;"OUTPTMEM?"`
	`ENTER @E5100;A(*)`

OUTPRTMEMP?␣<*value*>

Outputs the specified points of the sub-trace array of the active channel. (This command equals `OUTPMEMTP?`) The sub-trace is available when the number of channels is set to 2. (Query only)

Parameter Description	<*value*>: Point number : (1 to Number of points)
Query Response	{*value*}
Examples	`OUTPUT @E5100;"OUTPRTMEMP? 2"` `ENTER @E5100;A`

OUTPSTIM?

Outputs the stimulus array data from the active channel. (Query only)

Query Response	{*numeric (1)*} {*numeric (2)*} . . . {*numeric (n)*} (n = number of points)

Examples	`DIM A(1:201)` *NOP: 201*
	`OUTPUT @E5100;"OUTPSTIM?"`
	`ENTER @E5100;A(*)`

OUTPTITL?

Outputs the display title for the active channel. (Query only)

Query Response	{*string*}: less than 54 characters
Examples	`OUTPUT @E5100;"OUTPTITL?"` `ENTER @E5100;A$`

OUTPTRAC?␣<value>

Outputs the phase value to trap at a measured point and the setting of I/O port at the point specified by the parameter. The phase value and the I/O port setting are specified by `INPUTRAC` command. (Query only, Option 022 only)

Parameter Description	{*value*} : measurement point number
Query Response	{*value*} {0|1} {*value*} : Phase value {1|0} : I/O port setting (ON/OFF)
Examples	`OUTPUT @E5100;"OUTPTRAC? 10"` `ENTER @E5100;A,B`

OUTPTRACB?␣<*value1*>,<*value2*>

Outputs the data outputted to the I/O port when the measured value of phase reaches the limit value. The data outputted to the I/O port is specified by `INPUTRACB` command. (Query only,Option 022 only)

Parameter Description	<*value1*> : Measurement point number <*value2*> : Output port number (1 to 6)
Query Response	{*value*} {*value*} : Data
Examples	`OUTPUT @E5100;"INPUTRACB? 10,1"` `ENTER @E5100;A`

OUTPXF2?␣<*value1*>,<*value2*>,<*value3*>, ... ,<*valuen + 3*>

Outputs filter parameters within the range specified by the `ANARANG` command. This command outputs the same parameters as `OUTPXFIL` and outputs up to 20 sets of frequency offsets from center of the analysis range to left and right cutoff points. (Query only)

Where,

- If two cutoff points which are x_1 dB below the maximum peak are not found, zeros will be returned for all parameters.
- If two cutoff points which are X_n dB below the maximum value are not found, zeroes will be returned for $\Delta f_{left\ n}$ and $\Delta f_{right\ n}$.
- If no peaks are found, zeroes will be returned for $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, $Pole_{stim2}$.

Parameter Description	<*value1*> : defference from the maxmum value <*value2*> : stop frequency in rejection band <*value3*> : start frequency of spurious level range <*value4*> : offset to the maximum peak value to determine the cutoff points x_1 [dB] <*value5*> : offset to the maximum peak value to determine the cutoff points x_2 [dB] ⋮ <*valuen+3*> : offset to the maximum peak value to determine the cutoff points x_n [dB] ($2 \leq n \leq 20$)
Query Response	{*value1*}{*value2*}{*value2*}{*value2*}{*value2*}{*value2n+11*} {*value1*}:insertion loss {*value2*}:x_1dB bandwidth {*value3*}:center frequency {*value4*}:Q {*value5*}:passband ripple {*value6*}:rejection level {*value7*}:spurious levle {*value8*}:negative peak in the left range ($pole_{x1}$) {*value9*}:stimulus value of negative peak in the left range ($pole_{stim1}$) {*value10*}:negative peak in the right range ($pole_{x2}$) {*value11*}:stimulus value of negative peak in the right range ($pole_{stim2}$) {*value12*}:frequency offset from center of the analysis range to left cutoff point (Δf_{left1}) {*value13*}:frequency offset from center of the analysis range to right cutoff point (Δf_{right1}) {*value14*}:Δf_{left2} {*value15*}:Δf_{right2} ⋮ {*value2n+10*}:$\Delta f_{left\ n}$ {*value2n+11*}:$\Delta_{fright\ n}$
Examples	`OUTPUT @E5100;"OUTPXF2? 3,90M,95M,3,10,12,15,20"` `ENTER @E5100;A(10),B(10)`

OUTPXFIL?␣<*value1*>,<*value2*>,<*value3*>,<*value4*>,<*value5*>

Outputs filter parameters within the range specified by the `ANARANG` command. Command parameter sets the offset of *x1*dB and *x2*dB to the maximum peak value to determine the cutoff points, same parameter as `POLE?`, and f_1 and f_2 for determining the rejection level and the spurious level respectively. For details, refer to Appendix D. (Data format:loss, bandwidth, center frequency, Q, ΔL.F1, ΔR.F1, ΔL.F2, ΔR.F2, passband ripple, rejection level, spurious level, pole_x1, pole_stimulus1, pole_x2, pole_stimulus2) (Query only)

OUTPXFIL?␣*<value1>*,*<value2>*,*<value3>*,*<value4>*,*<value5>*

Parameter Description	*<numeric1>* : offset to the maximum peak value to determine the cutoff points x_1[dB] *<numeric2>* : offset to the maximum peak value to determine the cutoff points x_2[dB] *<numeric3>* : defference from the maxmum value (same parameter as `POLE?`) *<numeric4>* : stop frequency in rejection band *<numeric5>* : start frequency of spurious level range
Query Response	{*numeric1*}, {*numeric2*}, ... , {*numeric15*} {*numeric1*} : insertion loss {*numeric2*} : x_1dB bandwidth {*numeric3*} : center frequency {*numeric4*} : Q {*numeric5*} : frequency offset from center of the analysis range to left cutoff point (ΔL.F1) {*numeric6*} : frequency offset from center of the analysis range to right cutoff point (ΔR.F1) {*numeric7*} : frequency offset from center of the analysis range to left cutoff point (ΔL.F2) {*numeric8*} : frequency offset from center of the analysis range to right cutoff point (ΔR.F2) {*numeric9*} : passband ripple {*numeric10*} : rejection level {*numeric11*} : spurious levle {*numeric12*} : negative peak in the left range (pole_x1) {*numeric13*} : stimulus value of negative peak in the left range (pole_stimulus1) {*numeric14*} : negative peak in the right range (pole_x2) {*numeric15*} : stimulus value of negative peak in the right range (pole_stimulus2)

Examples	`DIM ANS(1:15)`
	`OUTPUT @E5100;"OUTPXFIL? ";-10,-20,-40,1E3,1E3`
	`ENTER @E5100;ANS(*)`

PARS␣{OFF|ON|0|1}

Sets the partial search of the marker search function on or off.

Equivalent Key Sequence	(Marker) `MKR SEARCH []` `SEARCH RANGE` `PART SRCH ON OFF`
Parameter Description	`OFF` or 0 : partial search OFF `ON` or 1 : partial search ON
Query Response	{1|0}

Examples	`OUTPUT @E5100;"PARS ON"`
	
	`OUTPUT @E5100;"PARS?"`
	`ENTER @E5100;A`

PARSMODE␣{OFF|ON|0|1}

Sets the partial sweep on or off. For more information on the partial sweep function, refer to Appendix I.

Parameter Description	`OFF` or `0` : partial sweep OFF `ON` or `1` : partial sweep ON
Query Response	{1|0}

Examples	`OUTPUT @E5100;"PARSMODE ON"`
	
	`OUTPUT @E5100;"PARSMODE?"`
	`ENTER @E5100;A`

PARSRANG␣*<value1>*, *<value2>*

Set the start point and the end point for the partial sweep using the measurement points. If the start and end points are not between 1 to the value specified by `Number of Point`, the analyzer will perform a partial sweep within the possible range. For more information on the partial sweep function, refer to Appendix I

Parameter Description	{*value1*} : start point of partial sweep {*value2*} : end point of partial sweep
Parameter Range	1 to 1601
Query Response	*<value1>*, *<value2>*

Examples	`OUTPUT @E5100;"PARSRANG 10,50"`
	
	`OUTPUT @E5100;"PARSRANG?"`
	`ENTER @E5100;A,B`

PEAK?

Outputs the maximum local maximum value and its stimulus within range which is set by the `ANARANG` command. For more information, refer to Appendix D. (Data format: maximum Local-maximum value, stimulus)

Query Response	{*numeric1*} {*numeric 2*} {*numeric1*} : maxmum value {*numeric2*} : stimulus
Examples	`OUTPUT @E5100;"PEAK?"` `ENTER @E5100;A,B`

PEAKLIST?␣<*value*>

Outputs the *n* peak values in order of the peak value within the range specified by the ANARANG command.

Parameter Range	1 to NOP
Query Response	{*value*}

Examples	`DIM A(1:5,1:2)`
	`OUTPUT @E5100;"PEAKLIST? 5"`
	`ENTER @E5100;A(*)`

PEAKSORT?␣<*value*>

Searches n peaks starting from the largest peak value within the range specified by the ANARANG commands. Outputs the n stimulus (frequency or level) values in increasing order.

Parameter Description	The number of searched peak
Parameter Range	1 to 100
Query Response	{*value1*},{*value2*}, ...

Examples	`DIM A(1:5,1:2)`
	`OUTPUT @E5100;"PEAKSORT? 5"`
	`ENTER @E5100;A(*)`

PHAO␣<*value*>

Adds or subtracts a phase offset.

Equivalent Key Sequence	(Display) PHASE OFFSET
Parameter Range	−360 to +360 [°]
Query Response	{*value*}

Examples	`OUTPUT @E5100;"PHAO 90"`
	
	`OUTPUT @E5100;"PHAO?"`
	`ENTER @E5100;A`

PHAS

This command equals FMT PHAS. See FMT.

PICIRC␣{OFF|ON|0|1}

Selects to use π network test fixture. When `PICIRC` is set to ON, the analyser sets the test signal power level so that the power level appried to the DUT is equal to the power setting value, even the power unit is selected as W or A. This command uses the CI value set by `CIVAL`. The power unit can be selected by `POWU`.

Equivalent Key Sequence	(System) `PI CIRCUIT ON off`
Parameter Description	`ON` or `1`: π network test fixture is used `OFF` or `0` : π network test fixture is not used
Query Response	{1|0}

Examples	`OUTPUT @E5100;"PICIRC ON"`
	
	`OUTPUT @E5100;"PICIRC?"`
	`ENTER @E5100;A`

POIN␣<*value*>

Sets the number of the data points per sweep.

Equivalent Key Sequence	(Sweep) `NUMBER OF POINTS`
Parameter Range	2 to 1601
Query Response	{*value*}

Examples	`OUTPUT @E5100;"POIN 201"`
	
	`OUTPUT @E5100;"POIN?"`
	`ENTER @E5100;A`

POLE?␣<*value*>

Outputs the first found local minimum for both side from the maximum point below the value which is the subtracted parameter from the maximum value. For example, to specify as -10 dB down, a command parameter becomes a `-10`. For more information, refer to Appendix D. (Data format: left local minimum, stimulus, right local minimum, stimulus) (Query only)

Parameter Range	TBD
Query Response	{*numeric1*} {*numeric 2*} {*numeric 3*} {*numeric 4*} {*numeric1*} : local mininum (left) {*numeric 2*} : stimulus {*numeric 3*} : local minimum (right) {*numeric 4*} : stimulus
Examples	`OUTPUT @E5100;"POLE? -50"` `ENTER @E5100;A,B,C,D`

POSL

Sets output of the 24-bit I/O port to the positive logic. (No Query)

Examples	`OUTPUT @E5100;"POSL"`

POWE␣<*value*>

Sets the source output level.

Equivalent Key Sequence	(Sweep) POWER
Parameter Range	−29 to +5 dBm (−48 to +22 dBm option 010 only) (−9 dBm to +11 dBm option 001 only) (−15 dBm to +5 dBm option 002 only) (−12 dBm to +8 dBm option 003 only) (−48 dBm to +22 dBm option 001 and 010 only) (−54 dBm to +16 dBm option 002 and 010 only) (−51 dBm to +19 dBm option 003 and 010 only) (−52 dBm to +18 dBm option 600 (RF OUT 1) only) (−65 dBm to +5 dBm option 600 (RF OUT 2) only)
Query Response	{*numeric*}

Examples	`OUTPUT @E5100;"POWE 0"`
	
	`OUTPUT @E5100;"POWE?"`
	`ENTER @E5100;A`

POWS

This command equals SWPT POWE. See SWPT.

POWU␣{DBM|WATT|AMP}

Selelcts the unit for the test signal power input.

Equivalent Key Sequence	(Menu2) MORE SRC UNIT [ ]
Parameter Description	DBM : dBm WATT : W AMP : A
Query Response	{DBM|WATT|AMP}

Examples	`OUTPUT @E5100;"POWU WATT"`
	
	`OUTPUT @E5100;"POWU?"`
	`ENTER @E5100;A$`

PRES

Presets the instrument state.

Equivalent Key Sequence	(Preset)
Examples	`OUTPUT @E5100;"PRES"`

PRINALL

Copies the measurement display to the printer according to plotting options.

Equivalent Key Sequence	(System) PRINT
Examples	`OUTPUT @E5100;"PRINALL"`

PRIR␣<*string*>

Displays the query value of the GPIB command given as a parameter. The query value is displayed on the Instrument BASIC area.

Parameter Description	<*string*>:Query command
Examples	`OUTPUT @E5100;"PRIR ""*IDN?"""`

PTABORT␣{OFF|ON|0|1}

Sets the measurement abort ON/OFF when the phase value is over the limit during the tracking. (Option 023 only)

Parameter Description	ON or 1 : Abort a measurement when the tracking is failed OFF or 0 : Continue a measurement even the tracking is failed
Query Response	{1|0}

Examples	`OUTPUT @E5100;"PTABORT ON"`
	
	`OUTPUT @E5100;"PTABORT?"`
	`ENTER @E5100;A`

PTFOVHD␣<*value*>

Input the parameters required to display the time base at the temperature characteristic measurement (the aging measurement). Refer to "Compensation of Sweep Time for Aging Characteristics" in Appendix H for inputting. (Option 023 only)

Parameter Range	0 to 1 sec
Query Response	{*value*}

PTFR␣<*value*>

Input the start frequency for tracking. (Option 023 only)

Parameter Range	10 kHz to 300 MHz
Query Response	{*value*}

Examples	`OUTPUT @E5100;"PTFR 199 kHz"`
	
	`OUTPUT @E5100;"PTFR?"`
	`ENTER @E5100;A`

PTFRSR␣<*value*>

Sets the range for searching for F_r. (Option 023 only)

Parameter Range	0 Hz to 100 kHz
Query Response	{*value*} (Hz)

Examples	`OUTPUT @E5100;"PTFRSR 5000"`
	
	`OUTPUT @E5100;"PTFRSR?"`
	`ENTER @E5100;A`

PTPARA␣<*value*>

Sets the tracking parameter. The tracking paramter is given by the SRCHFR? command query. (Option 023 only)

Parameter Range	−1000 to 1000
Query Response	{*value*}

Examples	`OUTPUT @E5100;"SRCHFR? 2,0"`
	`ENTER @E5100;Fr,Ci,Param`
	`OUTPUT @E5100;"PTPARA";Param`
	
	`OUTPUT @E5100;"PTPARA?"`
	`ENTER @E5100;A`

PTRACK␣{OFF|ON|0|1}

Set the phase tracking ON/OFF. (Option 023 only)

Parameter Description	OFF or 0 : The phase tracking is OFF ON or 1 : The phase tracking is ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"PTRACK ON"`
	
	`OUTPUT @E5100;"PTRACK?"`
	`ENTER @E5100;A`

PTREPN␣<*value*>

Sets the number of tracking on each point. (Option 023 only)

Parameter Range	1 to 1,000,000
Query Response	{*value*}

Examples	`OUTPUT @E5100;"PTREPN 5"`
	
	`OUTPUT @E5100;"PTREPN?"`
	`ENTER @E5100;A`

PTSTAT?

Returns the status of the phase tracking. (Query only,Option 023 only)

Query Response	{0|1} 0 : Error encountered during phase tracking 1 : The phase tracking was successful.
Examples	`OUTPUT @E5100;"PTSTAT?"` `ENTER @E5100;A`

PTTRGLMT␣<*value*>

Defines the range value for a phase, which is used for the phase tracking. (Option 023 only)

Parameter Range	0° to 180°
Query Response	{*value*}

Examples	`OUTPUT @E5100;"PTTRGLMT 8"`
	
	`OUTPUT @E5100;"PTTRGLMT?"`
	`ENTER @E5100;A`

PTTRGPHS␣<*value*>

Defines the phase at the resonant point. (Option 023 only)

Parameter Range	$-180°$ to $180°$
Query Response	{*value*}

Examples	`OUTPUT @E5100;"PTTRGPHS 0"`
	
	`OUTPUT @E5100;"PTTRGPHS?"`
	`ENTER @E5100;A`

PURG␣<*value*>

Removes a file saved on the disk in the built-in flexible disk drive. (No Query)

Equivalent Key Sequence	(Save/Recall) FILE UTILITY PURGE FILE
Parameter Description	<*string*> File name, up to 10 characters including the extension
Examples	`OUTPUT @E5100;"PURG ""TEST_S"""`

RAID

Completes the response and isolation calibration. (No Query)

Equivalent Key Sequence	(Cal) DONE:
Examples	`OUTPUT @E5100;"RAID"`

RAIISOL

Executes the isolation measurement for the response and isolation calibration. (No Query)

Equivalent Key Sequence	(Cal) ISOL'N STD
Examples	`OUTPUT @E5100;"RAIISOL"`

RAIRESP

Selects the response class for the response and isolation calibration. (No Query)

Equivalent Key Sequence	(Cal) RESPONSE
Examples	`OUTPUT @E5100;"RAIRESP"`

READ?

Reads data from a file that has been read-enabled using the `ROPEN` command. The returned data is in the fixed length block format defined in IEEE488.2. The fixed length block format, as shown in Figure J-2, consists of a header part indicating the data size and an actual data part. In the case of the E5100A/B, the number of digits to indicate the data size is 6 and the maximum length of the actual data part is 16 Kbytes. If a file contains data greater than 16 Kbytes, execute this command repeatedly to read it. Note that acceptable file format for this command is the DOS format.

Generally, this command is used in combination with the `ROPEN` command and the `CLOSE` command, as shown in Figure J-1. (Query only)

- Query Response

  {*block*} <new line><^END>

RECD␣<*value*>

Loads the instrument states or data from the disk in the built-in flexible disk drive. (No Query)

Equivalent Key Sequence	(Save/Recall) `RECALL`
Parameter Range	File name, Up to 10 characters including the extension
Query Response	{*string*}
Examples	`OUTPUT @E5100;"RECD ""TEST_S"""`

REFP␣<*value*>

Sets the position of the reference line on the graticule of a Cartesian format.

Equivalent Key Sequence	(Display) `SCALE MENU` `REFERENCE POSITION`
Parameter Range	0 to 10 [Div]
Query Response	{*value*}

Examples	`OUTPUT @E5100;"REFP 0"`
	`OUTPUT @E5100;"REFP?"`
	`ENTER @E5100;A`

REFV␣<*value*>

Changes the value of the reference line, moving the measurement trace correspondingly.

REFV␣<*value*>

Equivalent Key Sequence	(Display) Scale Menu REFERENCE VALUE
Query Response	{*value*}

Examples	`OUTPUT @E5100;"REFV 0"`
	
	`OUTPUT @E5100;"REFV?"`
	`ENTER @E5100;A`

RESAVD␣<*string*>

Updates an already saved file on the disk in the built-in flexible disk drive. (No Query)

Equivalent Key Sequence	(Save/Recall) SAVE RE-SAVE
Parameter Range	File name up to 10 characters including the extension
Examples	`OUTPUT @E5100;"RESAVD ""TEST_S"""`

RESPDONE

Completes the response calibration. (No Query)

Equivalent Key Sequence	(Cal) DONE:
Examples	`OUTPUT @E5100;"RESPDONE"`

REST

Aborts the sweep in progress, then restarts the measurement. (No Query)

Equivalent Key Sequence	(Trigger) MEASURE RESTART
Examples	`OUTPUT @E5100;"REST"`

RFOUTSW␣{1|2}

Changes the RF OUTPUT port. (option 003 only)

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU RF OUTPUT [ ]
Parameter Description	1 : RF OUT 1 2 : RF OUT 2
Query Response	{0|1}

Examples	`OUTPUT @E5100;"RFOUTSW 1"`
	
	`OUTPUT @E5100;"RFOUTSW?"`
	`ENTER @E5100;A`

ROPEN␣<*string*>

Makes a specified file read-enabled. If the file does not exist, an error occurs.

Generally, this command is used in combination with the `READ?` command and the `CLOSE` command, as shown in Figure J-1. (No query)

Parameter	Description
<*string*>	File name of up to 12 characters including its extension

RPLENV?

Searches all sets of neighboring peaks and their included valleys for the maximum perpendicular height from the valley minimum point included between neighboring peaks, to the intersection of an imaginary slope line drawn between the maximum peak points of the neighboring peaks in range specified by `ANARANG`, and outputs the resultant data via GPIB. For details, refer to "RPLENV?" in Appendix D in Appendix D. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"RPLENV?"` `ENTER @E5100;A`

RPLHEI?

Searches for the maximum height between neighboring ripple peaks and outputs the resultant data via GPIB. For details, refer to "RPLHEI?" in Appendix D in Appendix D. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"RPLHEI?"` `ENTER @E5100;A`

RPLLHEI?

Searches for the maximum height between neighboring ripple peaks (measured from the ripple maximum peak point to the valley minimum point to the left of the ripple peak) and outputs the resultant data via GPIB. For details, refer to "RPLLHEI?" in Appendix D in Appendix D. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"RPLLHEI?"` `ENTER @E5100;A`

RPLMEA?

Averages all heights between neighboring local maximums and minimums within a specified range and outputs the result by GPIB. If no ripple is detected, a zero is returned. For details, refer to Appendix D. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"RPLMEA?"` `ENTER @E5100;A`

RPLMM?

Outputs the difference value between the maximum and minimum values within the range specified with the `ANARANG` command. (The maximum and minimum values are same as ones `OTUPMINMAX?` outputs.) (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"RPLMM?"` `ENTER @E5100;A`

RPLPP?

Searches for the maximum ripple peak to peak value and outputs the resultant data via GPIB. For details, refer to "RPLPP?" in Appendix D in Appendix D. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"RPLPP?"` `ENTER @E5100;A`

RPLPPS?

Searches for the maximum ripple peak to peak value and outputs the resultant data and stimulus values at these points. (Query only)

Query Response	{*numeric1*} {*numeric2*} {*numeric3*} {*numeric1*} : maximum ripple peak {*numeric2*} : stimulus value at local maximum point {*numeric3*} : stimulus value at local minimum point
Examples	`OUTPUT @E5100;"RPLPPS?"` `ENTER @E5100;A,B,C`

RPLRHEI?

Searches for the maximum height between neighboring ripple peaks (measured from the ripple peak to the valley point to the right of the ripple peak) and outputs the resultant data via GPIB. For details, refer to "RPLRHEI?" in Appendix D in Appendix D. (Query only)

Query Response	{*value*}
Examples	`OUTPUT @E5100;"RPLRHEI?"` `ENTER @E5100;A`

RPLVAL?

Outputs the maximum sum of the difference between the local minimum and the both sides local maximum, and the stimulus of the corresponding local minimum within range which is specified by `ANARANG` command. For more information, refer to Appendix D. (Data format: sum, stimulus) (Query only)

Query Response	{*numeric1*}{*numeric2*} {*numeric1*} : Maximum value of sum {*numeric2*} : stimulus value of local minimum
Examples	`OUTPUT @E5100;"RPLVAL?"` `ENTER @E5100;A,B`

RPOS

This command equals `REFP`. See `REFP`.

SADD

Adds a new segment to a list sweep table. (No Query)

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU EDIT LIST
Examples	`OUTPUT @E5100;"SADD"`

SAV1

Saves the 3 term calibration results. (No Query)

Equivalent Key Sequence	(CAL) DONE:
Examples	`OUTPUT @E5100;"SAV1"`

SAVCA␣{OFF|ON|0|1}

Selects whether or not the calibration coefficients arrays are to be saved.

Equivalent Key Sequence	(Save/Recall) SAVE DEFINE SAVE DATA CAL ARRAY ON OFF
Parameter Description	OFF or 0 : calibration coefficients arrays are not saved ON or 1 : calibration coefficients arrays are saved
Query Response	{ 1|0}

Examples	`OUTPUT @E5100;"SAVCA ON"`
	
	`OUTPUT @E5100;"SAVCA?"`
	`ENTER @E5100;A`

SAVDA␣{OFF|ON|0|1}

Sets the data array to be saved (ON) or not (OFF).

Equivalent Key Sequence	(Save/Recall) SAVE DEFINE SAVE DATA DATA ARRAY ON OFF
Parameter Description	OFF or 0 : OFF ON or 1 : ON
Query Response	{1|0}

Examples	`OUTPUT @E5100;"SAVDA ON"`
	
	`OUTPUT @E5100;"SAVDA?"`
	`ENTER @E5100;A`

SAVDALL␣<*string*>

Saves the instrument states, the data array, and the memory array to the disk in the built-in flexible disk drive. (No Query)

Equivalent Key Sequence	(Save/Recall) SAVE ALL
Parameter Range	File name, up to 8 characters
Examples	`OUTPUT @E5100;"SAVDALL ""TEST"""`

SAVDASC␣<*string*>

Save the current measurement data in ASCII file format. (No Query)

Equivalent Key Sequence	(Save/Recall) SAVE DATA ONLY (ASCII SAVE)
Parameter Range	File name, up to 8 characters
Examples	`OUTPUT @E5100;"SAVDASC ""DATASCII"""`

SAVDDAT␣<*string*>

Saves the internal data arrays. (No Query)

Equivalent Key Sequence	(Save/Recall) SAVE DATA ONLY (BINARY SAVE)
Parameter Range	File name up to 8 characters
Query Response	{*string*}
Examples	`OUTPUT @E5100;"SAVDDAT ""DATABIN"""`

SAVDGRAP␣<*string*>

Saves the display image to the disk in the built-in flexible disk drive or the RAM disk drive. Two graphic file formats, TIFF and PCL, are available. `GRAPFORM` command is used to select the format. (No Query)

Equivalent Key Sequence	(Save/Recall) SAVE GRAPHICS
Parameter Range	Up to 8 characters
Examples	`OUTPUT @E5100;"SAVDGRAP ""RESULT1"""`

SAVDMNU3␣<*string*>

Save the user definition of softkey accessed from (Menu3). (No Query)

Equivalent Key Sequence	(Save/Recall) SAVE MENU3
Parameter Range	Up to 8 characters
Examples	`OUTPUT @E5100;"SAVDMNU3 ""USERMENU"""`

SAVDSTA␣<*string*>

Saves only the instrument states and the calibration coefficients to the disk in the built-in flexible disk drive. (No Query)

Equivalent Key Sequence	(Save/Recall) SAVE STATE ONLY
Parameter Range	File name up to 8 characters
Examples	`OUTPUT @E5100;"SAVDSTA ""STA1"""`

SAVFA␣{OFF|ON|0|1}

Sets the formatted arrays to be saved (ON) or not (OFF).

Equivalent Key Sequence	(Save/Recall) SAVE DEFINE SAVE DATA FORMD ARRAY ON OFF
Parameter Description	OFF or 0 : formatted arrays are not saved ON or 1 : formatted arrays are saved
Query Response	{1|0}

Examples	`OUTPUT @E5100;"SAVFA ON"`
	
	`OUTPUT @E5100;"SAVFA?"`
	`ENTER @E5100;A`

SAVMA␣{ON|OFF|1|0}

Sets the memory array to be saved (ON) or not (OFF).

Equivalent Key Sequence	(Save/Recall) SAVE DEFINE SAVE DATA MEM ARRAY ON OFF
Parameter Description	OFF or 0 : memory array are not saved ON or 1 : memory array are saved
Query Response	{1|0}

Examples	`OUTPUT @E5100;"SAVMA ON"`
	
	`OUTPUT @E5100;"SAVMA?"`
	`ENTER @E5100;A`

SAVRA⊔{OFF|ON|0|1}

Sets the raw data arrays to be saved (ON) or not (OFF).

Equivalent Key Sequence	(Save/Recall) SAVE DEFINE SAVE DATA RAW ARRAY ON OFF
Parameter Description	OFF or 0 : raw arrays are not saved ON or 1 : raw arrays are saved
Query Response	{1|0}

Examples	`OUTPUT @E5100;"SAVRA ON"`
	
	`OUTPUT @E5100;"SAVRA?"`
	`ENTER @E5100;A`

SAVTA⊔{OFF|ON|0|1}

Sets the trace arrays to be saved (ON) or not (OFF).

Equivalent Key Sequence	(Save/Recall) SAVE DEFINE SAVE DATA TRACE ARRAY ON OFF
Parameter Description	OFF or 0 : trace arrays are not saved ON or 1 : trace arrays are saved
Query Response	{1|0}

Examples	`OUTPUT @E5100;"SAVTA ON"`
	
	`OUTPUT @E5100;"SAVTA?"`
	`ENTER @E5100;A`

SAVTMA⊔{OFF|ON|0|1}

Sets the memory trace arrays to be saved (ON) or not (OFF).

SAVTMA␣{OFF|ON|0|1}

Equivalent Key Sequence	(Save/Recall) SAVE DEFINE SAVE DATA SUB ARRAY ON OFF
Parameter Description	OFF or 0 : sub trace arrays are not saved ON or 1 : sub trace arrays are saved
Query Response	{1|0}

Examples	`OUTPUT @E5100;"SAVTMA ON"`
	
	`OUTPUT @E5100;"SAVTMA?"`
	`ENTER @E5100;A`

SCAFDATA

Selects the data trace to be scaled. (No Query)

Equivalent Key Sequence	(Display) SCALE MENU SCALL FOR [ ]
Examples	`OUTPUT @E5100;"SCAFDATA"`

SCAFMEMO

Selects the sub-trace to be scaled. (No Query)

Equivalent Key Sequence	(Display) SCALE MENU SCALL FOR [ ]
Examples	`OUTPUT @E5100;"SCAFMEM"`

SCAL␣<*value*>

Changes the response value scale per graticule division.

Equivalent Key Sequence	(Display) SCALE MENU SCALE/DIV
Parameter Description	0.001 to 500 : (Log mag format) 0.01 to 500 : (Phase format) 1×10^{-14} to 10 : (Delay format) 1×10^{-11} to 10000 : (Lin mag, Real, and Imaginary formats)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"SCAL 1"`
	
	`OUTPUT @E5100;"SCAL?"`
	`ENTER @E5100;A`

SCAY␣{1|0}

Selects Y-axis scale from log sacale and linear scale. (No Query)

Equivalent Key Sequence	(Meas/Format) FORMAT Y-AXIS[ ]
Parameter Description	0 : Linear scale 1 : LOG scale
Examples	`OUTPUT @E5100;"SCAY 1"`

SDEL

Deletes a segment from a list sweep table. (No Query)

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU EDIT LIST
Examples	`OUTPUT @E5100;"SDEL"`

SDON

Completes editing a segment of a list sweep table. (No Query)

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU EDIT LIST
Examples	`OUTPUT @E5100;"SDON"`

SEAL

Searches the trace for the next occurrence of the target value to the left of the marker. (No Query)

Examples	`OUTPUT @E5100;"SEAL"`

SEALMAX

Moves the active marker to the maximum peak point on the trace in the search range. (No Query)

Examples	`OUTPUT @E5100;"SEALMAX"`

SEALMIN

Moves the active marker to the minimum peak point on the trace in the search range. (No Query)

Examples	`OUTPUT @E5100;"SEALMIN"`

SEAM␣{OFF|MAX|MIN|TARG|MEAN|LMAX|LMIN|PPEAK}

Selects the marker search function. (Query)

Equivalent Key Sequence	(Marker) MKR SEARCH[ ] SEARCH:MAX, MIN, TARGET
Parameter Description	OFF : Marker search function OFF MAX : Maximum MIN : Minimum TARG : Target MEAN : Mean LMAX : Local maximum LMIN : Local minimum PPEAK : Peak to peak
Query Response	{OFF|MAX|MIN|TARG|MEAN|LMAX|LMIN|PPEAK}

Examples	`OUTPUT @E5100;"SEAM PEAK"`
	
	`OUTPUT @E5100;"SEAM?"`
	`ENTER @E5100;A$`

SEAMAX

Moves the active marker to the maximum point on the trace. (No Query)

Equivalent Key Sequence	(Marker) MKR SEARCH[ ] SEARCH:MAX
Examples	`OUTPUT @E5100;"SEAMAX"`

SEAMEAN

Moves the active marker to the mean point on the trace. (No Query)

Examples	`OUTPUT @E5100;"SEAMEAN"`

SEAMIN

Moves the active marker to the minimum point on the trace. (No Query)

Equivalent Key Sequence	(Marker) MKR SEARCH[ ] MIN
Examples	`OUTPUT @E5100;"SEAMIN"`

SEAOFF

Turns off the marker search function. (No Query)

Equivalent Key Sequence	(Marker) MKR SEARCH [] TRACKING ON OFF
Examples	`OUTPUT @E5100;"SEAOFF"`

SEAPPEAK

Moves the active marker and the delta reference marker to the maximum peak point and the minimum peak point on the trace in the search range. (No Query)

Examples	`OUTPUT @E5100;"SEAPPEAK"`

SEAR

Searches the trace for the next occurrence of the target value to the right of the marker. (No Query)

Examples	`OUTPUT @E5100;"SEAR"`

SEARSTOR

Stores the search range, which is defined between the active marker and the delta reference marker. (No Query)

Equivalent Key Sequence	(Marker) MKR SEARCH [ ] SEARCH RANGE SERCH RNG STORE
Examples	`OUTPUT @E5100;"SEARSTR"`

SEATARG␣<*value*>

Places the active marker at a specified target point on a trace.

Equivalent Key Sequence	(Marker) MKR SEARCH[ ] TARGET
Query Response	{*value*}

Examples	`OUTPUT @E5100;"SEATARG 0"`
	
	`OUTPUT @E5100;"SEATARG?"`
	`ENTER @E5100;A`

SEDI␣<*value*>

Determines a segment of a list sweep table to be modified.

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU EDIT LIST
Parameter Range	1 to 31
Query Response	{*value*}

Examples	`OUTPUT @E5100;"SEDI"`
	
	`OUTPUT @E5100;"SEDI?"`
	`ENTER @E5100;A`

SEET␣{OFF|ON|0|1}

`SEET ON` makes the analyzer display both of text displayed using `PRINT` statement of HP Instrument BASIC and measurement traces.

Parameter Description	`OFF` or 0 : operation completion beeper OFF `ON` or 1 : operation completion beeper ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"SEET ON"`
	
	`OUTPUT @E5100;"SEET?"`
	`ENTER @E5100;A`

SET1PT␣<*value*>

Set the output frequency(level) of the RF OUT for the active channel to the output frequency (level) on the measurement point specified by the parameter. For example, by using the **SET1PT␣1** command when switching over the output frequency after a single sweep (changing it from the stop frequency to the start frequency), you can reduce time during the switch and remove the signal distortion.

(No Query)

Parameter Range	1 to NOP
Examples	`OUTPUT @E5100;"SET1PT 1"`

SETCDATE␣<*numeric (year)*>,<*numeric (month)*>,<*numeric (day)*>

Changes date of the internal clock.

Equivalent Key Sequence	System SET CLOCK
Parameter Range	<*numeric (year)*> : 1901 to 2059 <*numeric (month)*> : 1 to 12 <*numeric (day)*> : 1 to 31
Query Response	{*numeric (year)*} {*numeric (month)*} {*numeric (day)*}

Examples	`OUTPUT @E5100;"SETCDATE 1993,1,1"`
	
	`OUTPUT @E5100;"SETCDATE?"`
	`ENTER @E5100;A,B,C`

SETCTIME␣<*numeric (hour)*>,<*numeric (minute)*>,<*numeric (second)*>

Changes time of the internal clock.

Equivalent Key Sequence	System SET CLOCK
Parameter Range	<*numeric (hour)*> : 0 to 23 <*numeric (minute)*> : 0 to 59 <*numeric (second)*> : 0 to 59
Query Response	{*numeric (hour)*}{*numeric (minute)*}{*numeric (second)*}

Examples	`OUTPUT @E5100;"SETCTIME 10,30,0"`
	
	`OUTPUT @E5100;"SETCTIME?"`
	`ENTER @E5100;A,B,C`

SETZ␣<*value*>

Sets the characteristic impedance used by the E5100A/B in calculating measured impedance.

Equivalent Key Sequence	Cal SET Z0
Parameter Range	0.1 to 5×10^6 [Ω]
Query Response	{*value*}

Examples	`OUTPUT @E5100;"SETZ 75"`
	
	`OUTPUT @E5100;"SETZ?"`
	`ENTER @E5100;A`

SING?

Makes a single measurement sweep and returns 1 when the sweep is completed. (Query Only)

Equivalent Key Sequence	(Trigger) SINGLE
Query Response	1
Examples	`OUTPUT @E5100;"SING?"` `ENTER @E5100A`

SINSPEAK␣{OFF|ON|0|1}

SINSPEAK ON makes the analyzer search the maximum or minimum point with each sweep.

Parameter Description	OFF or 0 : searching OFF ON or 1 : searching ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"SINSPEAK ON"`
	
	`OUTPUT @E5100;"SINSPEAK?"`
	`ENTER @E5100;A`

SMOO␣{OFF|ON|0|1}

Sets the smoothing function to ON or OFF.

Equivalent Key Sequence	(Display) SMOOTHING on OFF
Parameter Description	OFF or 0 : smoothing OFF ON or 1 : smoothing ON
Query Response	{1|0}

Examples	`OUTPUT @E5100;"SMOO ON"`
	
	`OUTPUT @E5100;"SMOO?"`
	`ENTER @E5100;A`

SMOOAPER␣<*value*>

Changes the value of the smoothing aperture as a percent of the span.

Equivalent Key Sequence	(Display) SMOOTHING APERTURE
Parameter Range	0.05 to 100 [%]
Query Response	{*value*}

Examples	`OUTPUT @E5100;"SMOOAPER 10"`
	
	`OUTPUT @E5100;"SMOOAPER?"`
	`ENTER @E5100;A`

SPAN␣<*value*>

Sets the frequency span of a segment about a specified center frequency. For power sweep, use `POWU` command to set unit for power sweep.

Equivalent Key Sequence	(SPAN)
Parameter Range	0 to 299.999×10^6 (=299.999 M) : Hz (frequency) 0 to 82 : dB (power)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"SPAN 100MHZ"`
	
	`OUTPUT @E5100;"SPAN?"`
	`ENTER @E5100;A`

SPLD␣{OFF|ON|0|1}

Sets the multi channel display mode.

Equivalent Key Sequence	(Display) SPLIT DISP ON off
Parameter Description	`OFF` or `0` : a full-screen single graticule display `ON` or `1` : a split display with two half-screen graticules
Query Response	{1|0}

Examples	`OUTPUT @E5100;"SPLD ON"`
	
	`OUTPUT @E5100;"SPLD?"`
	`ENTER @E5100;A`

SRCHFR?␣{1|2|3|4|5|6},<*value*>

Searches for the resonance frequency (F_r). (Query only, Option 023 only)

SRCHFR?␣{1|2|3|4|5|6},<*value*>

Parameter Description	Search mode: Rough (High speed) 1 ↔ 6 Finer (Slow) <*value*> : Waiting time during searching (sec)
Query Response	{*value1*}, {*value2*},{*value3*} {*value1*} : F_r [Hz] {*value2*} : CI [Ω] {*value3*} : Tracking parameter
Examples	`OUTPUT @E5100;"SRCHFR? 2,0"` `ENTER @E5100;A,B,C`

SRCHR␣<*Value*>

Use this command to specify the frequency range (span) in [Hz] when you make search for load resonance frequency (FL) of a resonator using the SRCHTRFL? command.

The narrower the frequency range, the shorter the measurement time required for the SRCHTRFL? command to determine the load resonance frequency, provided that the rest of the parameters remain the same. Note, however, that you must make sure that the resonance point lies within the specified range. This command returns −1 if no resonance point is found.

Note that this frequency range is internally linked to the range within which the search is made for the series resonance frequency (Fr) that is specified using the PTFRSR command (option 023). Therefore, if you specify one of these frequency ranges, the other range will be automatically set at the same value.

See the sample programs in Appendix K for more information.

Parameter Range	0 to 100000 (= 0 Hz to 100 kHz) Unit: Hz Default: 0
Query Response	{*Value*} {*Value*} : Frequency range in [Hz] for making search using the SRCHTRFL? command
Example	`OUTPUT @E5100;"SRCHR 10000"`

SRCHTRFL?␣{1|2|3|4|5|6},<*Value*>

Use this command to measure through search the load resonance frequency (FL) of a sample (resonator) with the known capacitance and determine the load resonance frequency for the target load capacitance (target CL) through corrections. (Query only)

Before executing this command, be sure to specify the actual load capacitance (actual CL), the target load capacitance, the nominal load resonance frequency, and the frequency range for making search using the CLACT, the CLTGT, the NOMF, and the SRCHR commands, respectively.

To ensure improved accuracy in corrections, be sure to connect a capacitance that is closest to the target capacitance. Note that this command is executed only for those channels that have been selected with the ANAOCHN command for waveform analysis.

See the sample programs in Appendix K for more information.

Parameter Description+	{1|2|3|4|5|6}: Select the desired search mode, or intervals at which the frequency changes. When you select "1", the frequency will change at the largest intervals of all. The greater the number you select, the smaller the intervals become, thus assuring increasingly high accuracy. As long as the wait time per frequency remains constant, the smaller the intervals, the longer time it takes before the search is complete. <*Value*>: Specify the wait time per frequency during search in [sec]. To prevent excessive search time, be sure not to specify a large value.
Query Response	{*Value*} {*Value*} : Load resonance frequency (FL) in [Hz] measured when the target load capacitance is specified using the CLTGT command
Example	`OUTPUT @E5100;"SRCHTRFL? 2,0.001"` `ENTER @E5100;A`

STANC

Measures the calibration standard in the THRU. (No Query)

Equivalent Key Sequence	(Cal) THRU
Examples	`OUTPUT @E5100;"STANA"`

STAR␣<*value*>

Defines the start value of the stimulus.

Equivalent Key Sequence	(START)
Parameter Range	10×10^3 (=10k) to 300×10^6 (=300M) Hz −9 to +11 dBm (−48 to +22 dBm option 010 only)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"STAR 100KHZ"`
	 `OUTPUT @E5100;"STAR?"`
	 `ENTER @E5100;A`

STAS␣<*value1*>,<*value2*>

Sets start and stop stimulus values.

STAS␣<*value1*>,<*value2*>

Equivalent Key Sequence	(START) (STOP)
Parameter Range	10×10^3 (=10k) to 300×10^6 (=300M) Hz −9 to +11 dBm (−48 to +22 dBm option 010 only)
Examples	`OUTPUT @E5100;"STAS 100KHZ,1MHZ"`

STAW␣<*value*>

Sets the wait time for sweep.

Parameter Range	0 to 100 (sec.)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"STAW 1"`
	
	`OUTPUT @E5100;"STAW?"`
	`ENTER @E5100;A`

STIDROUT{1-16}␣<*value*>

Sets stimulus of data trace up to 16 for `OUTPDATAT?` query. To execute `STIDROUT?` query, pass a number as the parameter.

Parameter Range	10×10^3 (=10k) to 300×10^6 (=300M) Hz −9 to +11 dBm (−48 to +22 dBm option 010 only)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"STIDROUT1 100KHZ"`
	
	`OUTPUT @E5100;"STIDROUT1?"`
	`ENTER @E5100;A`

STIMROUT{1-16}␣<*value*>

Sets stimulus of memory trace up to 16 for `OUTPDATAT?` query. To execute `STIDROUT?` query, pass a number as the parameter.

Parameter Range	10×10^3 (=10k) to 300×10^6 (=300M) Hz −9 to +11 dBm (−48 to +22 dBm option 010 only)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"STIMROUT1 100KHZ"`
	
	`OUTPUT @E5100;"STIMROUT1?"`
	`ENTER @E5100;A`

STOD{DISK|MEMO}

Selects mass storage device. (No Query)

Equivalent Key Sequence	(Save/Recall) FILE UTILITY STOR DEV [ ]
Parameter Range	STODDISK : internal flexible disc STODMEMO : internal RAM disk memory

Examples	`OUTPUT @E5100;"STODDISK"`
	
	`OUTPUT @E5100;"STODMEMO"`

STOMDISK

Stores the all files in the RAM disk to the FLASH disk. (No Query)

Equivalent Key Sequence	(Save/Recall) BACK UP MEMO DISK
Examples	`OUTPUT @E5100;"STOMDISK"`

STOP␣*<value>*

Defines the stop value of the stimulus.

Equivalent Key Sequence	(STOP)
Parameter Range	10×10^3 (=10k) to 300×10^6 (=300M) Hz -9 to $+11$ dBm (-48 to $+22$ dBm option 010 only)
Query Response	{*value*}

Examples	`OUTPUT @E5100;"STOP 100MHZ"`
	
	`OUTPUT @E5100;"STOP?"`
	`ENTER @E5100;A`

STPSIZE␣*<value>*

Specifies the frequency step for a list sweep table.

Equivalent Key Sequence	(SWEEP) SWEEP TYPE MENU EDIT LIST STEP SIZE
Parameter Range	10×10^3 (=10k) to 300×10^6 (=300M) Hz
Query Response	{*value*}

Examples	`OUTPUT @E5100;"STPSIZE 1MHZ"`
	
	`OUTPUT @E5100;"STPSIZE?"`
	`ENTER @E5100;A`

STPSIZE␣<*value*>

STR␣{OFF|ON|0|1}

Sets the trace storage ON or OFF.

Equivalent Key Sequence	(Display) STORAGE ON OFF
Parameter Description	ON or 1 : Display storage ON OFF or 0 : Display storage OFF
Query Response	{0|1}
Examples	`OUTPUT @E5100;"STR ON"`

SWED␣{DOWN|UP}

Sets the sweep direction.

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU SWEEP DIR [ ]
Parameter Description	UP : sweep from START to STOP DOWN : sweep from STOP to START
Query Response	{DOWN|UP}

Examples	`OUTPUT @E5100;"SWED DOWN"`
	
	`OUTPUT @E5100;"SWED?"`
	`ENTER @E5100;A$`

SWET␣<*value*>

Manually sets the sweep time.

Equivalent Key Sequence	(Sweep) SWEEP TIME [ ]
Parameter Range	6.0×10^{-4} to 86400[1] [sec]
Query Response	{*value*}

Examples	`OUTPUT @E5100;"SWET 1"`
	
	`OUTPUT @E5100;"SWET?"`
	`ENTER @E5100;A`

1 depends on stimulus settings

SWETAUTO

Automatically sets the sweep time. (No Query)

Equivalent Key Sequence	(Sweep) SWEEP TIME [ ] SWEEP TIME AUTO
Examples	`OUTPUT @E5100;"SWETAUTO ON"`

SWPT␣{LINF|LOGF|LIST|POWE|RAMPF}

Selects the sweep type.

Equivalent Key Sequence	(Sweep) SWEEP TYPE MENU LIN FREQ, POWER, LIST
Parameter Description	LINF : Linear frequency sweep LIST : Frequency list sweep POWE : Power sweep RAMPF : Ramp sweep
Query Response	{LINF|LIST|POWE|RAMPF}

Examples	`OUTPUT @E5100;"SWPT LINF"`
	
	`OUTPUT @E5100;"SWPT?"`
	`ENTER @E5100;A$`

TARL?␣<*value*>

Output stimulus of the first fund point which has a value specified by the parameter of this command for left direction from the right edge of analysis range which is set by the ANARANGE command. For more information, see Appendix D. (Data format: stimulus) (Query only)

Parameter Range	-5.0×10^5 to 5.0×10^5
Query Response	{*value*}
Examples	`OUTPUT @E5100;"TARL? -10"` `ENTER @E5100;A`

TARR?␣<*value*>

Outputs stimulus of the first found point which has value specified by parameter of this command for right direction from left edge of analysis range which is set by ANARANG command. For more information, refer to Appendix D. (Query only)

Parameter Range	-5.0×10^5 to 5.0×10^5
Query Response	{*value*}
Examples	`OUTPUT @E5100;"TARR? -10"` `ENTER @E5100;A`

TARSUBL?␣<*value1*>[,<*value2*>]

On LOG MAG&PHASE, search the response value of subtrace which this command enmarks, and outputs the relating stimulus value and main trace's response value. This command searchs <*value2*> times from right end to left word; the analisys range are defined with ANARANG command. If E5100A/B can not get the target, E5100A/B outputs 0. (Query only)

Parameter Range	{*value1*} {*value2*} {*value1*} : Target value of the sub trace {*value2*} : Number of search (omittable, default=1)
Query Response	{*value1*}, {*value2*}[,{*value3*}, {*value4*} . . .] {*value1,3...*} : Stimulus value {*value2,4...*} : Response value of the main trace
Examples	`OUTPUT @E5100;"TARSUBL?"` `ENTER @E5100;A`

TARSUBR?␣<*value1*>[,<*value2*>]

On LOG MAG&PHASE, search the response value of subtrace which this command enmarks, and outputs the relating stimulus value and main trace's response value. This command searchs <*value2*> times from right end to left word; the analisys range are defined with ANARANG command. If E5100A/B can not get the target, E5100A/B outputs 0. (Query only)

Parameter Range	{*value1*} {*value2*} {*value1*} : Target value of the sub trace {*value2*} : Number of search (omittable, default=1)
Query Response	{*value1*}, {*value2*}[,{*value3*}, {*value4*} . . .] {*value1,3...*} : Stimulus value {*value2,4...*} : Response value of the main trace
Examples	`OUTPUT @E5100;"TARSUBR? -10"` `ENTER @E5100;A`

THRR␣<*value*>

Specifies threshold height of peak for waveform analysis command. Waveform analysis commands ignore ripples which has less height than specified value.

Query Response	{*value*}
Examples	`OUTPUT @E5100;"THRR -50"`

TIMO␣{OFF|ON|0|1}

Sets the time limit for the trapping on/off. (Option 022 only)

Parameter Description	`ON` or `1` : set the time limit for the trapping ON `OFF` or `0` : set the time limit for the trapping OFF
Query Response	{1|0}

Examples	`OUTPUT @E5100;"TIMO ON"`
	
	`OUTPUT @E5100;"TIMO?"`
	`ENTER @E5100;A`

TITL␣<*string*>

Sends the string to the title area on the display.

Equivalent Key Sequence	(Display) `TITLE`
Parameter Range	up to 68 characters
Query Response	{*string*}

Examples	`OUTPUT @E5100;"TITL ""COMMENT"""`
	
	`OUTPUT @E5100;"TITL?"`
	`ENTER @E5100;A$`

TOPV␣<*value*>

Sets the value at the top line of the graticule.

Equivalent Key Sequence	(Display) `TOP VALUE`
Parameter Range	-10^9 to 10^9
Query Response	{*value*}

Examples	`OUTPUT @E5100;"TOPV 100"`
	
	`OUTPUT @E5100;"TOPV?"`
	`ENTER @E5100;A`

TOTIME␣<value>

Sets the limit time for the trapping. (Option 022 only)

TOTIME␣<value>

Parameter Range	0 to
Query Response	{*value*} (ms)

Examples	`OUTPUT @E5100;"TOTIME 2000"`
	
	`OUTPUT @E5100;"TOTIME?"`
	`ENTER @E5100;A`

TRABGE

Sets the condition so that the measurement is advanced to the next point when a measurement value (phase value) is greater than or equal to the threshold value specified by the INPUTRAC command. (Option 022 only)

Query Response	{1|0}

Examples	`OUTPUT @E5100;"TRABGE"`
	
	`OUTPUT @E5100;"TRABGE?"`
	`ENTER @E5100;A`

TRABLE

Sets the condition so that the measurement is advanced to the next point when a measurement value (phase value) is less than or equal to the threshold value specified by the INPUTRAC command. (Option 022 only)

Query Response	{1|0}

Examples	`OUTPUT @E5100;"TRABLE"`
	
	`OUTPUT @E5100;"TRABLE?"`
	`ENTER @E5100;A`

TRACK␣{OFF|ON|0|1}

Tracks the search at the specified target value with each new sweep.

Equivalent Key Sequence	(Marker) MKR SEARCH [ ] TRACKING ON OFF
Parameter Description	OFF or 0 : not tracks the search at the specified target value ON or 1 : tracks the search at the specified target value
Query Response	{0|1}

Examples	`OUTPUT @E5100;"TRACK ON"`
	
	`OUTPUT @E5100;"TRACK?"`
	`ENTER @E5100;A`

TRAFDATA

Sets the trap function on the data trace. (Option 022 only)

Query Response	{0|1}

Examples	`OUTPUT @E5100;"TRAFDATA"`
	
	`OUTPUT @E5100;"TRAFDATA?"`
	`ENTER @E5100;A`

TRAFMEMO

Sets the trap function on the sub trace. (Option 022 only)

Query Response	{0|1}

Examples	`OUTPUT @E5100;"TRAFMEMO"`
	
	`OUTPUT @E5100;"TRAFMEMO?"`
	`ENTER @E5100;A`

TRAP␣{OFF|ON|0|1}

Sets the trap function on/off. (Option 022 only)

Parameter Description	`OFF` or `0` : Set the trap function off. `ON` or `1` : Set the trap function on.
Query Response	{0|1}

Examples	`OUTPUT @E5100;"TRAP ON"`
	
	`OUTPUT @E5100;"TRAP?"`
	`ENTER @E5100;A`

TRAR␣*<value1>*,*<value2>*

Sets the start and the end points for the partial sweep for the trap function.

If the start point and the end point are not between 1 to the value specified by `Number of Point`, E5100A/B option 022 will perform a sweep within the possible range. Use this function to reserve wider sweep range so that you can change the sweep range by simply changing this setup. This way, you can save more measurement time than changing the whole sweep setup. (Option 022 only)

TRAR␣<*value1*>**,**<*value2*>

Parameter Description	<*value1*> : The start point for the partial sweep <*value2*> : The end point for the partial sweep
Query Response	{*value1*}{*value2*}

Examples	`OUTPUT @E5100;"TRAR 10, 20"`
	
	`OUTPUT @E5100;"TRAR?"`
	`ENTER @E5100;A,B`

TRIGMEAS

Triggers DSP to start measurement and get measurement data into DSP. This command is only used for the parallel processing using with `MOVADARY` and `ADTOTRAC` command. (No Query)

TRIM␣{CONT|HOLD|SING}

Selects the trigger mode.

Equivalent Key Sequence	(Trigger) `HOLD` , `SINGLE`, `CONTINUOUS`
Parameter Description	`CONT` : continuous sweep `HOLD` : hold `SING` : single sweep
Query Response	{CONT|HOLD}SING}

Examples	`OUTPUT @E5100;"TRIM SING"`
	
	`OUTPUT @E5100;"TRIM?"`
	`ENTER @E5100;A$`

UPDD␣{OFF|ON|0|1}

Sets the refresh of the display on or off.When `UPDD` is turned `OFF`, the operating speed to measure or setup will be faster.It is recommended to use this command with `ALL BASIC` to avoid to make a misreading because the status display on the LCD may not coincide with a current status when `UPDD` is turned `OFF`.This command is not effect to the limit table.

Parameter Description	`OFF` or 0 : set the refresh OFF `ON` or 1 : set the refresh ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"UPDD ON"`
	
	`OUTPUT @E5100;"UPDD?"`
	`ENTER @E5100;A`

UPPELIMI␣<*value1*><*value2*><*value3*> ... <*valuen*>

Sets the upper limit values of the limit line. The upper value can be set at each measurement point.

Parameter Description	n = Number of points <value> : Upper value of limit line
Query Response	{*value1*} {*value2*} {*value3*} ... {*valuen*}

Examples	`DIM A(1:201)` *NOP: 201*
	Set upper value of limit line
	`OUTPUT @E5100;"UPPELIMI";A(*)`
	
	`OUTPUT @E5100;"UPPELIMI?"`
	`ENTER @E5100;A(*)`

WIDT␣{ON|OFF|0|1}

Sets the bandwidth search feature (ON) or not (OFF).

Equivalent Key Sequence	(Marker) `UTILITY MENU` `WIDTHS ON OFF`
Parameter Description	`OFF` or 0 : Bandwidth search OFF `ON` or 1 : Bandwidth search ON (display center stimulus value, bandwidth, Q, insertion loss, frequency difference between center and cut off points)
Query Response	{1|0}

Examples	`OUTPUT @E5100;"WIDT ON"`
	
	`OUTPUT @E5100;"WIDT?"`
	`ENTER @E5100;A`

WIDV␣<*value*>

Sets the amplitude parameter that defines the start and stop points for a bandwidth search.

Equivalent Key Sequence	(Marker) `UTILITY MENU` `WIDTH VALUE`
Query Response	{*value*}

Examples	`OUTPUT @E5100;"WIDV 0"`
	
	`OUTPUT @E5100;"WIDV?"`
	`ENTER @E5100;A`

WIDV␣<*value*>

WOPEN␣<*string*>

If the specified file exists, this command makes it write-enabled; otherwise, creates a new file and makes it write-enabled. Note that acceptable file format for this command is the DOS format.

The format and size of an existing file cannot be changed. Therefore, if you want to change them, delete the file itself using the `PURG` command and then create a new file using this command.

This command is used in combination with the `WRITE` command and the `CLOSE` commands, as shown in Figure J-1. (No query)

Parameter	Description
<*string*>	File name of up to 12 characters including its extention

WRITE␣<*block*>

Writes data in a file that has been write-enabled using the `WOPEN` command. Written data must take the fixed length block format (see Figure J-2) defined in IEEE488.2. The maximum length of data is 16 Kbytes. If data is greater than 16 Kbytes, execute this command repeatedly to write it. (No query)

Generally, this command is used in combination with the `WOPEN` command and the `CLOSE` command, as shown in Figure J-1. (No query)

Parameter	Description
<*block*>	Data in the fixed length block format

WRIT16

Set the bit width of the data outputted to the I/O port while performing the trap function to 16 bit. The port F is used. (Option 022 only)

Query Response	{0|1}

Examples	`OUTPUT @E5100;"WRIT16"`
	
	`OUTPUT @E5100;"WRIT16?"`
	`ENTER @E5100;A`

WRIT24

Set the bit width of the data outputted to the I/O port while performing the trap function to 24 bit. The port H is used. (Option 022 only)

Query Response	{0|1}

Examples	`OUTPUT @E5100;"WRIT24"`
	
	`OUTPUT @E5100;"WRIT24?"`
	`ENTER @E5100;A`

SCPI Command (PROGram sub-system command)

:PROGram:CATalog?

Returns all the defined program names. The program name is always "PROG", because the analyzer's HP Instrument BASIC only executes a single program at a time. This command can be used from an external controller only. (Query only)

Query Response	{"PROG"}
Examples	`OUTPUT @E5100;":PROG:CAT?"` `ENTER @E5100;A$`

:PROGram[:SELected]:DEFine␣<*block*>

Creates and downloads programs. The `DEFine` query uploads programs. This command can be used from an external controller only.

:PROGram[:SELected]:DEFine␣<*block*>

Equivalent Key Sequence	
Parameter Description	<*block*> : block data of program The <*block*> must be arbitrary block program data containing the lines of program code. The first line of <*block*> must be a header, which shows the program size. There are two formats for the header as follows: #0 : Allows the OUTPUT statement to send program line until END is specified in the OUTPUT statement. * #NMM. . . . M : Specifies the program size. N specifies the number of digits that define the program size M. . . . M is program size in byte (N digits) Each line of the program must be separated by <CR> or <CR> <LF>. When the size of the <*block*> exceeds the amount of available memory in the instrument, the program lines are saved up to the point of memory overflow. In the response to the DEFine query, the selected program and its size are returned. The selected program must be in either the paused or stopped state for the program to be uploaded. The <*block*> is uploaded as definite length arbitrary block response data. The program size is returned in the first line as the header, then program lines are returned.
Query Response	{*block*}

Examples

```
OUTPUT @E5100;":PROG:DEF #0"
OUTPUT @E5100;"10 PRINT ""HELLO!"""
OUTPUT @E5100;"20 END"
OUTPUT @E5100;" " END


DIM A$[100000]
OUTPUT @E5100;":PROG:DEF?"
ENTER @E5100 USING "%,2A";HEAD$  !    Gets the header.
B=VAL(HEAD$[2])                  !
FOR I=1 TO B                     !
  ENTER @E5100 USING "%,A";HEAD$ !
NEXT I                           !
ENTER @E5100 USING "-K";A$       !    Gets the program.
```

:PROGram[:SELected]:DELete[:SELected]

Deletes the program in the BASIC editor of the analyzer. This command can be used from an external controller only. (No query)

Examples

```
OUTPUT @E5100;":PROG:DEL"
```

:PROGram[:SELected]:DELete:ALL

Deletes the program in the BASIC editor of the analyzer. This command can be used from an external controller only. (No query)

Examples	`OUTPUT @E5100;":PROG:DEL:ALL"`

:PROGram[:SELected]:EXECute␣<*string*>

Executes the program command. The program must be in either paused or stopped before the `EXECute` command is allowed. This command can be used from an external controller only. (No query)

Parameter Description	<*string*> :Legal program command
Examples	`OUTPUT @E5100;":PROG:EXEC ""STEP"""`

:PROGram[:SELected]:NUMBer␣<*string*>,<*numeric (1)*>[,<*numeric (2)*>[, ... [,<*numeric (n)*>]

Sets or queries the contents of numeric program variables and arrays in the program on the BASIC editor of the analyzer. This command can be used from an external controller only.

Parameter Description	<*string*> : Name of an existing variable in the selected program (either character data or string data) <*numeric*> : Variable value
Query Response	{*numeric (1)*} [{*numeric (2)*} [... [{*numeric (n)*}]
Examples	`OUTPUT @E5100;":PROG:NUMB A,1"`

:PROGram[:SELected]:STATe␣{RUN|PAUSe|STOP|CONTinue}

Sets or queries the state of the program in the BASIC editor of the analyzer. The table below defines the affect of setting the state to the specified state from each of the possible current states. This command can be used from an external controller only.

Desired State	Current State		
	RUN	**PAUSE**	**STOP**
RUN	error (−221)	RUN	RUN
CONT	error (−221)	RUN	error (−221)
PAUSE	PAUSE	PAUSE	STOP
STOP	STOP	STOP	STOP

:PROGram[:SELected]:STATe␣{RUN|PAUSe|STOP|CONTinue}

Query Response	{"RUN"|"PAUS"|"STOP"}

Examples	`OUTPUT @E5100;":PROG:STAT ""STOP"""`
	
	 `OUTPUT @E5100;":PROG:STAT?"`
	 `ENTER @E5100;A$`

:PROGram[:SELected]:STRing␣<*string (varname)*>,<*string (value1)*>[,<*string (value2)*>[, ... [,<*string (value n)*>

Sets or queries the contents of string program variables and arrays in the program in the BASIC editor of the analyzer. If a string value is too long it is truncated when stored in the program's variable. This command can be used from an external controller only.

Parameter Description	<*string(varname)*> : Name of an existing variable in the selected program (either character data or string data). <*string(value)*> : Variable value
Query Response	{*string (1)*} [{*string (2)*} [... [{*string (n)*]

Examples	`OUTPUT @E5100;"PROG:STR ""A$"", ""TEST"""`
	
	 `OUTPUT @E5100;":PROG:STR? ""A$"""`
	 `ENTER @E5100;B$`
	
	 `OUTPUT @E5100;":PROG:STR? 'A$'"`
	 `ENTER @E5100;B$`

:PROGram[:SELected]:WAIT

Causes no further commands or queries to be executed until the defined program exits from the RUN state. That is, the program is either stopped or paused. This command can be used from an external controller only.

Query Response	{1} (1 is returned when the program is either stopped or paused.)

Examples	`OUTPUT @E5100;":PROG:WAIT"`
	
	 `OUTPUT @E5100;":PROG:WAIT?"`
	 `ENTER @E5100;A`

A

Manual Changes

Introduction

This appendix contains the information required to adapt this manual to earlier versions or configurations of the analyzer than the current printing date of this manual. The information in this manual applies directly to the E5100A/B Network Analyzer serial number prefix listed on the title page of this manual.

Manual Changes

To adapt this manual to your E5100A/B, see Table A-1 and Table A-2, and make all the manual changes listed opposite your instrument's serial number and firmware version.

Instruments manufactured after the printing of this manual may be different from those documented in this manual. Later instrument versions will be documented in a manual changes supplement that will accompany the manual shipped wih that instrument. If your instrument's serial number is not listed on the title page of this manual or in Table A-1, it may be documented in a *yellow MANUAL CHANGES* supplement.

In additions to change information, the supplement may contain information for correcting errors (Errata) in the manual. To keep this manual as current and accurate as possible, Agilent Technologies recommends that you periodically request the latest *MANUAL CHANGES* supplement.

For information concerning serial number prefixes not listed on the title page or in the *MANUAL CHANGE* supplement, contact the nearest Agilent Technologies office.

Turn on the line switch or execute the `*IDN?` command by GPIB to confirm the firmware version. See the *GPIB Command Reference* manual for information on the `*IDN?` command.

Table A-1. Manual Changes by Serial Number

Serial Prefix or Number	Make Manual Changes
JP1KC	Change 1
JP2KC,JP3KC,JP4KC,JP5KC	none

Table A-2. Manual Changes by Firmware Version

Version	Make Manual Changes
Rev. 1.xx	Change 1 and 2
Rev. 2.00	Change 2
Rev. 2.13 and below	Change 3
Rev. 3.00 and below	Change 4
Rev. 3.11 and below	Change 5

Serial Number

Agilent Technologies uses a two-part, ten-character serial number that is stamped on the serial number plate (see Figure A-1) attached to the rear panel. The first five characters are the serial prefix and the last five digits are the suffix.

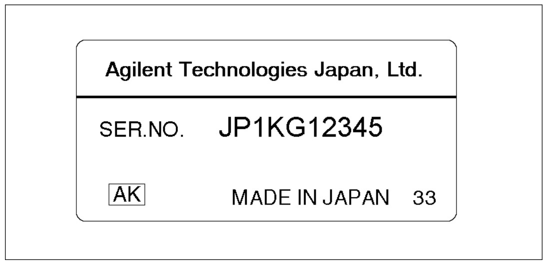


Figure A-1. Serial Number Plate (Sample)

Change 1

The firmware revision 1.xx does not support the following commands. Please delete the descriptions about these commands in this manual.

ANARANGP

CALCOPY

CIVAL

CLEMNU3

EQUCPARA

GRAPFORM

INPUSTIM

MARKTIME

MENU3

OUTPCF2

OUTPXF2

PICIRC

PRIR

POWU

RPLMM

SAVDGRAP

SAVDMNU3

Change 2

The firmware revision 1.xx and 2.00 do not support the following commands. Please delete the descriptions about these commands in this manual.

COPYRIGHT

MAXPOIN

MAXPORT

PARSMODE

PARSRANG

Change 3

The firmware revision 2.13 and below do not support the following commands. Please delete the descriptions about these commands in this manual.

ANAPOINS?

ANASTIMP?

CORRS?

DATAM

DATAMN

GRAPCOL

INPUDATM

LMAXS?

LMINS?

NEXTNPK?

NPEAK

NPEAKLIST?

OUTPDATM

SET1PT

TARSUBL?

TARSUBR?

The firmware revision 2.13 and below, please delete these chapter in this manual.

Appendix J New functions of Firmware Revision 3

Change 4

The firmware revision 3.00 and below do not support the following commands. Please delete the descriptions about these commands in this manual.

NPEAKSORT?

PEAKSORT?

Change 5

The firmware revision 3.11 and below do not support the following commands. Please delete the descriptions about these commands in this manual.

```
CLACT
CLTGT
NOMF
SRCHR
SRCHTRFL?
```

Please delete Appendix K in this manual.

B

Command Summary

Meas/Format

Meas/Format Menu

Key Label	GPIB Command
FUNCTION []	ANAMODE
MEAS	MEAS
FORMAT	FMT
GROUP DELY APERTURE	GRODAPER
NUM of CH	NUMC
ACTIVE CH	CHAN

Function Menu

Key Label	GPIB Command
GAIN-PHASE	ANAMODE GAINP
IMPEDANCE:Refl	ANAMODE ZREFL
Trans	ANAMODE ZTRAN

Port Select Menu

Key Label	GPIB Command
A/R	MEAS AR
B/R	MEAS BR
C/R	MEAS CR
R/A	MEAS RA
B/A	MEAS BA
C/A	MEAS CA
R/B	MEAS RB
A/B	MEAS AB
C/B	MEAS CB
R	MEAS R
A	MEAS A
B	MEAS B
C	MEAS C

Gain-Phase Format Menu

Key Label	GPIB Command
LOG MAG & PHASE	FMT LOGMP
LOG MAG & DELAY	FMT LOGMD
LIN MAG & PHASE	FMT LINMP
LIN MAG & DELAY	FMT LINMD
REAL & IMAGINARY	FMT RIMAG
LOG MAG	FMT LOGM
LIN MAG	FMT LINM
PHASE	FMT PHAS
DELAY	FMT DELA
REAL	FMT REAL
IMAGINARY	FMT IMAG
EXPANDED PHASE	FMT EXPP

Z Format Menu

Key Label	GPIB Command
|Z| & PHASE z	FMT MAGZP
|Y| & PHASE y	FMT MAGYP
R-X	FMT IMPRX
G-B	FMT ADMGB
|Z|	FMT MAGZ
|Y|	FMT MAGY
PHASE z	FMT PHAZ
PHASE y	FMT PHAY
R	FMT IMPR
X	FMT IMPX
G	FMT ADMG
B	FMT ADMB
Y-AXIS []	SCAY
EXPANDED PHASE on OFF	EXPZP

(Display)

Display Menu(1/3) (2/3) (3/3)

Key Label	GPIB Command
AUTOSCALE	AUTO
MULTI CH on OFF	MULC
SPLIT DISP on OFF	SPLD
SMOOTHING on OFF	SMOO
SMOOTHING APERTURE	SMOOAPER
ELECTRICAL DELAY	ELED
PHASE OFFSET	PHAO
TITLE	TITL
STORAGE on OFF	STR
GRATICULE on OFF	DISG

Linear Scale Menu

Key Label	GPIB Command
SCALE/DIV	SCAL
REFERENCE POSITION	REFP
REFERENCE VALUE	REFV
MARKER -> REFERENCE	MARKREF
ACTIVE TRC []	ATRC

Log Scale Menu

Key Label	GPIB Command
TOP VALUE	TOPV
BOTTOM VALUE	BOTV
ACTIVE TRC []	ATRC

Define Trace Menu

Key Label	GPIB Command
TRACE: DATA	DISPDATA
MEMORY	DISPMEMO
DATA and MEMORY	DISPDATM
DATA-MEM	DISPDMM
DATA/MEM	DISPDDM
DATA→MEMORY	DATI

Basic Allocation Menu

Key Label	GPIB Command
ALL INSTRUMENT	DISAALLI
HALF INSTR HALF BASIC	DISAHIHB
ALL BASIC	DISAALLB
BASIC STATUS	DISABASS

Cal

Gain-Phase CAL Menu

Key Label	GPIB Command
CORRECTION on OFF	CORR
CALIBRATE: NONE	CALI NONE
RESPONSE	CALI RESP
RESPONSE & ISOL'N	CALI RAI
1-PORT 3-TERM	CALI ONE
SET ZO	SETZ

Thru CAL Menu

Key Label	GPIB Command
THRU	STANC
DONE:	RESPDONE

Response & Isolation CAL Menu

Key Label	GPIB Command
RESPONSE	RAIRESP
ISOL'N STD	RAIISOL
DONE:	RAID

Gain-Phase 3 Term CAL Menu

Key Label	GPIB Command
OPEN	CALSS11A
SHORT	CLASS11B
LOAD	CLASS11C
DONE:	SAV1

Gain-Phase CAL STD value menu

Key Label	GPIB Command
OPEN STD	CALKO{RS|LS|CP}
SHORT STD	CALKS{RS|LS|CP}
LOAD STD	CALKL{RS|LS|CP}

Z:Refl CAL Menu

Key Label	GPIB Command
CORRECTION on OFF	CORR
CALIBRATE: NONE	CALI NONE
1-PORT 3-TERM	CALI ONP
SET Z0	SETZ

Z: Refl CAL Menu

Key Label	GPIB Command
OPEN	CLASS11A
SHORT	CLASS11B
LOAD	CLASS11C
DONE:	SAV1

Z: Refl CAL STD value menu

Key Label	GPIB Command
OPEN STD	CALKO{RS|LS|CP}
SHORT STD	CALKS{RS|LS|CP}
LOAD STD	CALKL{RS|LS|CP}

Z: Trans CAL Menu

Key Label	GPIB Command
CORRECTION on OFF	CORR
CALIBRATE: NONE	CALI NONE
3 TERM	CALI ONEP
1 TERM	CALI RESP
SET Z0	SETZ

Z:Trans 3 Term CAL Menu

Key Label	GPIB Command
OPEN	CALSS11A
SHORT	CALSS11B
LOAD	CALSS11C
DONE:	SAV1

1 Term CAL Menu

Key Label	GPIB Command
THRU	STANC
DONE:	RESPDONE

Z: Trans CAL STD value menu

Key Label	GPIB Command
OPEN STD	CALKO{RS|LS|CP}
SHORT STD	CALKS{RS|LS|CP}
LOAD STD	CALKL{RS|LS|CP}

Marker

Marker Menu

Key Label	GPIB Command
ACTIVE MARKER	MARK{1|4}
CLEAR MARKER	CLEM{1|4}

Marker Serch Menu

Key Label	GPIB Command
SEARCH:MAX	SEAMAX
MIN	SEAMIN
TARGET	SEATARG
TRACKING on OFF	TRACK

Serch Range Menu

Key Label	GPIB Command
SEARCH RNG STORE	SEARSTOR
PART SRCH on OFF	PARS

Marker Utility Menu

Key Label	GPIB Command
STATISTICS on OFF	MEASTAT
WIDTHS on OFF	WIDT
WIDTH VALUE	WIDV
MKR LIST on OFF	MARKL

ΔMode Menu

Key Label	GPIB Command
Δ REF MARKER	DELR{1|4}
ΔREF=Δ FIXED MKR	DELRFIXM
Δ MODE OFF	DELO
MKR ZERO	MARKZERO

Fixed Marker Position Menu

Key Label	GPIB Command
FIXED MKR STIMULUS	MARKFSTI
FIXED MKR VALUE	MARKFVAL

Marker Mode Menu

Key Label	GPIB Command
MARKERS: DISCRETE	MARKDISC
CONTINUOUS	MARKCONT
MARKERS: COUPLED	MARKCOUP
UNCOUPLED	MARKUNCO
MKR TIME on OFF	MARKTIME

Sweep

Sweep Menu

Key Label	GPIB Command
SWEEP TYPE MENU	SWPT
SWEEP TIME	SWET
NUMBER of POINTS	POIN
POWER	POWE
CW FREQ	CWFREQ
IF BW	IFBW
COUPLED CH on OFF	COUC
ACTIVE CH [CH1]	CHAN

Sweep Type Menu

Key Label	GPIB Command
LIN FREQ	SWPT LINF
POWER SWEEP	SWPT POWE
LIST FREQ	SWPT POWE
LIST NO. []	LISSLIS{1|2}
LIN FREQ [STEP]	SWPT RAMPF
LIN FREQ [RAMP]	SWPT LINF
SWEEP DIR []	SWED
LIST DISP: FREQ BASE	LISDFBASE
ORDER BASE	LISDOBASE

Sweep Time Menu

Key Label	GPIB Command
SWEEP TIME AUTO	SWETAUTO

Trigger

Trigger Menu

Key Label	GPIB Command
HOLD	TRIM HOLD
SINGLE	TRIM SING
CONTINUOUS	TRIM CONT
TRIG EVENT []	EXTT{ON|OFF}
MEASURE RESTART	REST

Start Stop Center Span

Key Label	GPIB Command
Start	STAR
Stop	STOP
Center	CENT
Span	SPAN

→ Function Menu

Key Label	GPIB Command
MKR→START	MARKSTAR
MKR→STOP	MARKSTOP
MKR→CENTER	MARKCENT
MKR→SPAN	MARKSPAN
MKR→REFRENCE	MARKREF

System

System Menu(1/3) (2/3) (3/3)

Key Label	GPIB Command
PRINT	PRINALL

Clock Menu

Key Label	GPIB Command
TIME HH:MM:SS	SETCTIME
DATE MM/DD/YY	SETCDATE
DATE MODE: MonDayYear	MONDYEAR
DayMonYear	DAYMYEAR

Att Setting Menu

Key Label	GPIB Command
AUTO	ATTI{R|A|B|C}AUTO ON
0 dB	ATTI{R|A|B|C}AUTO OFF;ATTI{R|A|B|C} 0
25 dB	ATTI{R|A|B|C}AUTO OFF;ATTI{R|A|B|C} 25

Save/Recall

Save/Recall Menu

Key Label	GPIB Command
Recall	RECD
BACKUP MEMO DISK	STOMDISK
STOR DEV [ ]	STOD{DISK|MEMO}

Save Menu

Key Label	GPIB Command
ALL	SAVALL
STATE ONLY	SAVDSTA
DATA ONLY(BINARY)	SAVDDAT
DATA ONLY(ASCII)	SAVDASC
RE-SAVE FILE	RESAVD

File Utility Menu

Key Label	GPIB Command
PURGE FILE	PURG
CREATE DIRECTORY	CRED
CHANGE DIRECTORY	CHAD
COPY FILE	FILC
INITIALIZE	INID

Binary Define Save Data Menu

Key Label	GPIB Command
RAW ARRY on OFF	SAVCA
CAL ARRY on OFF	SAVRA
DATA ARRY on OFF	SAVDA
MEM ARRY on OFF	SAVMA
FORMD ARRY on OFF	SAVFA
MAIN ARRAY on OFF	SAVTA
SUB ARRAY on OFF	SAVTMA

ASCII Define Save Data Menu

Key Label	GPIB Command
MAIN ARRAY on OFF	SAVTA
SUB ARRAY on OFF	SAVTMA

MISC Save Menu

Key Label	GPIB Command
SAVE GRAPHICS	SAVDGRAP
GRAPH [ ]	GRAPFORM
SAVE MENU3	SAVDMNU3
RE-SAVE	RESAVD

Menu2

Key Label	GPIB Command
USER CI [ ]	CIVAL

Key Label	GPIB Command
SRC UNIT [ ]	POWU
PICIRCUIT ON OFF	PICIRC

C

Data I/O Format

Data Format

The E5100A/B can output data over GPIB in four different formats. The type of format affects what kind of data array is declared (real or integer), since the format determines what type of data is transferred.

Form 2

IEEE 32-bit floating point format. In this mode, each number takes 4 bytes. This means that a 201 point transfer takes 1,608 bytes. Figure C-1 shows the data transfer format of Form 2.

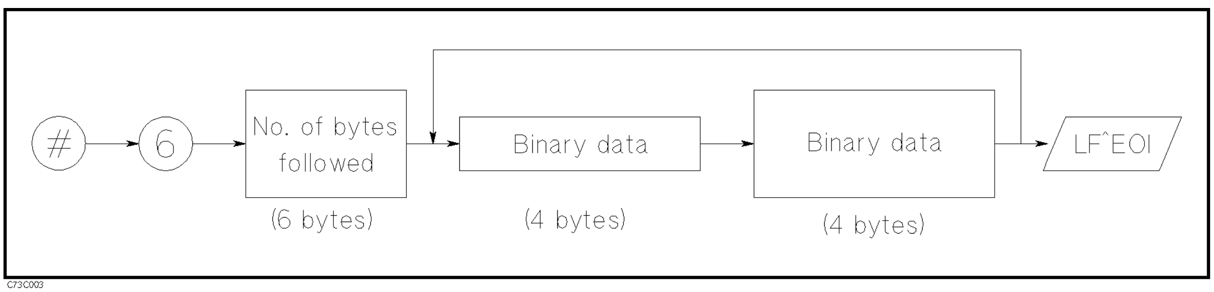


Figure C-1. Form 2 Data Transfer Format

Form 3

IEEE 64-bit floating point format. In this mode, each number takes 8 bytes. This means that a 201-point transfer takes 3,216 bytes. Data is stored internally in the 200/300 series computer with the IEEE 64-bit floating point format, eliminating the need for any reformatting by the computer. Figure C-2 shows the data transfer format of Form 3.

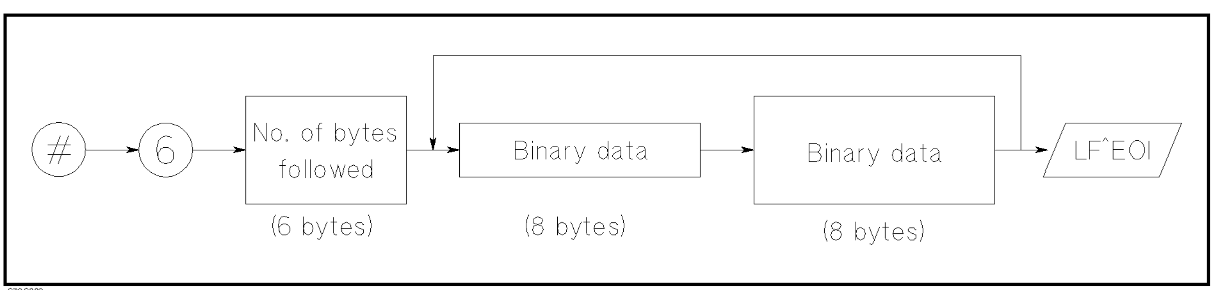


Figure C-2. Form 3 Data Transfer Format

Form 4

ASCII data transfer format. In this mode, each number is sent as a 24 character string, each character being a digit, sign, or decimal point. Use this format, when E5100A/B enters data to an internal array using `INPUxxx` command.

Form 5

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

Internal Data Array

The data is stored in data arrays, denoted by double-line boxes in Figure C-3.

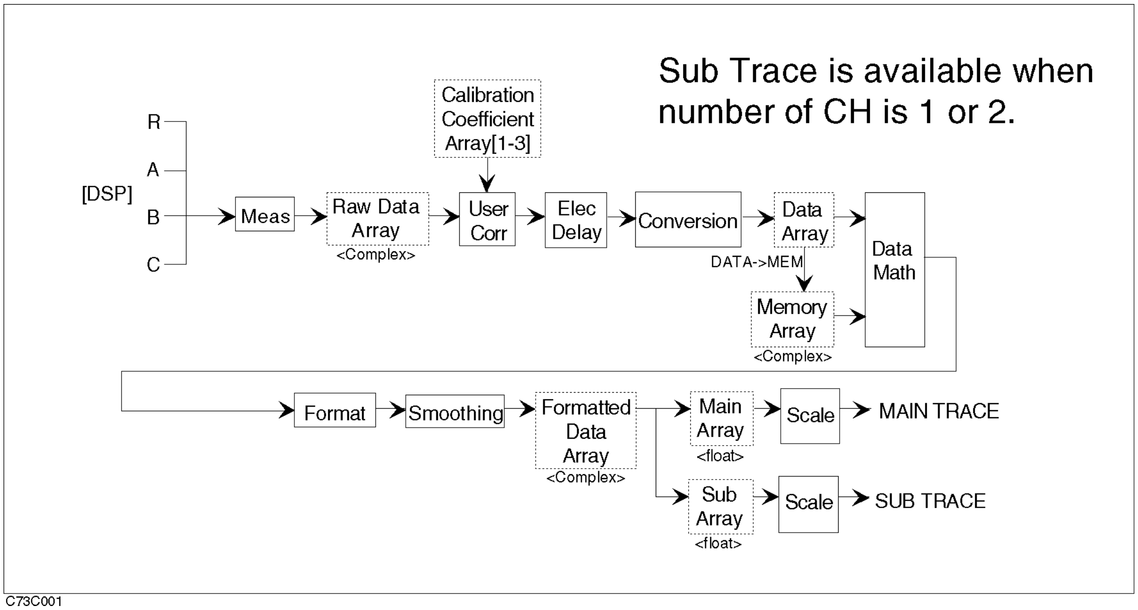


Figure C-3. Data Processing Flow Diagram

The following tables list GPIB commands to output and enter data from/to the internal array.

Table C-1. Data Array Output Commands

Array Name	Array Output	One Point Output
Raw Data Array	OUTPRAW?	-
Cal. Coef. Array	OUTPCALC{01|02|03}	-
Data Array	OUTPDATA?	-
Memory Array	OUTPMEMO?	-
Formatted Array	OUTPFORM?	OUTPFORMP?
Main Array	OUTPRFORM?	OUTPRFORMP? (OUTPDATTP?)
Sub Array	OUTPRTMEM?	OUTPRTMEMP? (OUTPMEMTP?)

Table C-2. Data Array Input Commands

Array Name	Array Input
Raw Data Array	INPURAW
Cal. Coef. Array	INPUCALC{01|02|03}
Data Array	INPUDATA
Memory Array	INPUMEMO
Formatted Array	INPUFORM
Main Array	INPURFORM
Sub Array	INPURTMEM

D

Waveform Analysis Commands

The E5100A/B has added a command set that can be used to analyze waveforms of specific devices. The waveform analysis commands analyze and output the results using only a single command. This appendix provides information about the added waveform analysis commands.

The commands are divided into five groups as follows:

- Waveform analysis setup commands
- Maximum/Minimum/Mean search commands
- Ripple analysis commands
- Filter and Resonator analysis commands
- Equivalent circuit analysis commands

Conventions and Definitions

This section describes the conventions and definitions that are used to describe the waveform analysis commands.

① →	**ANARANG**			
② →	Sets the stimulus range for the waveform. . . .			
③ →	**Syntax**	`ANARANG` *start*,*stop*		
④ →		Where,		
		0	*start*	Start value of the analysis range
		1	*stop*	Stop value of the analysis range
⑤ →	**Query Response**			
⑥ →	**Semantics**			
⑦ →	**Note**			
⑧ →	**Examples**			

<table>
<tr><td>①</td><td>Command name.</td></tr>
<tr><td>②</td><td>Command description.</td></tr>
<tr><td>③</td><td>Command syntax

This part shows the syntax of the command. You must put a space between the command and the parameters.</td></tr>
<tr><td>④</td><td>Command parameter description

The first column of the table lists the register number that is used by the <code>EXECUTE</code> command. You must put the parameter in the indicated register before using the <code>EXECUTE</code> command. For example (in the above case):

<code>WRITEIO 15,0;Start</code> Put “Start” in register 0.
<code>WRITEIO 15,1;Stop</code> Put “Stop” in register 1.
<code>EXECUTE "ANARANG"</code> Execute “ANARANG”.

The second column lists the parameter name that is shown in the Syntax area. The third column describes the parameters.</td></tr>
<tr><td>⑤</td><td>Query response.

This part shows what values will be returned as the query response. The description of the query response is similar to the description of the Syntax area shown above.</td></tr>
<tr><td>⑥</td><td>Semantics

This part describes how the command obtains the values for the query response.</td></tr>
<tr><td>⑦</td><td>Note

This part describes the required conditions or limitations when using the command.</td></tr>
<tr><td>⑧</td><td>Examples

This part shows examples of how to use the command. Examples are provided for both HP BASIC on an external controller and Instrument BASIC on the analyzer.</td></tr>
</table>

Waveform Analysis Setup Commands

The following commands are used for setting up the conditions for waveform analysis:

- ANAOCH{1|2|3|4}
- ANARANG
- ANARANGP
- ANARFULL
- ANAODATA
- ANAOMEMO
- THRR

The settings are effective for all of the waveform analysis commands.

ANAOCH{1|2|3|4}

Selects channel for waveform analysis.

Syntax ANAOCH{1|2|3|4}

Query Response *boolean*

Where,

Register	Parameter	Description
0	*boolean*	1 or 0. Channel 1 is selected (1) or is not selected (0) for waveform analysis.

Note

- The ANAOCH{1|2|3|4} channel setting is independent of the active channel setting.

ANARANG

Sets the stimulus range for waveform analysis commands by start and stop value.

Syntax `ANARANG` *start* , *stop*

Where,

Register	Parameter	Description
0	*start*	Start value of the analysis range.
1	*stop*	Stop value of the analysis range.

Query Response *start, stop*

Note

- The waveform analysis range is independent of the marker search range.
- You can set the range for each channel independently. Therefore, you need to set the analysis channel using `ANAOCH1` or `ANAOCH2` before using `ANARANG`.
- The waveform analysis range will be truncated to fit the displayed stimulus range if the setting is exceeded.
- If the displayed stimulus range is changed, the waveform analysis range is set equal to the displayed range.
- Store the waveform analysis range setting using (SAVE) ALL or STATE ONLY .
- The waveform analysis range is set to equal to the displayed stimulus range when the power is turned on.

Examples

```
INPUT "Enter Start for Analysis Range.",Start
INPUT "Enter Stop for Analysis Range.",Stop
OUTPUT @E5100;"ANARANG ";Start,Stop
```

ANARANGP

Sets the waveform analysis stimulus range by entering the point number of START and point number of STOP values.

Syntax `ANARANG` *value 1* , *value 2*

Where,

Register	Parameter	Description
0	*value 1*	Start point number of the analysis range.
1	*value 2*	Stop point number of the analysis range.

Query Response *start point number, stop point number*

Note

- The power on default setting and other actions are same as that of `ANARANG`.

Examples

```
INPUT "Enter Start Point Number for Analysis Range.",Startp
INPUT "Enter Stop Point Number for Analysis Range.",Stopp
OUTPUT @E5100;"ANARANGP ";Startp,Stopp
```

ANARFULL

Sets the waveform analysis range equal to the displayed stimulus range. (No Query)

Syntax `ANARFULL`

Note

- You can set the range for each channel independently. Therefore, you need to set the analysis channel using `ANAOCH1` or `ANAOCH2` before using `ANARFULL`.
- If change stimulus after sending `ANARFULL`, the analysis range will be changed.

ANAODATA

Selects the date trace for waveform analysis.

Syntax `ANAODATA`

Query Response *boolean*

Where,

Register	Parameter	Description
0	*boolean*	1 or 0. Data trace is selected (1) or is not selected (0) for waveform analysis.

Note

- You can select the trace for each channel independently. Therefore, you need to set the analysis channel using `ANAOCH1` or `ANAOCH2` before using `ANAODATA`.

ANAOMEMO

Selects the date trace for waveform analysis.

Syntax `ANAOMEMO`

Query Response *boolean*

Where,

Register	Parameter	Description
0	*boolean*	1 or 0. Sub-trace is selected (1) or is not selected (0) for waveform analysis.

Note

- You can select the trace for each channel independently. Therefore, you need to set the analysis channel using `ANAOCH1` or `ANAOCH2` before using `ANAOMEMO`.

THRR

Sets threshold ripple height for waveform analysis commands.

Syntax THRR *height*

Where,

Register	Parameter	Description
0	*height*	(Peak height) - (negative peak height)

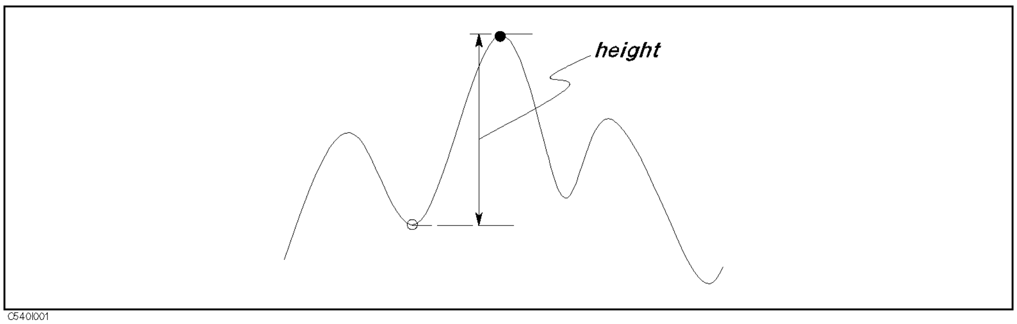


Figure D-1. THRR

Query Response *height*

Semantics

- Ripple height is defined as the difference between the positive peak and the negative peak.
- Waveform analysis commands search only for ripples greater than the threshold value, any others are ignored.

Note

- Default threshold value is 0.

Examples

```
INPUT "Enter Pos. Peak Gain [dB].",Local_max
INPUT "Enter Neg. Peak Gain [dB].",Local_min
Height=Local_max-Local_min
OUTPUT @E5100;"THRR ";Height
```

Maximum/Minimum/Mean Value Search Commands

The following commands return the maximum, minimum, and mean value of a trace within the range specified using the ANARANG command.

- OUTPMAX?
- OUTPMIN?
- OUTPMINMAX?
- OUTPMEAN?
- PEAK?
- NEXPK?
- NUMLMAX?
- NUMLMIN?
- LMAX?
- LMIN?
- TARR?
- TARL?

OUTPMAX?

Returns the maximum point value and its stimulus within the specified range. (Query only)

Syntax OUTPMAX?

Query Response MAX, f_{max}

Where,

Register	Parameter	Description
0	MAX	Maximum value
1	f_{max}	Stimulus at maximum point (Frequency or Power)

Examples

```
OUTPUT @E5100;"OUTPMAX?"
ENTER @E5100;Max_value,F_max
PRINT Max_value,F_max
```

OUTPMIN?

Returns the minimum point value and its stimulus within the specified range. (Query only)

Syntax OUTPMIN?

Query Response MIN, f_{min}

Where,

Register	Parameter	Description
0	MIN	Minimum value
1	f_{min}	Stimulus at minimum point (Frequency or Power)

OUTPMINMAX?

Returns the maximum and minimum values and their stimulus values within the specified range. (Query only)

Syntax `OUTPMINMAX?`

Query Response *MIN*, f_{min}, *MAX*, f_{max}

Where,

Register	Parameter	Description
0	*MIN*	Minimum value
1	f_{min}	Stimulus at minimum point (Frequency or Power)
2	*MAX*	Maximum value
3	f_{max}	Stimulus at maximum point (Frequency or Power)

Examples

```
OUTPUT @E5100;"OUTPMINMAX?"
ENTER @E5100;Min_value,F_min,Max_value,F_max
PRINT "MIN:",Min_value,F_min
PRINT "MAN:",Max_value,F_max
```

OUTPMEAN?

Returns the mean value within the specified range. (Query only)

Syntax `OUTPMEAN?`

Query Response *mean*

Where,

Register	Parameter	Description
0	*mean*	Mean value.

Examples

```
OUTPUT @E5100;"OUTPMEAN?"
ENTER @E5100;Mean
PRINT Mean
```

PEAK?

Returns maximum peak and its stimulus within the specified range. (Query only)

Syntax PEAK?

Query Response MAX_{peak}, $f_{maxpeak}$

Where,

Register	Parameter	Description
0	MAX_{peak}	Maximum peak value
1	$f_{maxpeak}$	Stimulus at maximum peak

Semantics

- The analyzer defines the searched value and point as a reference point for the next NEXPK? command. The reference point is stored using (SAVE) ALL or STATE ONLY.

Note

- If the search fails, the analyzer returns 0,0.

Examples

```
OUTPUT @E5100;"PEAK?"
ENTER @E5100;Peak,F_maxpeak
PRINT "Peak:",Peak,"[dB],",F_maxpeak,"[Hz]"
```

NEXPK?

Returns the maximum peak having a value less than the value that was found using last PEAK? or NEXPK? command within the specified range. It also returns the corresponding stimulus value. (Query only)

Syntax NEXPK?

Query Response $Peak$, f_{Peak}

Where,

Register	Parameter	Description
0	$Peak$	Searched peak value
1	f_{Peak}	Searched stimulus

Note

- The analyzer defines the searched value and point as a reference point for the next NEXPK? command. The reference point is stored using (SAVE) ALL or STATE ONLY.
- If the multiple corresponded points are found, the analyzer returns right-hand nearest peak of the reference point.
- If the search fails, the analyzer returns 0,0.

Examples

```
OUTPUT @E5100;"NEXPK?"
ENTER @E5100;N_peak,F_npeak
PRINT N_peak,F_npeak
```

NUMLMAX?

Returns the number of positive peaks within the specified range. (Query only)

Syntax `NUMLMAX?`

Query Response n

Where,

Register	Parameter	Description
0	n	Number of peaks

Note

- If the search fails, the analyzer returns 0.

Examples

```
OUTPUT @E5100;"NUMLMAX?"
ENTER @E5100;N
PRINT N
```

NUMLMIN?

Returns the number of negative peaks within the specified range. (Query only)

Syntax `NUMLMIN?`

Query Response n

Where,

Register	Parameter	Description
0	n	Number of negative peaks

Note

- If the search fails, the analyzer returns 0.

LMAX?

Returns the nth positive peak counted from the left end of the range.

Syntax `LMAX?` n

Where,

Register	Parameter	Description
0	n	Peak counted from the left end of the range.

Query Response $LMAX_n$

Where,

Register	Parameter	Description
0	$LMAX_n$	Value of nth peak

Note

- If the search fails, the analyzer returns `3.40282346639E+38`.

Examples

```
OUTPUT @E5100;"LMAX? 5"
ENTER @E5100;Lmax
PRINT Lmax
```

LMIN?

Returns the nth negative peak counted from the left end of the range.

Syntax `LMIN?` n

Where,

Register	Parameter	Description
0	n	Negative peak counted from the left end of the range.

Query Response $LMIN_n$

Where,

Register	Parameter	Description
0	$LMIN_n$	Value of nth negative peak

Note

- If the search fails, the analyzer returns `3.40282346639E+38`.

TARR?

Searches to the right for the point having the specified parameter-value from the left end of the range, and returns its stimulus.

Syntax `TARR?` *target*

Where,

Register	Parameter	Description
0	*target*	Search value.

Query Response f_{target}

Where,

Register	Parameter	Description
0	f_{target}	Stimulus of the first point found.

Note

- If the search fails, the analyzer returns `0`.

Examples

```
INPUT "Enter Target Value.",Target
OUTPUT @E5100;"TARR? ";Target
ENTER @E5100;F_target
PRINT F_target
```

TARL?

Searches to the left for the point having the specified parameter-value from the right end of the range, and returns its stimulus.

Syntax `TARL?` *target*

Where,

Register	Parameter	Description
0	*target*	Search value.

Query Response f_{target}

Where,

Register	Parameter	Description
0	f_{target}	Stimulus of the first point found.

Note

- If the search fails, the analyzer returns `0`.

Ripple Analysis Commands

Ripple analysis commands analyze the ripples of the waveform and return the results.

- RPLPP?
- RPLHEI?
- RPLRHEI?
- RPLLHEI?
- RPLENV?
- RPLMEA?
- RPLMM?
- RPLVAL?
- POLE?

RPLPP?

Returns the maximum difference between the positive peak and the negative peak within the specified range. (Query only)

Syntax RPLPP?

Query Response MAX_{diff}

Where,

Register	Parameter	Description
0	MAX_{diff}	Maximum difference between positive and negative peak.

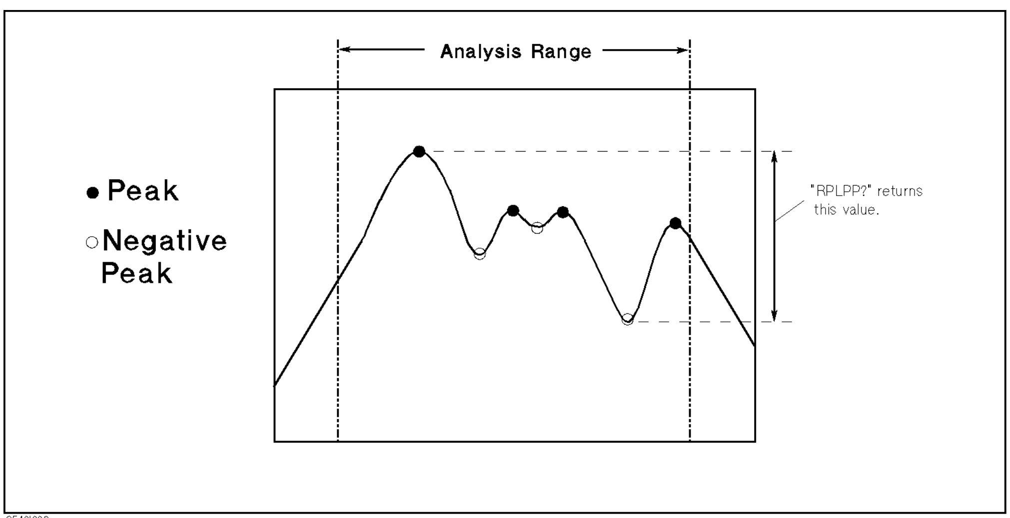


Figure D-2. RPLPP?

Note

- If the search fails, the analyzer returns 0.

Examples

```
ASSIGN @E5100 TO 717
OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
OUTPUT @E5100;"RPLPP?"
ENTER @E5100;Max_diff
PRINT Max_diff;"[dB]"
END
```

RPLHEI?

Returns the maximum difference between adjacent positive and negative peaks. (Query only)

Syntax `RPLHEI?`

Query Response *value*

Where,

Register	Parameter	Description
0	*value*	Maximum difference between adjacent positive and negative peaks.

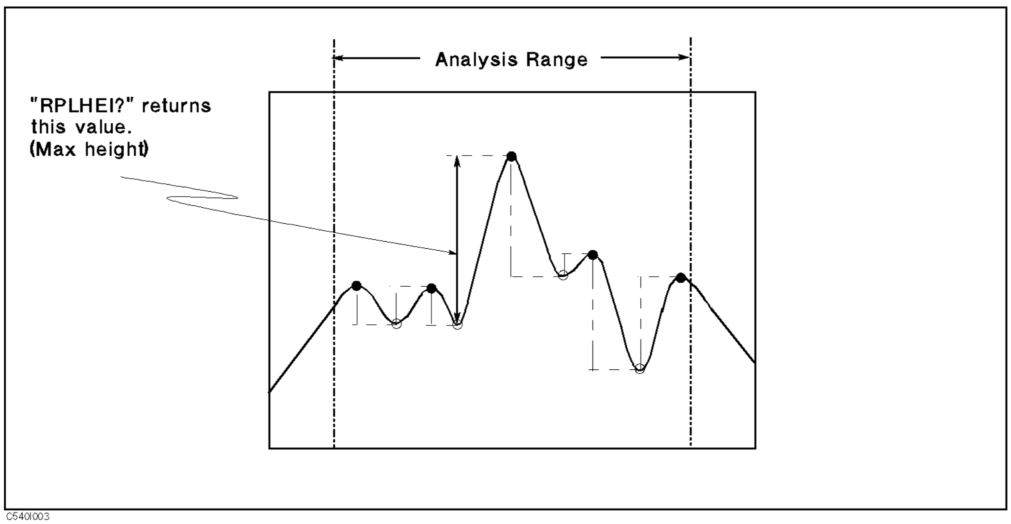


Figure D-3. RPLHEI?

Note ■ If the search fails, the analyzer returns 0.

Examples

```
ASSIGN @E5100 TO 717
OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
OUTPUT @E5100;"RPLHEI?"
ENTER @E5100;Adj_diff
PRINT Adj_diff;"[dB]"
END
```

RPLRHEI?

Returns the maximum difference between the positive peak and the right-hand adjacent negative peak. (Query only)

Syntax RPLRHEI?

Query Response *value*

Where,

Register	Parameter	Description
0	*value*	Maximum difference between the positive peak and the right-hand adjacent negative peak.

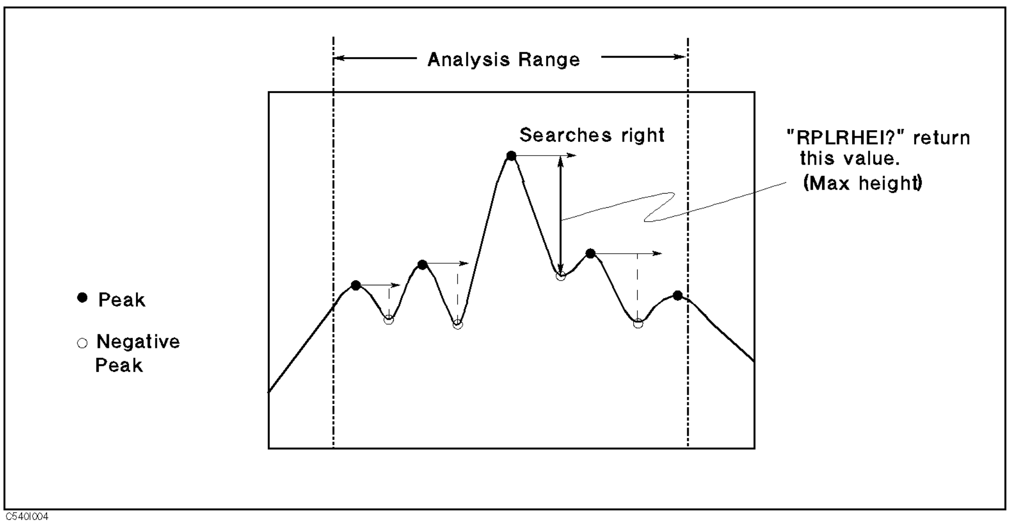


Figure D-4. RPLRHEI?

Note ■ If the search fails, the analyzer returns 0.

Examples

```
ASSIGN @E5100 TO 717
OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
OUTPUT @E5100;"RPLRHEI?"
ENTER @E5100;Adj_diff
PRINT Adj_diff;"[dB]"
END
```

RPLLHEI?

Returns the maximum difference between the positive peak and the left-hand adjacent negative peak. (Query only)

Syntax `RPLLHEI?`

Query Response *value*

Where,

Register	Parameter	Description
0	*value*	Maximum difference between the positive peak and the left-hand adjacent negative peak.

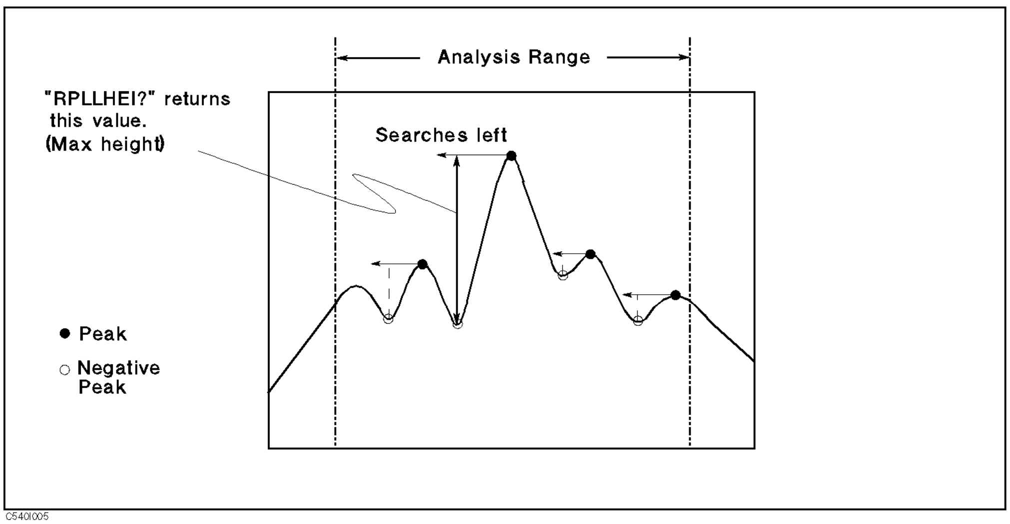


Figure D-5. RPLLHEI?

Note
- If the search fails, the analyzer returns 0.

Examples

```
ASSIGN @E5100 TO 717
OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
OUTPUT @E5100;"RPLLHEI?"
ENTER @E5100;Adj_diff
PRINT Adj_diff;"[dB]"
END
```

RPLENV?

Returns the maximum height between the negative peak and the intersection of an imaginary slope line between the adjacent positive peaks. (Query only)

Syntax `RPLENV?`

Query Response *value*

Where,

Register	Parameter	Description
0	*value*	Maximum height between the negative peak and the intersection of an imaginary slope line between the adjacent positive peaks. (See Figure D-6.)

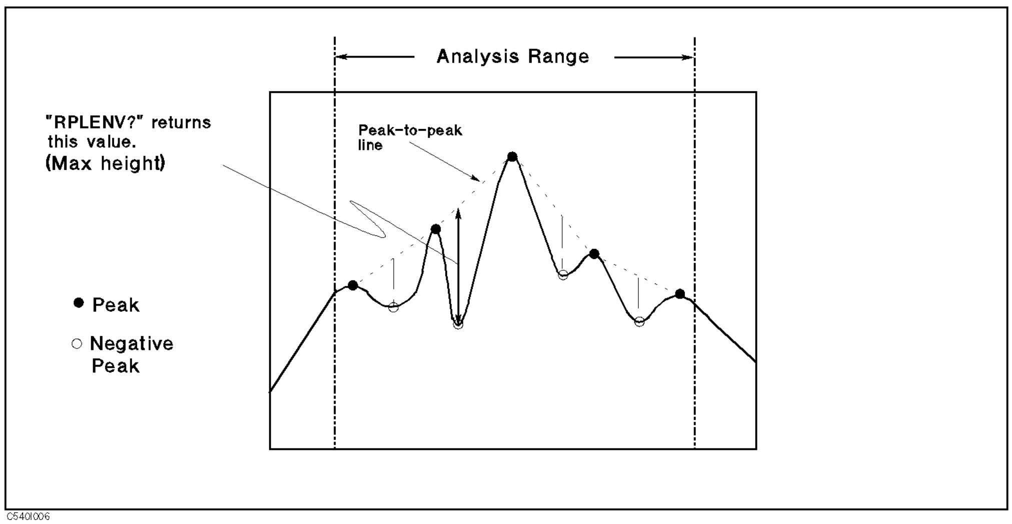


Figure D-6. RPLENV?

Note

- If the search fails, the analyzer returns 0.

Examples

```
OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
OUTPUT @E5100;"RPLENV?"
ENTER @E5100;Env_diff
PRINT Env_diff;"[dB]"
END
```

RPLMEA?

Returns the mean of the difference between the adjacent positive and negative peaks within the specified range. (Query only)

Syntax `RPLMEA?`

Query Response *value*

Where,

Register	Parameter	Description
0	*value*	Mean of the difference between the adjacent positive and negative peaks. (See Figure D-7)

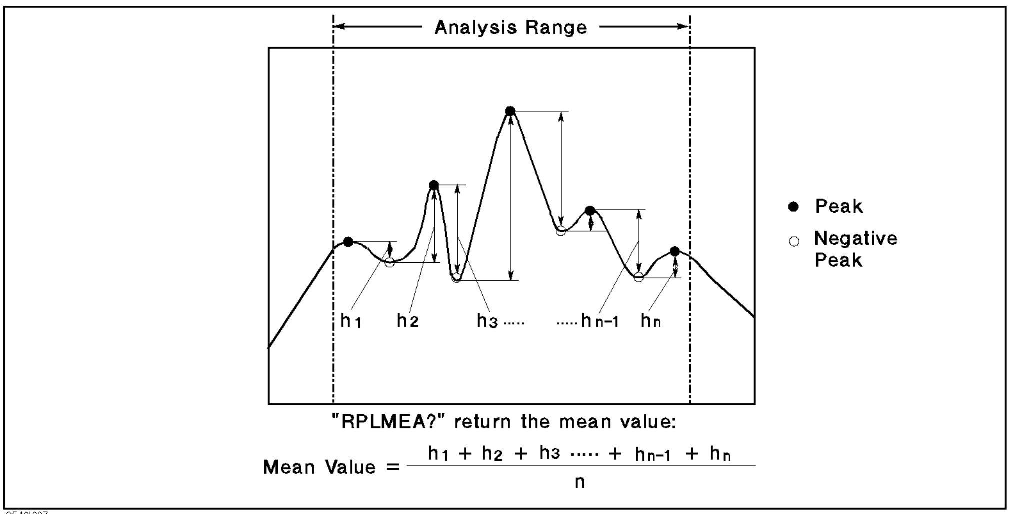


Figure D-7. RPLMEA?

Note ■ If the search fails, the analyzer returns `0`.

Examples

```
OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
OUTPUT @E5100;"RPLMEA?"
ENTER @E5100;Mean_diff
PRINT Mean_diff;"[dB]"
END
```

RPLMM

Outputs the difference value between the maximum and minimum values within the range specified with the ANARANG command. (The maximum and minimum values are same as ones OTUPMINMAX? outputs.) (Query only)

Syntax RPLMM?

Query Response *value*

Where,

Register	Parameter	Description
0	*value*	difference value between the maximum and minimum values

Note

- If the search fails, the analyzer returns 0.

Examples

```
OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
OUTPUT @E5100;"RPLMM?"
ENTER @E5100;Max_min
PRINT Max_min;"[dB]"
END
```

RPLVAL?

Returns the maximum total of the differences between the negative peaks and the adjacent positive peaks on both sides and the stimulus of the corresponding negative peak. (Query only)

Syntax `RPLVAL?`

Query Response Rpl_{val}, *stimulus*

Where,

Register	Parameter	Description
0	Rpl_{val}	Maximum total of the differences between the negative peaks and the adjacent positive peaks on both sides. (See Figure D-8)
1	*stimulus*	Stimulus of the corresponding negative peak

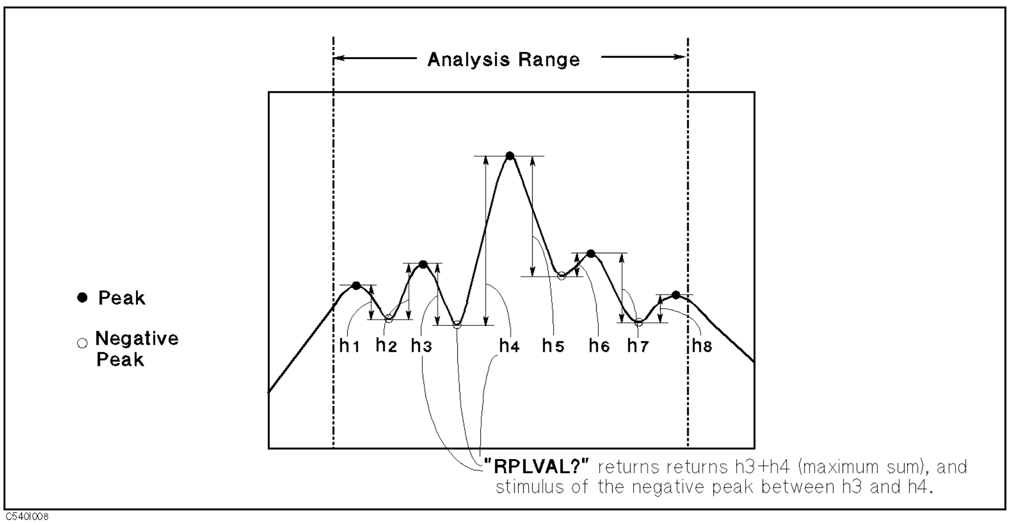


Figure D-8. RPLVAL?

Note

- If the search fails, the analyzer returns 0.

Examples

```
OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
OUTPUT @E5100;"RPLVAL?"
ENTER @E5100;Val,Stim
PRINT Val;"[dB]";Stim;"[Hz]"
END
```

POLE?

Returns the stimulus and value of the first negative peak found on each side of the maximum point that are below the specified value from the maximum peak. (Query only)

Syntax `POLE?` *D*

Where,

Register	Parameter	Description
0	*D*	Difference from the maximum peak.

Query Response x_1, $stim_1$, x_2, $stim_2$

Where,

Register	Parameter	Description
0	x_1	Left negative peak value.
1	$stim_1$	Stimulus of x_1.
2	x_2	Right negative peak value.
3	$stim_2$	Stimulus of x_2.

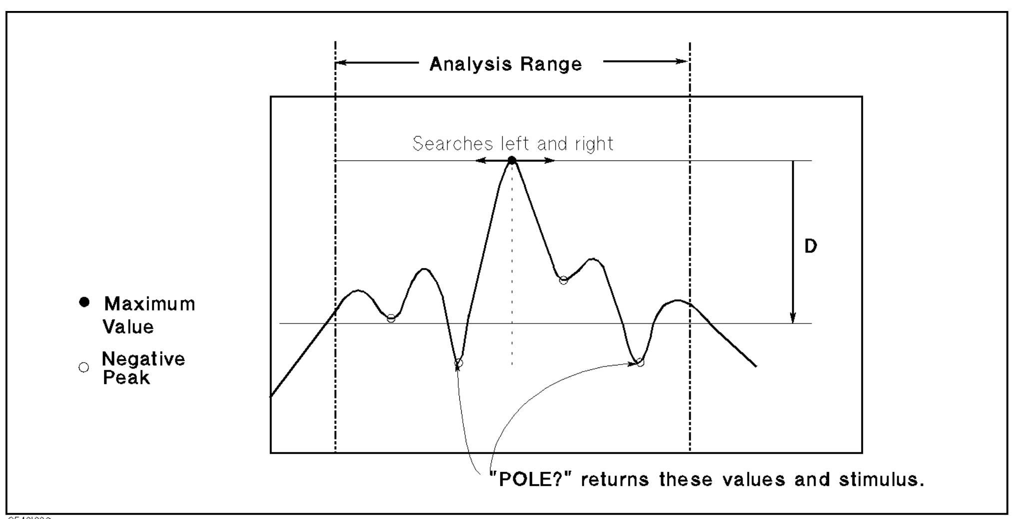


Figure D-9. POLE?

Note

- If the search fails, the analyzer returns `0`.
- Give the command parameter as a negative value. For instance, to specify 50 dB down from the maximum peak as a reference level, the parameter is `-50`.

Examples

```
ASSIGN @E5100 TO 717
OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
OUTPUT @E5100;"POLE? -50"
ENTER @E5100;X1,S1,X2,S2
PRINT "LEFT :";X1;"[dB]";S1;"[Hz]"
PRINT "RIGHT:";X2;"[dB]";S2;"[Hz]"
END
```

Filter and Resonator Analysis Commands

The following commands are device related. They are easy to use for specific device analysis because they can output many parameters using only a single command.

- OUTPFILT?
- OUTPXFIL?
- OUTPXF2?
- OUTPCFIL?
- OUTPCF2?
- OUTPRESO?
- OUTPRESR?
- OUTPRESF?
- OUTPCERR?

OUTPFILT?

Analyzes the filter and returns the parameters.

Syntax `OUTPFILT?` x

Where,

Register	Parameter	Description
0	x	The dB value down the bandwidth filter.

Query Response *Loss*, *BW*, f_{cent}, Q, Δf_{left}, Δf_{right} (Total6)

Register	Parameter	Description
0	*Loss*	Insertion loss
1	*BW*	x dB down bandwidth
2	f_{cent}	Center frequency
3	Q	Q (Quality factor)
4	Δf_{left}	Frequency difference between the left cutoff point and the middle of the range.
5	Δf_{right}	Frequency difference between the right cutoff point and the middle of the range.

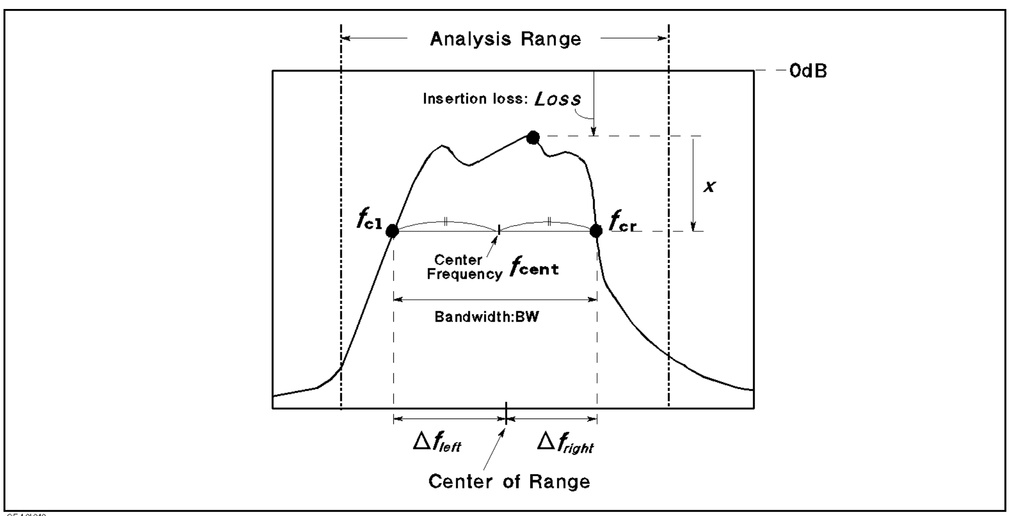


Figure D-10. OUTPFILT?

Semantics

- Insertion loss is the maximum value within the specified range.
- x dB bandwidth is the frequency difference between both of the xdB down cutoff points.
- Center frequency is the middle point of both cutoff points.
- Q is calculated using the following equation:

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

Note

- If both of the two cutoff points are not found, the analyzer returns 0 for all values of the query response.

Examples

```
10 ASSIGN @E5100 TO 717
20 CALL Sweep(1) ! Goes to the subroutine.
30 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
40 OUTPUT @E5100;"OUTPFILT? -3"
50 ENTER @E5100;Loss,Bw,Fc,Q,Dfl,Dfr
60 PRINT "Loss:";Loss;"[dB] BW:";Bw;"[Hz]"
70 PRINT "fc:";Fc;"[Hz] Q:";Q
80 PRINT "Dfl:";Dfl;"[Hz] Dfr:";Dfr;"[Hz]"
90 END
100 SUB Sweep(Ch)! Sweep End Detection Subroutine
101              ! (Parameter: No. of channel)
110   ASSIGN @E5100 TO 717
120   ON INTR 7 GOTO Sweep_end
130   OUTPUT @E5100;"TRGS BUS"
140   OUTPUT @E5100;"ESNB 2; *SRE 4"
150 FOR I=1 TO Ch
160   OUTPUT @E5100;"*CLS;*OPC?"
170   ENTER @E5100;Opc
180   ENABLE INTR 7;2
190   TRIGGER @E5100
200  Waiting:GOTO Waiting
210   Sweep_end:!
220 NEXT I
230 SUBEND
```

OUTPXFIL?

Outputs filter parameters within the range specified by the `ANARANG` command.

Syntax `OUTPXFIL?` x_1, x_2, D, f_1, f_2

Where,

Register	Parameter	Description
0	x_1	The dB value down the bandwidth filter. (1) x_1[dB]
1	x_2	The dB value down the bandwidth filter. (2) x_2[dB]
2	D	Difference from maximum value. (Same as `POLE?` parameter.)
3	f_1	Stop frequency of the range for the rejection level.
4	f_2	Start frequency of the range for the spurious level.

Query Response $Loss$, BW, f_{cent}, Q, Δf_{left1}, Δf_{right1}, Δf_{left2}, Δf_{right2}, $Pass$, $Reject$, $Spurious$, $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, $Pole_{stim2}$ (15)

Register	Parameter	Description
0	$Loss$	Insertion loss
1	BW	x_1 dB down bandwidth
2	f_{cent}	Center frequency
3	Q	Q
4	Δf_{left}	Frequency difference between the left cutoff point (f_{cl}) and the middle of the range.
5	Δf_{right}	Frequency difference between the right cutoff point (f_{cr}) and the middle of the range.
6	Δf_{left2}	Frequency difference between the left cutoff point (f_{cl2}) and the middle of the range.
7	Δf_{right2}	Frequency difference between the right cutoff point (f_{cr2}) and the middle of the range.
8	$Pass$	Passband ripple
9	$Reject$	Rejection level
10	$Spurious$	Spurious level
11	$Pole_{x1}$	First negative peak found to the left of the maximum point.
12	$Pole_{stim1}$	Stimulus of $Pole_{x1}$.
13	$Pole_{x2}$	First negative peaks found to the right of the maximum point.
14	$Pole_{stim2}$	Stimulus of $Pole_{x2}$.

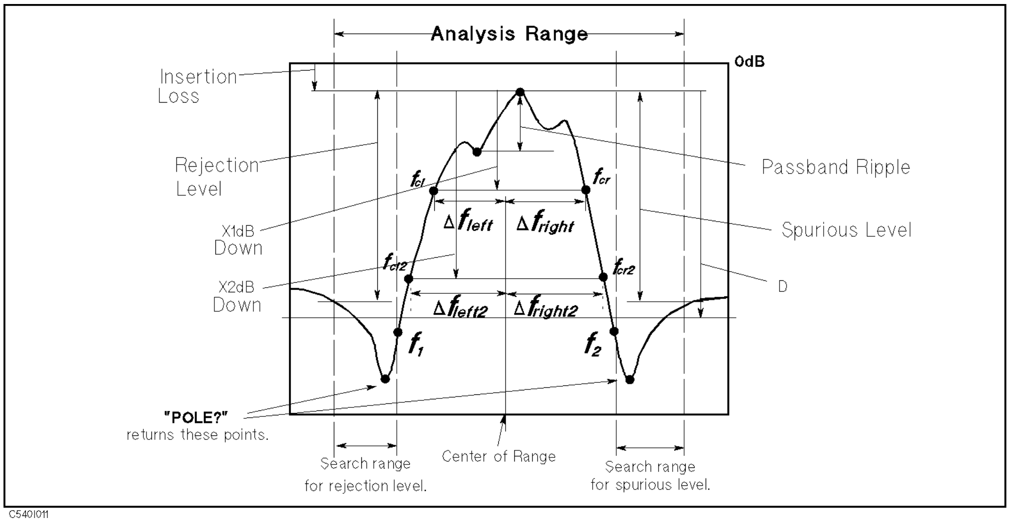


Figure D-11. OUTPXFIL?

Semantics

- Insertion loss, x_1 dB bandwidth, center frequency, Q, Δf_{left}, *and* Δf_{right} are the same as the responses of `OUTPFILT?`.
- Δf_{left2} *and* Δf_{right2} are the frequency differences between both sides at the x_2 dB down cutoff points (f_{cl2} and f_{cr2}) and the middle of the range.
- Passband ripple is the frequency difference of the maximum positive peak and the minimum negative peak between the x_1 dB down cutoff points (f_{cl}, f_{cr}).
- Rejection level is the frequency difference from the insertion loss to the maximum level in the range from the left edge of analysis range to f_1.
- Spurious level is the frequency difference from the insertion loss to the maximum level between f_2 and the right edge of analysis range.
- $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, $Pole_{stim2}$ are the same as the query response of `POLE?` with the parameter *D*.

Note

- If both of the two x_1dB down cutoff points are not found, the analyzer returns 0 for all values of the query response.
- If both of the two x_2dB down cutoff points are not found, the analyzer returns 0 for Δf_{left2} *and* Δf_{right2}.
- If the corresponding peak for `POLE?` is not found, the analyzer returns 0 for $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, *and* $Pole_{stim2}$.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"CENT 70MHZ; SPAN 100KHZ"
30 CALL Sweep(1) ! Goes to sub routine.
40 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
50 OUTPUT @E5100;"OUTPXFIL? -3,-10,-50,69.98MHz,70.02MHz"
60 ENTER @E5100;Loss,Bw,Fc,Q,Dfl,Dfr,Dfl2,Dfr2,Pass,Reject,
Spurious,Pole1,Fp1,Pole2,Fp2
70 PRINT "Loss:";Loss;"[dB] BW:";Bw;"[Hz] fc:";Fc;"[Hz]"
80 PRINT "Q:";Q;" Dfl:";Dfl;"[Hz] Dfr:";Dfr;"[Hz]"
90 PRINT "Dfl2:";Dfl2;"[Hz] Dfr2:";Dfr2;"[Hz] Pass:";Pass;"[dB]"
100 PRINT "Reject:";Reject;"[dB] Spurious:";Spurious;"[dB]"
```

```
110 PRINT "Pole (left):";Pole1;"[dB] ";Fp1;"[Hz]"
120 PRINT "Pole (right):";Pole2;"[dB] ";Fp2;"[Hz]"
130 END
```

OUTPXF2?

Outputs filter parameters within the range specified by the `ANARANG` command.This command outputs the same parameters as `OUTPXFIL` and outputs up to 20 sets of frequency offsets from center of the analysis range to left and right cutoff points. (Query only)

Syntax `OUTPXF2?` D, f_1, f_2, x_1, x_2, ..., x_n Where,

Register	Parameter	Description
0.	D	Difference from maximum value. (Same as `POLE?` parameter.)
1.	f_1	Stop frequency of the range for the rejection level.
2.	f_2	Start frequency of the range for the spurious level.
3.	x_1	The dB value down the bandwidth filter. (1) x_1[dB]
4.	x_2	The dB value down the bandwidth filter. (2) x_2[dB]
⋮	⋮	⋮
n+2.	x_n	The dB value down the bandwidth filter. (n) x_n[dB]

Query Response $Loss$, BW, f_{cent}, Q, $Pass$, $Reject$, $Suprious$, $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, $Pole_{stim2}$, Δf_{left1}, Δf_{right1}, Δf_{left2}, Δf_{right2}, ..., $\Delta f_{left\ n}$, $\Delta f_{right\ n}$ (2n+11)

Register	Parameter	Description
0	$Loss$	Insertion loss
1	BW	x_1 dB down bandwidth
2	f_{cent}	Center frequency
3	Q	Q
4	$Pass$	Passband ripple
5	$Reject$	Rejection level
6	$Spurious$	Spurious level
7	$Pole_{x1}$	First negative peak found to the left of the maximum point.
8	$Pole_{stim1}$	Stimulus of $Pole_{x1}$.
9	$Pole_{x2}$	First negative peaks found to the right of the maximum point.
10	$Pole_{stim2}$	Stimulus of $Pole_{x2}$.
11	Δf_{left}	Frequency difference between the left cutoff point (f_{cl}) and the middle of the range.
12	Δf_{right}	Frequency difference between the right cutoff point (f_{cr}) and the middle of the range.
13	Δf_{left2}	Frequency difference between the left cutoff point (f_{cl2}) and the middle of the range.
14	Δf_{right2}	Frequency difference between the right cutoff point (f_{cr2}) and the middle of the range.
⋮	⋮	⋮
2n+9	$\Delta f_{left\ n}$	Frequency difference between the left cutoff point ($f_{cl\ n}$) and the middle of the range.
2n+10	$\Delta f_{right\ n}$	Frequency difference between the right cutoff point ($f_{cr\ n}$) and the middle of the range.

Semantics

- Insertion loss, x_1 dB bandwidth, center frequency, Q, Δf_{left}, and Δf_{right} are the same as the responses of `OUTPFILT?`.
- $\Delta f_{left\ n}$ and $\Delta f_{right\ n}$ are the frequency differences between both sides at the x_n dB down cutoff points ($f_{cl\ n}$ and $f_{cr\ n}$) and the middle of the range.
- Passband ripple is the frequency difference of the maximum positive peak and the minimum negative peak between the x_1 dB down cutoff points (f_{cl}, f_{cr}).
- Rejection level is the frequency difference from the insertion loss to the maximum level in the range from the left edge of analysis range to f_1.
- Spurious level is the frequency difference from the insertion loss to the maximum level between f_2 and the right edge of analysis range.
- $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, $Pole_{stim2}$ are the same as the query response of `POLE?` with the parameter D.

Note

- If both of the two x_1dB down cutoff points are not found, the analyzer returns `0` for all values of the query response.
- If both of the two x_ndB down cutoff points are not found, the analyzer returns `0` for $\Delta f_{left\ n}$ and $\Delta f_{right\ n}$.
- If the corresponding peak for `POLE?` is not found, the analyzer returns `0` for $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, and $Pole_{stim2}$.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"CENS 70MHz,100kHz"
30 OUTPUT @E5100;"SING?"
40 ENTER @E5100;Tmp
50 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
60 OUTPUT @E5100;"OUTPXF2? -3,69.98MHz,70.02MHz,-3,-10,-12,-50"
70 ENTER @E5100;Loss,Bw,Fc,Q,Pass,Reject,Suprious,Pole1,Fp1,Pole2,Fp2,
   Dfl,Dfr,Dfl,Dfl2,Dfr2,Dfl3,Dfr3,Dfl4,Dfr4
80 PRINT "Loss:";Loss;"[dB] BW:";Bw;"[Hz] fc:";Fc;"[Hz]"
90 PRINT "Q:";Q;" Dfl:";Dfl;"[Hz] Dfr:";Dfr;"[Hz]"
100 PRINT "Dfl2:";Dfl2;"[Hz] Dfr2:";Dfr2;"[Hz] Pass:";Pass;"[dB]"
110 PRINT "Dfl3:";Dfl3;"[Hz] Dfr3:";Dfr3;"[Hz] Pass:";Pass;"[dB]"
120 PRINT "Dfl4:";Dfl4;"[Hz] Dfr4:";Dfr4;"[Hz] Pass:";Pass;"[dB]"
130 PRINT "Reject:";Reject;"[dB] Suprious:";Suprious;"[dB]"
140 PRINT "Pole (left):";Pole1;"[dB] ";Fp1;"[Hz]"
150 PRINT "Pole (right):";Pole2;"[dB] ";Fp2;"[Hz]"
160 END
```

OUTPCFIL?

Analyzes the filter at the nominal frequency, and returns the parameters.

Syntax `OUTPCFIL?` f_c, x_1, x_2, D, f_1, f_2

Where,

Register	Parameter	Description
0	f_c	Nominal frequency
1	x_1	The dB value down the bandwidth filter. (1) x_1[dB]
2	x_2	The dB value down the bandwidth filter. (2) x_2[dB]
3	D	Difference from maximum value. (Same as `POLE?` parameter.)
4	f_1	Stop frequency of the range for the rejection level.
5	f_2	Start frequency of the range for the spurious level.

Query Response $Loss$, $Loss_c$, BW, f_{cent}, Q, Δf_{left1}, Δf_{right1}, Δf_{left2}, Δf_{right2}, $Pass$, $Reject$, $Spurious$, $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, $Pole_{stim2}$ (Total 16)

Register	Parameter	Description
0	$Loss$	Insertion loss
1	$Loss_c$	Const Loss
2	BW	x_1 dB down bandwidth
3	f_{cent}	Center frequency
4	Q	Q
5	Δf_{left}	Frequency difference between the left cutoff point (f_{cl}) and the middle of the range.
6	Δf_{right}	Frequency difference between the right cutoff point (f_{cr}) and the middle of the range.
7	Δf_{left2}	Frequency difference between the left cutoff point (f_{cl2}) and the middle of the range.
8	Δf_{right2}	Frequency difference between the right cutoff point (f_{cr2}) and the middle of the range.
9	$Pass$	Passband ripple
10	$Reject$	Rejection level
11	$Spurious$	Spurious level
12	$Pole_{x1}$	First negative peaks found to the left of the maximum point.
13	$Pole_{stim1}$	Stimulus of $Pole_{x1}$.
14	$Pole_{x2}$	First negative peak found to the right of the maximum point.
15	$Pole_{stim2}$	Stimulus of $Pole_{x2}$.

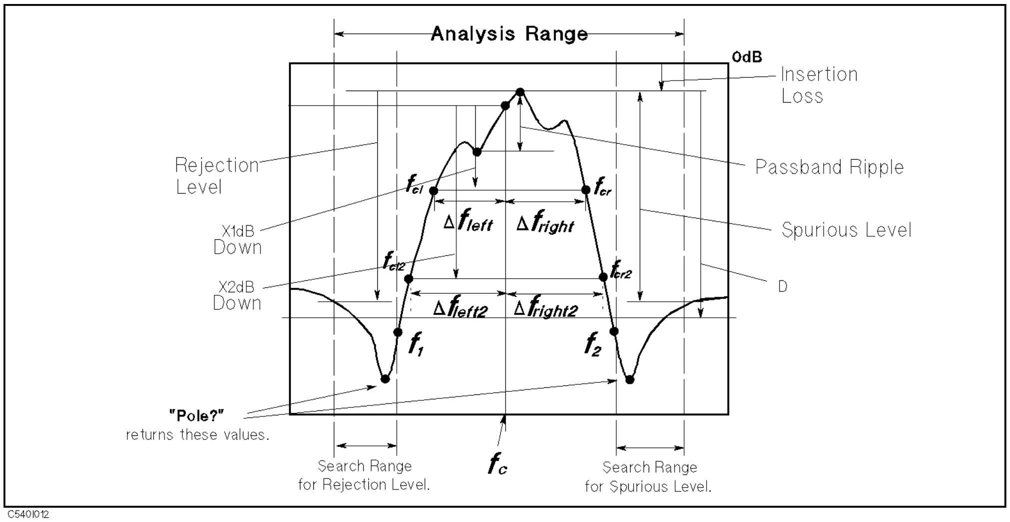


Figure D-12. OUTPCFIL?

Semantics

- Insertion loss, rejection level, spurious level, $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, and $Pole_{stim2}$ are the same as the responses of OUTPXFIL?.
- The const loss is the value of the point that is specified by command parameter, f_c.
- x_1 dB bandwidth is the frequency difference between two x_1 dB down cutoff points (f_{cl}, f_{cr}) from the const loss point.
- Center frequency is the middle point of f_{cl} and f_{cr}.
- Q is calculated using the following equation:

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

- Δf_{left} and Δf_{right} are the frequency differences between both sides at the x_1 dB down cutoff points (f_{cl} and f_{cr}) and f_c.
- Δf_{left2} *and* Δf_{right2} are the frequency differences between both sides at the x_2 dB down cutoff points (f_{cl2} and f_{cr2}) and f_c.
- Passband ripple is the frequency difference of maximum positive peak and minimum negative peak between x_1 dB down cutoff points (f_{cl1}, f_{cr1}).

Note

- If both of the two x_1dB down cutoff points are not found, the analyzer returns 0 for all values of the query response.
- If both of the two x_2dB down cutoff points are not found, the analyzer returns 0 for Δf_{left2} *and* Δf_{right2}.
- If the corresponding peak for POLE? is not found, the analyzer returns 0 for $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, *and* $Pole_{stim2}$.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"CENT 70MHZ; SPAN 100KHZ"
30 CALL Sweep(1) ! Goes to sub routine. (See OUTPFILT?)
40 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
50 OUTPUT @E5100;"OUTPCFIL? 70MHz,-3,-10,-50,69.98MHz,70.02MHz"
```

```
60 ENTER @E5100;Loss,Lc,Bw,Fc,Q,Dfl,Dfr,Dfl2,Dfr2,Pass,Reject,
Spurious,Pole1,Fp1,Pole2,Fp2
70 PRINT "Loss:";Loss;"[dB] Const Loss:";Lc;"[dB]"
80 PRINT "BW:";Bw;"[Hz] Fc:";Fc;"[Hz]"
90 PRINT "Q:";Q;" DFl:";Dfl;"[Hz] DFr:";Dfr;"[Hz]"
100 PRINT "Dfl2:";Dfl2;"[Hz] Dfr2:";Dfr2;"[Hz] Pass:";Pass;"[dB]"
110 PRINT "Reject:";Reject;"[dB] Spurious:";Spurious;"[dB]"
120 PRINT "Pole (left):";Pole1;"[dB] ";Fp1;"[Hz]"
130 PRINT "Pole (right):";Pole2;"[dB] ";Fp2;"[Hz]"
140 END
```

OUTPCF2?

Analyzes the filter at the nominal frequency, and returns the parameters. This command outputs the same parameters as `OUTPCFIL` and outputs up to 20 sets of frequency offsets from center frequency (f_c) to left and right cutoff points ($\Delta f_{left\ n}$, $\Delta f_{right\ n}$).

Syntax `OUTPCFIL?` f_c, D, f_1, f_2, x_1, x_2, ..., x_n

Where,

Register	Parameter	Description
0	f_c	Nominal frequency
1	D	Difference from maximum value. (Same as `POLE?` parameter.)
2	f_1	Stop frequency of the range for the rejection level.
3	f_2	Start frequency of the range for the spurious level.
4	x_1	The dB value down the bandwidth filter. (1) x_1[dB]
5	x_2	The dB value down the bandwidth filter. (2) x_2[dB]
⋮	⋮	⋮ level.
n+3	x_n	The dB value down the bandwidth filter. (n) x_n

Query Response $Loss$, $Loss_c$, BW, f_{cent}, Q, $Pass$, $Reject$, $Suprious$, $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, $Pole_{stim2}$, Δf_{left1}, Δf_{right1}, Δf_{left2}, Δf_{right2}, ..., $\Delta f_{left\ n}$, $\Delta f_{right\ n}$ (2n+11)

Register	Parameter	Description
0	*Loss*	Insertion loss
1	$Loss_c$	Const Loss
2	*BW*	x_1 dB down bandwidth
3	f_{cent}	Center frequency
4	*Q*	Q
5	*Pass*	Passband ripple
6	*Reject*	Rejection level
7	*Spurious*	Spurious level
8	$Pole_{x1}$	First negative peaks found to the left of the maximum point.
9	$Pole_{stim1}$	Stimulus of $Pole_{x1}$.
10	$Pole_{x2}$	First negative peak found to the right of the maximum point.
11	$Pole_{stim2}$	Stimulus of $Pole_{x2}$.
12	Δf_{left}	Frequency difference between the left cutoff point (f_{cl}) and the middle of the range.
13	Δf_{right}	Frequency difference between the right cutoff point (f_{cr}) and the middle of the range.
14	Δf_{left2}	Frequency difference between the left cutoff point (f_{cl2}) and the middle of the range.
15	Δf_{right2}	Frequency difference between the right cutoff point (f_{cr2}) and the middle of the range.
⋮	⋮	⋮
2n+10.	$\Delta f_{left\ n}$	Frequency difference between the left cutoff point ($f_{cl\ n}$) and the middle of the range.
2n+11.	$\Delta f_{right\ n}$	Frequency difference between the right cutoff point ($f_{cr\ n}$) and the middle of the range.

Semantics

- Insertion loss, constant loss, x_1 dB bandwidth, Center frequency, Q, passband ripple, rejection level, spurious level, $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$ are the same as the responses of `OUTPCFILT?`.
- $\Delta f_{left\ n}$ and $\Delta f_{right\ n}$ are the frequency differences between both sides at the x_n dB down cutoff points ($f_{cl\ n}$ and $f_{cr\ n}$) and f_c.

Note

- If both of the two x_1dB down cutoff points are not found, the analyzer returns 0 for all values of the query response.
- If both of the two x_ndB down cutoff points are not found, the analyzer returns 0 for $\Delta f_{left\ n}$ *and* $\Delta f_{right\ n}$.
- If the corresponding peak for `POLE?` is not found, the analyzer returns 0 for $Pole_{x1}$, $Pole_{stim1}$, $Pole_{x2}$, *and* $Pole_{stim2}$.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"CENS 70MHz,100kHz"
30 OUTPUT @E5100;"SING?"
40 ENTER @E5100;Tmp
50 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
60 OUTPUT @E5100;"OUTPCF2? 70MHz,-50,69.98MHz,70.02MHz,-3,-10,-20"
```

```
70 ENTER @E5100;Loss,Lc,Bw,Fc,Q,Pass,Reject,Suprious,Pole1,Fp1,Pole2,Fp2,
   Dfl,Dfr,Dfl2,Dfr2,Dfl3,Dfr3
80 PRINT "Loss:";Loss;"[dB] Const Loss:";Lc;"[dB]"
90 PRINT "BW:";Bw;"[Hz] Fc:";Fc;"[Hz]"
100 PRINT "Q:";Q;" DFl:";Dfl;"[Hz] DFr:";Dfr;"[Hz]"
110 PRINT "Dfl2:";Dfl2;"[Hz] Dfr2:";Dfr2;"[Hz]"
110 PRINT "Dfl3:";Dfl3;"[Hz] Dfr3:";Dfr3;"[Hz] Pass:";Pass;"[dB]"
120 PRINT "Reject:";Reject;"[dB] Suprious:";Suprious;"[dB]"
130 PRINT "Pole (left):";Pole1;"[dB] ";Fp1;"[Hz]"
140 PRINT "Pole (right):";Pole2;"[dB] ";Fp2;"[Hz]"
150 END
```

OUTPRESO?

Returns resonator specific parameters. (Query only)

Syntax `OUTPRESO?`

Query Response Z_r, f_r, Z_a, f_a (Total 4)

Where,

Register	Parameter	Description
0	Z_r	Resonant impedance
1	f_r	Resonant frequency
2	Z_a	Anti-resonant impedance
3	f_a	Anti-resonant frequency

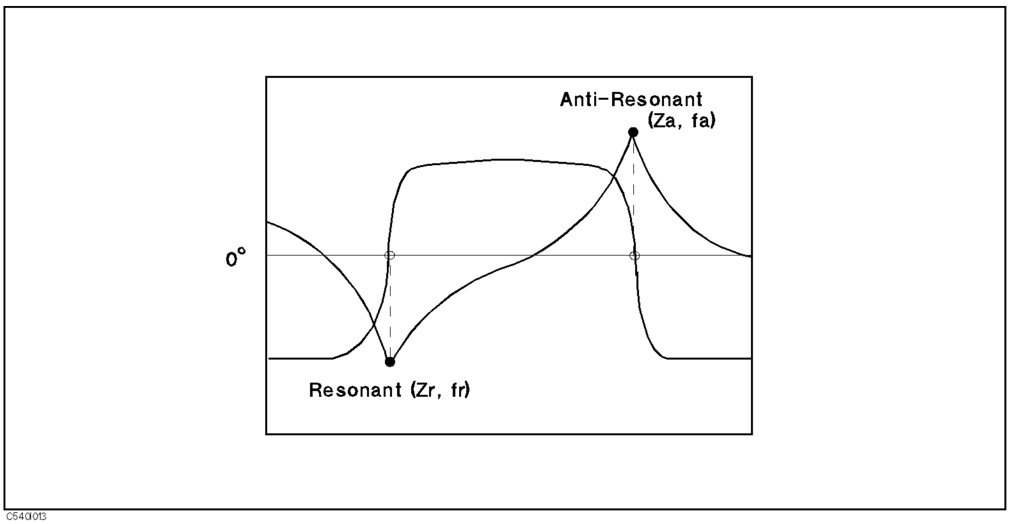


Figure D-13. OUTPRESO?

Semantics

- `OUTPRESO?` executes the following actions and returns their values:
  1. Searches for the 0° phase point from the left edge of the analysis range.
  2. Defines the first point found as the resonant point, and then returns its impedance and its frequency.
  3. Defines the next point found as the anti-resonant point, and then returns its impedance and its frequency.

Note

- You must select the following conditions to use this command:
  - □ Dual Channel & Coupled Channel: ON
  - □ Impedance Conversion: ON
  - □ Analysis channel: LOG MAG format
  - □ Non-analysis channel: Phase format
- `OUTPRESO?` returns the first two found 0° phase point events if there are more than three corresponding points.
- If there is only one 0° phase point in the range, `OUTPRESO?` defines that point as a resonant point and returns `0` for Z_a and f_a.

- If there is no 0° point, `OUTPRESO?` returns 0 for all parameters.
- If the impedance conversion is off, `OUTPRESO?` returns the magnitude (dB) at the 0° phase point.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"DUAC ON; COUC ON"
30 OUTPUT @E5100;"CHAN2; FMT PHAS"
40 OUTPUT @E5100;"CHAN1; FMT LOGM; CONV ZTRA"
50 OUTPUT @E5100;"CENT 70MHZ; SPAN 100KHZ"
60 CALL Sweep(2) ! Goes to sub routine. (See OUTPFILT?)
61               ! Parameter is 2 because of Dual Channel ON.
70 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
80 OUTPUT @E5100;"OUTPRESO?"
90 ENTER @E5100;Zr,Fr,Za,Fa
100 PRINT "Resonant:";Zr;"[ohm],";Fr;"[Hz]"
110 PRINT "Anti-Resonant:";Za;"[ohm],";Fa;"[Hz]"
120 END
```

OUTPRESR?

Returns the resonator specific parameters. (Query only)

Syntax `OUTPRESR?`

Query Response Z_r, f_r, Z_a, f_a, Rpl_1, Rpl_2, Rpl_3 (Total 7)

Where,

Register	Parameter	Description
0	Z_r	Resonant impedance
1	f_r	Resonant frequency
2	Z_a	Anti-resonant impedance
3	f_a	Anti-resonant frequency
4	Rpl_1	Maximum left height of the ripple where is on the left side of the resonant point.
5	Rpl_2	Maximum height right of the ripple that is between the resonant and anti-resonant points.
6	Rpl_3	Maximum height left of the ripple that is on the right side of the resonant point.

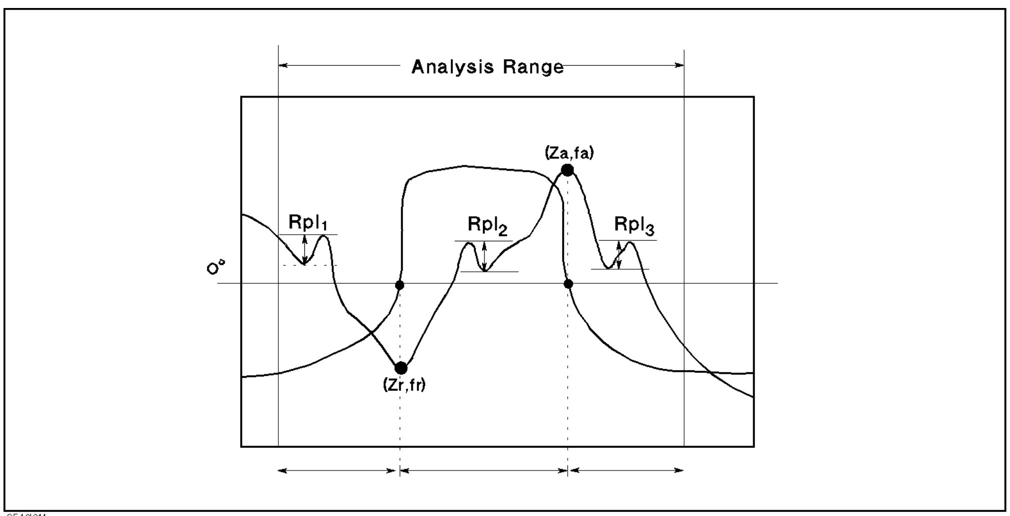


Figure D-14. OUTPRESR?

Semantics

- `OUTPRESR?` executes the following actions:
  1. Searches for the 0° phase point from the left edge of the analysis range.
  2. Defines the first point found as the resonant point, and then returns its impedance and its frequency.
  3. Defines the next point found as the anti-resonant point, and then returns its impedance and its frequency.
  4. Returns the maximum height of the ripple, Rpl_1, that is the difference between the peak and left adjacent negative peak.
  5. Returns the maximum height of the ripple, Rpl_2, that is the difference between the peak and right adjacent negative peak.

6. Returns the maximum height of the ripple, Rpl_3, that is the difference between the peak and left adjacent negative peak.

Note

- You must select the following conditions to use this command:
  - □ Dual Channel & Coupled Channel: ON
  - □ Impedance Conversion: ON
  - □ Analysis channel: LOG MAG format
  - □ Non-analysis channel: Phase format
- OUTPRESR? returns the first two 0° phase point events found if there are more than three corresponding points points.
- If there is only one 0° phase point in the range, OUTPRESR? defines that point as a resonant point and returns 0 for Z_a, f_a, Rpl_1, Rpl_2, and Rpl_3.
- If there is no 0° point, OUTPRESR? returns 0 for all parameters.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"DUAC ON; COUC ON"
30 OUTPUT @E5100;"CHAN2; FMT PHAS"
40 OUTPUT @E5100;"CHAN1; FMT LOGM; CONV ZTRA"
50 OUTPUT @E5100;"CENT 70MHZ; SPAN 100KHZ"
60 CALL Sweep(2) ! Goes to sub routine. (See OUTPFILT?)
70 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
80 OUTPUT @E5100;"OUTPRESR?"
90 ENTER @E5100;Zr,Fr,Za,Fa,R1,R2,R3
100 PRINT "Resonant:";Zr;"[ohm],";Fr;"[Hz]"
110 PRINT "Anti-Resonant:";Za;"[ohm],";Fa;"[Hz]"
120 PRINT "Ripple L:";R1;"[dB]"
130 PRINT "Ripple M:";R2;"[dB]"
140 PRINT "Ripple R:";R3;"[dB]"
150 END
```

OUTPRESF?

Returns the resonator specific parameters. (Query only)

Syntax `OUTPRESF?` x_1, x_2

Where,

Register	Parameter	Description
0	x_1	Value down from the maximum peak.
1	x_2	Value above the maximum peak.

Query Response f_s, f_p, f_{s1}, f_{s2}, f_{p1}, f_{p2} (Total 6)

Where,

Register	Parameter	Description
0	f_s	Middle point frequency between f_{s1} and f_{s2}.
1	f_p	Middle point frequency between f_{p1} and f_{p2}.
2	f_{s1}	Left one of the two points x_1 dB down from the maximum peak.
3	f_{s2}	Right one of the two points x_1 dB down from the maximum peak.
4	f_{p1}	Left one of the two points x_2 dB above the minimum negative peak.
5	f_{p2}	Right one of the two points x_2 dB above the minimum negative peak.

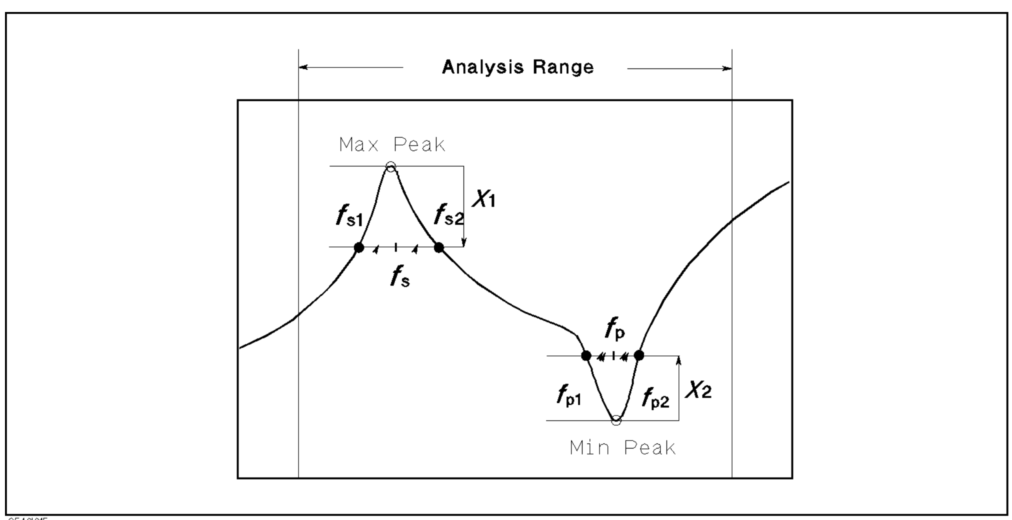


Figure D-15. OUTPRESF?

Semantics

- `OUTPRESF?` executes the following actions:
  1. Searches for the maximum peak in the analysis range.
  2. Searches for the x_1 dB below points on both sides, and defines the first found left and right side points as f_{s1} and f_{s2}, respectively.

3. Defines the middle point between f_{s1} and f_{s2} to f_s.
4. Searches for the x_2 dB above points on both sides, and defines the first found left and right side points as f_{p1} and f_{p2}, respectively.
5. Defines the middle point between f_{p1} and f_{p2} as f_p.

Note

- If there is no corresponding peak in the range, `OUTPRESF?` returns `0` for all parameters.
- If the maximum peak cannot be found, `OUTPRESF?` returns `0` for f_s, f_{s1}, and f_{s2}.
- If the minimum negative peak cannot be found, `OUTPRESF?` returns `0` for f_p, f_{p1}, and f_{p2}.
- Specify the negative value for x_1 and positive value for x_2.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"FMT LOGM; CENT 60.06MHz; SPAN 20kHz"
30 CALL Sweep(1) ! Goes to sub routine. (See OUTPFILT?)
40 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
50 OUTPUT @E5100;"OUTPRESF? -3dB,3dB"
60 ENTER @E5100;Fs,Fp,Fs1,Fs2,Fp1,Fp2
70 PRINT "Series-Resonant:";Fs;"[Hz]"
80 PRINT "Parallel-Resonant:";Fp;"[Hz]"
90 END
```

OUTPCERR?

Returns the ceramic resonator specific parameters. (Query only)

Syntax `OUTPCERR?`

Query Response Z_r, f_r, Z_a, f_a, Rpl_1, Rpl_2, Rpl_3 (Total7)

Where,

Register	Parameter	Description
0	Z_r	Resonant impedance
1	f_r	Resonant frequency
2	Z_a	Anti-resonant impedance
3	f_a	Anti-resonant frequency
4	Rpl_1	Maximum height of the ripple that is on the left side of the resonant point.
5	Rpl_2	Maximum height of the ripple that is between the resonant and anti-resonant points.
6	Rpl_3	Maximum height of the ripple that is on the right side of the resonant point.

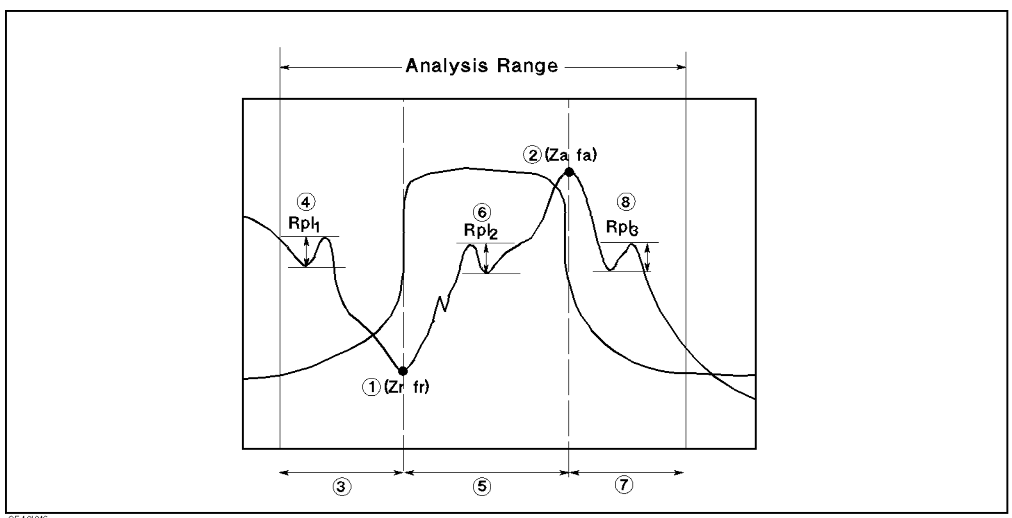


Figure D-16. OUTPCERR?

Semantics

- You need to select the LOG MAG format (`FMT LOGM`) and turn impedance conversion on (`CONV ZTRA`) to use this command.
- `OUTPCERR?` executes the following actions:

1. Searches for the minimum negative peak in the range and defines it as a resonant point. Then returns the resonant impedance, Z_r, and resonant frequency, f_r.
2. Searches for the maximum peak in the range and defines it as a anti-resonant point. Then returns the anti-resonant impedance, Z_a, and anti-resonant frequency, f_p.

3. Returns the maximum height of the ripple, Rpl_1, that is the difference between the peak and left adjacent negative peak.
4. Returns the maximum height of the ripple, Rpl_2, that is the difference between the peak and right adjacent negative peak.
5. Returns the maximum height of the ripple, Rpl_3, that is the difference between the peak and left adjacent negative peak.

Note

- This command can be used when the LOG MAG format (`FMT LOGM`) is selected. If another format is selected, `OUTPCERR?` returns `0` for all parameters.
- If no corresponding ripple is found, `OUTPCERR?` returns `0`.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"FMT LOGM; CONV ZTRA; CENT 60.02MHz; SPAN 20kHz"
30 CALL Sweep(1) ! Goes to sub routine. (See OUTPFILT?)
40 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
50 OUTPUT @E5100;"OUTPCERR?"
60 ENTER @E5100;Zr,Fr,Za,Fa,R1,R2,R3
70 PRINT "Resonant:";Zr;"[ohm],";Fr;"[Hz]"
80 PRINT "Anti-Resonant:";Za;"[ohm],";Fa;"[Hz]"
90 PRINT "Ripple L:";R1;"[dB]"
100 PRINT "Ripple M:";R2;"[dB]"
110 PRINT "Ripple R:";R3;"[dB]"
120 END
```

Equivalent circuit analysis commands

The following commands are for the equivalent circuit analysis. They are easy to use for specific device analysis because they can output many parameters using only a single command.

- `EQUCPARA?`
  - `EQUM`
- `EQUCPARA5?`
- `EQUCPARS?`
- `EQUCO?`
- `EQUCPARS4?`

EQUCPARA?

Returns the six-device equivalent circuit parameters of the crystal resonator. (Query only)

Syntax `EQUCPARA?`

Query Response C_0, C_1, L_1, R_1, G_0, R_0 (Total 6)

Where,

Register	Parameter	Description
0	C_0	Parallel capacitance
1	C_1	Motional capacitance
2	L_1	Motional inductance
3	R_1	Motional resistance
4	G_0	Electrode conductance
5	R_0	Electrode resistance

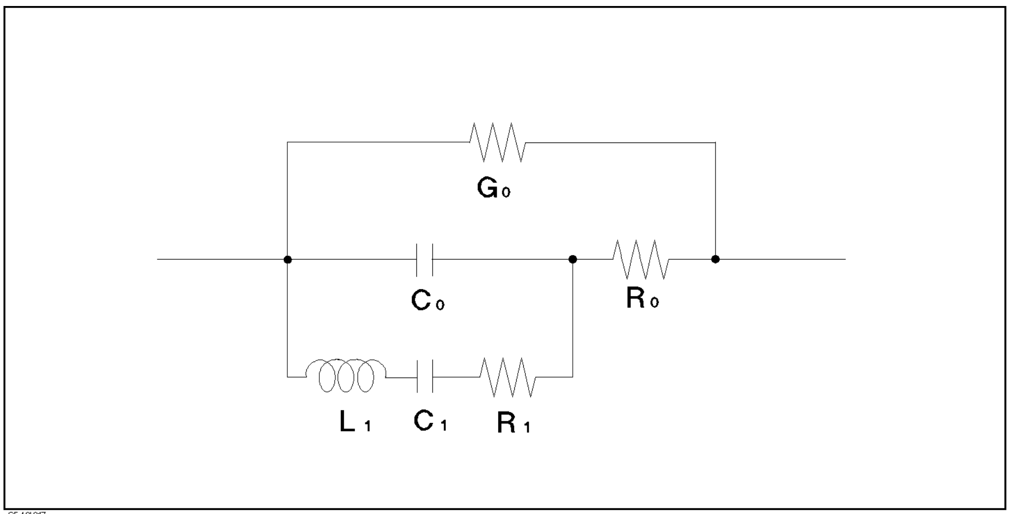


Figure D-17. Six-Device Equivalent Circuit of Crystal Resonator

Semantics

- `EQUCPARA?` executes the following actions:

1. Obtains the admittance characteristic circle diagram.
2. Obtains the maximum conductance. (G_{max})
3. Obtains frequencies f_1 and f_2 ($f_1 < f_2$) of the two points where the conductance is half the maximum conductance (G_{max}).
4. Calculates f_s by $f_s = \sqrt{f_1 \times f_2}$.
5. Obtains susceptance B_{fs} at f_s.
6. Calculates ω_s by $\omega_s = 2 \times \pi \times f_s$.
7. Assumes that the frequency at which the phase becomes $0°^*$ near the parallel resonance frequency is f_a, and obtains its conductance G_a.
8. Calculates ω_a by $\omega_a = 2 \times \pi \times f_a$.

9. Assumes that the frequency at which the phase becomes 0°* near the series resonance frequency is f_r.
10. Calculates the constants using the above values and the following equations:

$$Q_s = \frac{f_s}{f_2 - f_1} \qquad C_0{}' = \frac{B_1 + B_2}{2\omega_s}$$

$$L_1 = \frac{Q_s}{\omega_s G_{max}} \qquad R_1 = \frac{C_0{}'}{C_0 G_{max}}$$

$$C_1 = \frac{G_{max}}{\omega_s Q_s} \qquad R_0 = \frac{1}{G_{max}} - R_1$$

$$C_0 = \frac{B_{fs}}{\omega_s} \qquad G_0 = G_a - \frac{R_1 \omega_a{}^2 C_0{}^2}{1 + R_0 R_1 \omega_a{}^2 C_0{}^2}$$

* "EQUCPARA?" interpolates the 0° phase points even if it does not exist in measured data.

- If the number of points between the maximum peak point (f_{Bmax}) and the minimum peak point (f_{Bmin}) of the conductance is less than 10 points, `EQUCPARA?` approximates an admittance circle. The circle approximation can be performed if there are 3 points for analysis. You can specify how many points are used for circle approximation using the `EQUM` command to reduce the analysis time.
- If `EQUCPARA?` fails the circle approximation, `0` will be return for all parameters.
- If there are only 2 points for analysis, `EQUCPARA?` returns four-device equivalent circuit parameters. In this case, `EQUCPARA?` returns `0` for G_0 and R_0.
- If there is only 1 point for analysis, `EQUCPARA?` returns `0` for all parameters.

EQUM *value*

Specifies how many points are used for an approximation of a circle with the `EQUCPARA?` and `EQUCPARS?` commands. `EQUCPARA?` (or `EQUCPARS?`) thins the measured points out for the specified points, then makes a circle approximation. When the `EQUM` parameter is set greater than the number of points, `EQUCPARA?` uses all the points for the circle approximation. Default value is 8.

value 2 to 801

Note

- You must select the following conditions or Polar format to use this command:
  - □ Dual Channel & Coupled Channel: ON
  - □ Impedance Conversion: ON
  - □ Analysis channel: LOG MAG format
  - □ Non-analysis channel: Phase format

If another format is selected, `0` will be returned for query response.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"FMT POLA; CONV YTRA; CENT 60.06MHz; SPAN 20kHz"
30 CALL Sweep(1) ! Goes to sub routine. (See OUTPFILT?)
40 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
50 OUTPUT @E5100;"EQUCPARA?"
60 ENTER @E5100;C0,C1,L1,R1,G0,R0
70 PRINT "C0:";C0;" C1:";C1
80 PRINT "L1:";L1;" R1:";R1
90 PRINT "G0:";G0;" R0:";R0
100 END
```

EQUCPARA5?

Executes the equivalent circuit analysis for a resonator using the same equivalent circuit of `EQUCPARA?`, but does not output G0. `EQUCPARA5?` does not display the warning even an anti-resonance frequency is not in the analysis range. (Query only)

Syntax `EQUCPARA5?`

Query Response C_0, C_1, L_1, R_1, R_0 (Total 5)

Where,

Register	Parameter	Description
0	C_0	Parallel capacitance
1	C_1	Motional capacitance
2	L_1	Motional inductance
3	R_1	Motional resistance
4	R_0	Electrode resistance

EQUCPARS?

Outputs the six-device equivalent circuit parameters of the crystal resonator. (Query only)

Syntax `EQUCPARS?`

Query Response C_0, C_1, L_1, R_1, f_s, f_a, f_r, $f_1{}^*$, $f_2{}^*$, G_0, R_0 (Total 11)

$^*\ f_1 < f_2$

For information about each parameter, see "EQUCPARA?".

Note

- You must select the following conditions or Polar format to use this command:
  - Dual Channel & Coupled Channel: ON
  - Impedance Conversion: ON
  - Analysis channel: LOG MAG format
  - Non-analysis channel: Phase format

  If another format is selected, `0` will be returned for query response.

EQUC0? *value*

Returns the parallel capacitance (C_0) of the equivalent circuit of the resonator at the specified frequency. (Query only)

Syntax `EQUC0?` *value*

Where,

Register	Parameter	Description
0	*value*	Frequency for C_0

Query Response C_0

Where,

Register	Parameter	Description
0	C_0	Parallel capacitance

Semantics

- C_0 is calculated using the following equation:

$$C_0 = \frac{B_s}{\omega_s}$$

Where,

B_s	Imaginary part of the point on f_s.
ω_s	$= 2\pi f_s$
f_s	Frequency that is specified by the command parameter.

- If the impedance conversion is selected, C_0 is calculated using the following equation:

$$C_0 = \frac{-1}{B_s \times \omega_s}$$

Note

- You must select the following conditions or Polar format to use this command:
  - □ Dual Channel & Coupled Channel: ON
  - □ Impedance Conversion: ON
  - □ Analysis channel: LOG MAG format
  - □ Non-analysis channel: Phase format

  If another format is selected, 0 will be returned for query response.
- If the specified frequency is out of analysis range, 0 will be returned.
- If B_s is 0 when the impedance conversion is selected, `EQUC0?` returns 0.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"DUAC ON; COUC ON"
30 OUTPUT @E5100;"CHAN2; FMT PHAS"
40 OUTPUT @E5100;"CHAN1; FMT LOGM; CONV ZTRA"
50 OUTPUT @E5100;"CENT 60.06MHz; SPAN 20kHz"
60 CALL Sweep(2) ! Goes to sob routine. (See OUTPFILT?)
70 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
80 OUTPUT @E5100;"EQUC0? 60.06MHz"
90 ENTER @E5100;C0
100 PRINT "C0:";C0
```

```
110 END
```

EQUCPARS4?

Returns the 4-device equivalent circuit parameters of the crystal resonator. (Query only)

Syntax `EQUCPARS4?`

Query Response C_0, C_1, L_1, R_1, f_s, f_a, f_r, f_1, f_2 (Total9)

Where,

Register	Parameter	Description
0	C_0	Parallel capacitance
1	C_1	Motional capacitance
2	L_1	Motional inductance
3	R_1	Motional resistance
4	f_s	Motional (parallel) resonant frequency
5	f_a	Anti-resonant frequency
6	f_r	Resonant frequency
7	f_1	Frequency at the point where the half of maximum conductance.
8	f_2	Frequency at the point where the half of maximum conductance. ($f_1 < f_2$)

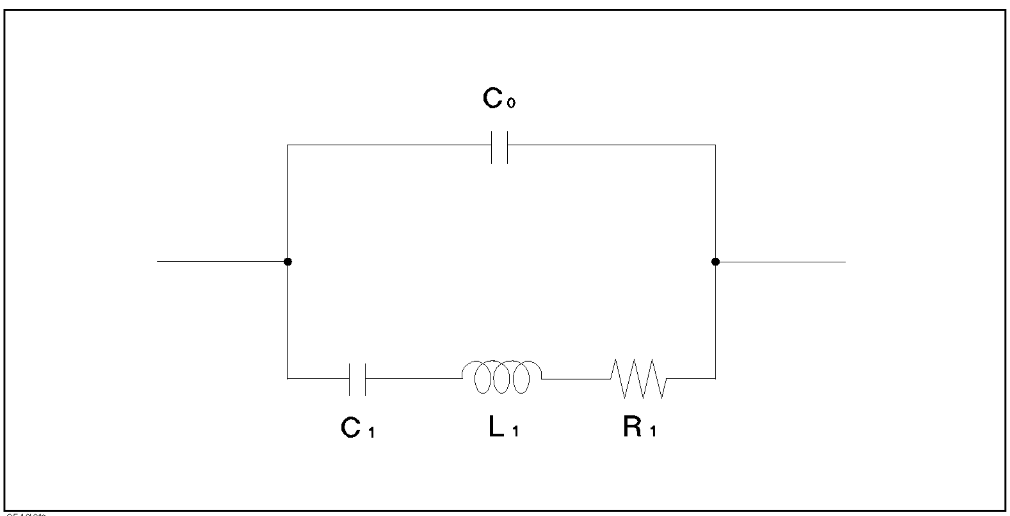


Figure D-18. Four-Device Equivalent Circuit of Crystal Resonator

Semantics

- You need to select the polar format (`FMT POLA`) and turn the admittance conversion on to use this command.
- `EQUCPARS4?` executes the following actions:

1. Obtains the admittance characteristic circle diagram. (See Figure D-19.)
2. Obtains the susceptance (B_{fs}) and its frequency (f_s) at the maximum conductance (G_{max}) point.
3. Obtains frequencies f_1 and f_2 ($f_1 < f_2$) of the two points where the conductance is half the maximum conductance (G_{max}).

4. Assumes that the frequency at which the phase becomes 0° near the parallel resonance frequency is f_a.
5. Assumes that the frequency at which the phase becomes 0° near the series resonance frequency is f_r.
6. Calculates the constants using the above values and the following equations:

$$C_0 = \frac{{f_r}^2}{{f_a}^2 - {f_r}^2} \times C_1 \qquad Q = \left|\frac{f_s}{f_2 - f_1}\right|$$

$$C_1 = \frac{1}{QR_1 2\pi f_s} \qquad R_1 = \frac{1}{G_{max}}$$

$$L_1 = \frac{QR_1}{2\pi f_s}$$

If there are no f_r and f_a points on the admittance chart, C_0 is calculated using the following equation:

$$C_0 = \frac{B_{fs}}{2\pi f_s}$$

Where, B_{fs} is the susceptance at the G_{max} point.

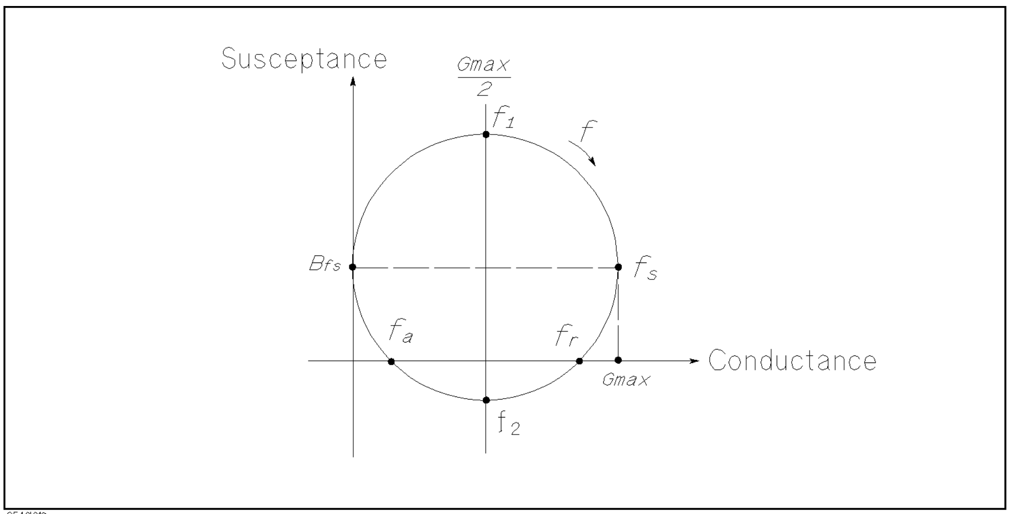


Figure D-19. Admittance Characteristic Circle Diagram

Note * This command is only available when Polar format and the admittance conversion is on. If these are not selected, 0 will be returned.

Examples

```
10 ASSIGN @E5100 TO 717
20 OUTPUT @E5100;"FMT POLA; CONV YTRA; CENT 60.06MHz; SPAN 20kHz"
30 CALL Sweep(1) ! Goes to sub routine. (See OUTPFILT?)
40 OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA"
50 OUTPUT @E5100;"EQUCPARS4?"
60 ENTER @E5100;C0,C1,L1,R1
70 PRINT "C0:";C0;" C1:";C1
80 PRINT "L1:";L1;" R1:";R1
90 END
```

E

Command 87510A v.s. E5100A

COMMAND

COMMAND	87510A	E5100A
*CLS	SUPPORT	SUPPORT
*ESE	SUPPORT	SUPPORT
*ESR?	SUPPORT	SUPPORT
*IDN?	SUPPORT	SUPPORT
*PCB	SUPPORT	SUPPORT
*RST	SUPPORT	SUPPORT
*SRE	SUPPORT	SUPPORT
*STB?	SUPPORT	SUPPORT
*TRG	SUPPORT	SUPPORT
*TST?	SUPPORT	SUPPORT
*WAI	SUPPORT	SUPPORT
ADDRCONT	SUPPORT	NO SUPPORT
ADDRPIN	SUPPORT	NO SUPPORT
ADDRPLOT	NO SUPPORT	NO SUPPORT
ADTOTRAC	NO SUPPORT	SUPPORT
ANAMODE	NO SUPPORT	SUPPORT
ANAOCH	SUPPORT	SUPPORT
ANAODATA	SUPPORT	SUPPORT
ANAOMEMO	SUPPORT	SUPPORT
ANARANG	SUPPORT	SUPPORT
ANARANGP	NO SUPPORT	SUPPORT
ANARFULL	SUPPORT	SUPPORT
ANASTIMP?	NO SUPPORT	SUPPORT
ANAPOINS?	NO SUPPORT	SUPPORT
AR	SUPPORT	NO SUPPORT
ASCE	NO SUPPORT	NO SUPPORT
ATTI	NO SUPPORT	SUPPORT
ATTIAAUTO	NO SUPPORT	SUPPORT
ATTIBAUTO	NO SUPPORT	SUPPORT
ATTICAUTO	NO SUPPORT	SUPPORT
ATTIRAUTO	NO SUPPORT	SUPPORT
ATTN	SUPPORT	NO SUPPORT
ATTW	SUPPORT	SUPPORT
AUTO	SUPPORT	SUPPORT
AVER	NO SUPPORT	NO SUPPORT

COMMAND	87510A	E5100A
AVERFACT	NO SUPPORT	NO SUPPORT
AVERREST	NO SUPPORT	NO SUPPORT
BASL	NO SUPPORT	SUPPORT
BEEPDONE	SUPPORT	SUPPORT
BEEPFAIL	SUPPORT	SUPPORT
BEEPWARN	SUPPORT	SUPPORT
BINCLEL	NO SUPPORT	NO SUPPORT
BINEDONE	NO SUPPORT	NO SUPPORT
BINESB	NO SUPPORT	NO SUPPORT
BINL	NO SUPPORT	NO SUPPORT
BINO	NO SUPPORT	NO SUPPORT
BINOA	NO SUPPORT	NO SUPPORT
BINOB	NO SUPPORT	NO SUPPORT
BINP	NO SUPPORT	NO SUPPORT
BINS	NO SUPPORT	NO SUPPORT
BINSADD	NO SUPPORT	NO SUPPORT
BINSDEL	NO SUPPORT	NO SUPPORT
BINSDON	NO SUPPORT	NO SUPPORT
BINSEDI	NO SUPPORT	NO SUPPORT
BINSIZE	SUPPORT	SUPPORT
BINSLINE	NO SUPPORT	NO SUPPORT
BINU	NO SUPPORT	NO SUPPORT
C0	SUPPORT	NO SUPPORT
C1	SUPPORT	NO SUPPORT
C2	SUPPORT	NO SUPPORT
CALCASSI	SUPPORT	NO SUPPORT
CALCOPY	NO SUPPORT	SUPPORT
CALI	SUPPORT	SUPPORT
CALIRAI	SUPPORT	NO SUPPORT
CALIRESP	SUPPORT	NO SUPPORT
CALIS111	SUPPORT	NO SUPPORT
CALK	SUPPORT	SUPPORT
CALK7MM	NO SUPPORT	NO SUPPORT
CALKN50	NO SUPPORT	NO SUPPORT
CALKN75	NO SUPPORT	NO SUPPORT
CALKUSED	SUPPORT	NO SUPPORT
CALN	SUPPORT	NO SUPPORT
CALS	SUPPORT	NO SUPPORT
CENS	SUPPORT	SUPPORT
CENT	SUPPORT	SUPPORT
CHAD	SUPPORT	SUPPORT
CHAIRANG	SUPPORT	NO SUPPORT
CHAN	NO SUPPORT	SUPPORT
CHAN1	SUPPORT	SUPPORT
CHAN2	SUPPORT	SUPPORT

COMMAND	87510A	E5100A
CIN	SUPPORT	SUPPORT
CIVAL	NO SUPPORT	SUPPORT
CLAD	SUPPORT	NO SUPPORT
CLASS11	SUPPORT	SUPPORT
CLEL	SUPPORT	SUPPORT
CLEM	SUPPORT	SUPPORT
CLEMNU3	NO SUPPORT	SUPPORT
CLES	SUPPORT	SUPPORT
CONT	SUPPORT	SUPPORT
CONV	SUPPORT	SUPPORT
CONV1DS	NO SUPPORT	NO SUPPORT
CONVMP16	SUPPORT	NO SUPPORT
CONVMP4	SUPPORT	NO SUPPORT
CONVMP8	SUPPORT	NO SUPPORT
CONVOFF	SUPPORT	NO SUPPORT
CONVYREF	SUPPORT	NO SUPPORT
CONVYTRA	SUPPORT	NO SUPPORT
CONVZREF	SUPPORT	NO SUPPORT
CONVZTRA	SUPPORT	NO SUPPORT
COPA	SUPPORT	NO SUPPORT
COPT	SUPPORT	NO SUPPORT
COPYRIGHT	NO SUPPORT	SUPPORT
CORR	SUPPORT	SUPPORT
CORRS?	NO SUPPORT	SUPPORT
COUC	SUPPORT	SUPPORT
COUT	SUPPORT	SUPPORT
CRED	SUPPORT	SUPPORT
CURD?	SUPPORT	SUPPORT
CURMPOIN?	NO SUPPORT	SUPPORT
CWFREQ	SUPPORT	SUPPORT
DATAM	NO SUPPORT	SUPPORT
DATAMN	NO SUPPORT	SUPPORT
DATAM3TER	SUPPORT	NO SUPPORT
DATAMNONE	SUPPORT	NO SUPPORT
DATAMTHRU	SUPPORT	NO SUPPORT
DATI	NO SUPPORT	SUPPORT
DAYMYEAR	SUPPORT	SUPPORT
DCBUS	SUPPORT	NO SUPPORT
DEFS	SUPPORT	NO SUPPORT
DELA	SUPPORT	SUPPORT
DELO	SUPPORT	SUPPORT
DELR	SUPPORT	SUPPORT
DELRFIXM	SUPPORT	SUPPORT
DESTOFF	SUPPORT	NO SUPPORT
DESTON	SUPPORT	NO SUPPORT

COMMAND	87510A	E5100A
DFLT	NO SUPPORT	NO SUPPORT
DIN	SUPPORT	SUPPORT
DISA	SUPPORT	SUPPORT
DISAALLB	SUPPORT	NO SUPPORT
DISAALLI	SUPPORT	SUPPORT
DISABASS	SUPPORT	SUPPORT
DISAHIHB	SUPPORT	SUPPORT
DISBLIST	NO SUPPORT	SUPPORT
DISFDOS	SUPPORT	NO SUPPORT
DISFLIF	NO SUPPORT	NO SUPPORT
DISG	SUPPORT	SUPPORT
DISL1	SUPPORT	NO SUPPORT
DISL2	SUPPORT	NO SUPPORT
DISLLIST	SUPPORT	NO SUPPORT
DISMCTSP	SUPPORT	NO SUPPORT
DISMMD	SUPPORT	NO SUPPORT
DISMNUM	SUPPORT	NO SUPPORT
DISMSTEP	SUPPORT	NO SUPPORT
DISMSTSP	SUPPORT	NO SUPPORT
DISMUL	SUPPORT	NO SUPPORT
DISP	SUPPORT	SUPPORT
DISPDATA	SUPPORT	NO SUPPORT
DISPDATM	SUPPORT	NO SUPPORT
DISPDDM	SUPPORT	NO SUPPORT
DISPDMM	SUPPORT	NO SUPPORT
DISPMEMO	SUPPORT	NO SUPPORT
DIST	SUPPORT	NO SUPPORT
DONE	SUPPORT	SUPPORT
DOUT	SUPPORT	SUPPORT
DSKEY	SUPPORT	SUPPORT
DUAC	SUPPORT	SUPPORT
EDITBINL	NO SUPPORT	NO SUPPORT
EDITDONE	SUPPORT	SUPPORT
EDITLIML	SUPPORT	NO SUPPORT
EDITLIS1	SUPPORT	SUPPORT
EDITLIS2	SUPPORT	SUPPORT
EDITLIST	SUPPORT	SUPPORT
ELED	SUPPORT	SUPPORT
ENKEY	SUPPORT	SUPPORT
EQUC0?	SUPPORT	SUPPORT
EQUCPARA5	NO SUPPORT	SUPPORT
EQUCPARA?	SUPPORT	SUPPORT
EQUCPARS4?	SUPPORT	SUPPORT
EQUCPARS?	SUPPORT	SUPPORT
EQUM	SUPPORT	SUPPORT

COMMAND	87510A	E5100A
ERRH	SUPPORT	NO SUPPORT
ESB?	SUPPORT	SUPPORT
ESNB	SUPPORT	SUPPORT
EXET	SUPPORT	SUPPORT
EXPP	SUPPORT	NO SUPPORT
EXPZP	NO SUPPORT	SUPPORT
EXTRLOCK?	SUPPORT	SUPPORT
EXTT	SUPPORT	SUPPORT
EXTTOFF	SUPPORT	NO SUPPORT
EXTTON	SUPPORT	NO SUPPORT
EXTTPOIN	SUPPORT	NO SUPPORT
FBUS	SUPPORT	NO SUPPORT
FILC	SUPPORT	SUPPORT
FIRLPNOR	SUPPORT	NO SUPPORT
FIRLPOPE	SUPPORT	NO SUPPORT
FIRR?	SUPPORT	NO SUPPORT
FMT	SUPPORT	SUPPORT
FMTMAGZDF	NO SUPPORT	SUPPORT
FNDAUTO	SUPPORT	NO SUPPORT
FNDMANU	SUPPORT	NO SUPPORT
FNDVALU	SUPPORT	NO SUPPORT
FNVNARR	SUPPORT	NO SUPPORT
FNVNORM	SUPPORT	NO SUPPORT
FNVOPEN	SUPPORT	NO SUPPORT
FNVWIDE	SUPPORT	NO SUPPORT
FORM2	SUPPORT	SUPPORT
FORM3	SUPPORT	SUPPORT
FORM4	SUPPORT	SUPPORT
FORM5	SUPPORT	SUPPORT
FREELIST	SUPPORT	NO SUPPORT
FREO	SUPPORT	NO SUPPORT
FULP	NO SUPPORT	NO SUPPORT
GRAE	NO SUPPORT	NO SUPPORT
GRAPCOL	NO SUPPORT	SUPPORT
GRAPFORM	NO SUPPORT	SUPPORT
GRODAPER	SUPPORT	SUPPORT
HOLD	SUPPORT	SUPPORT
IFBW	SUPPORT	SUPPORT
IFBWAUTO	SUPPORT	SUPPORT
IFRCH?	SUPPORT	NO SUPPORT
IFRX1	SUPPORT	NO SUPPORT
IFRX1X8	SUPPORT	NO SUPPORT
IFRX64	SUPPORT	NO SUPPORT
IFRX8X1	SUPPORT	NO SUPPORT
IMAG	SUPPORT	NO SUPPORT

COMMAND	87510A	E5100A
INID	SUPPORT	SUPPORT
INP8IO	SUPPORT	SUPPORT
INPT?	SUPPORT	SUPPORT
INPUCALC{01-03}	SUPPORT	SUPPORT
INPUCALK	SUPPORT	NO SUPPORT
INPUD	SUPPORT	NO SUPPORT
INPUDATA	NO SUPPORT	SUPPORT
INPUDATM	NO SUPPORT	SUPPORT
INPUDATM1	SUPPORT	NO SUPPORT
INPUDATM2	SUPPORT	NO SUPPORT
INPUDATM3	SUPPORT	NO SUPPORT
INPUDATM4	SUPPORT	NO SUPPORT
INPUDATTP	SUPPORT	NO SUPPORT
INPUFORM	SUPPORT	SUPPORT
INPUIFORM	NO SUPPORT	SUPPORT
INPULOAA	SUPPORT	SUPPORT
INPUMEMO	NO SUPPORT	SUPPORT
INPUMEMTP	SUPPORT	NO SUPPORT
INPUOPEA	SUPPORT	NO SUPPORT
INPURAW	NO SUPPORT	SUPPORT
INPURFORM	NO SUPPORT	SUPPORT
INPURTMEM	NO SUPPORT	SUPPORT
INPUSHOA	SUPPORT	NO SUPPORT
INPUTMEM	SUPPORT	NO SUPPORT
INPUTRAC	SUPPORT	SUPPORT
INPUTRACB	SUPPORT	SUPPORT
IOPO?	SUPPORT	SUPPORT
KEY	SUPPORT	NO SUPPORT
KITD	SUPPORT	NO SUPPORT
LABERES{P|I}	SUPPORT	NO SUPPORT
LABES11{A|B|C}	SUPPORT	NO SUPPORT
LABK	SUPPORT	NO SUPPORT
LABS	SUPPORT	NO SUPPORT
LEFL	NO SUPPORT	NO SUPPORT
LEFU	NO SUPPORT	NO SUPPORT
LIMCLEL	SUPPORT	NO SUPPORT
LIMD	SUPPORT	NO SUPPORT
LIMEDONE	SUPPORT	NO SUPPORT
LIMIAMPO	SUPPORT	NO SUPPORT
LIMILINE	SUPPORT	SUPPORT
LIMIMAOF	SUPPORT	NO SUPPORT
LIMIOPOIN	SUPPORT	NO SUPPORT
LIMIOSEND	SUPPORT	NO SUPPORT
LIMISTIO	SUPPORT	NO SUPPORT
LIMITEST	SUPPORT	SUPPORT

COMMAND	87510A	E5100A
LIML	SUPPORT	NO SUPPORT
LIMM	SUPPORT	NO SUPPORT
LIMS	SUPPORT	NO SUPPORT
LIMSADD	SUPPORT	NO SUPPORT
LIMSDEL	SUPPORT	NO SUPPORT
LIMSDON	SUPPORT	NO SUPPORT
LIMSEDI	SUPPORT	NO SUPPORT
LIMU	SUPPORT	NO SUPPORT
LINFREQ	SUPPORT	NO SUPPORT
LINM	SUPPORT	NO SUPPORT
LINM{F|P}	SUPPORT	NO SUPPORT
LINT{DATA|MEMO}	NO SUPPORT	NO SUPPORT
LISDFBASE	SUPPORT	SUPPORT
LISDOBASE	SUPPORT	SUPPORT
LISFREQ	SUPPORT	SUPPORT
LISSLIS1	SUPPORT	SUPPORT
LISSLIS2	SUPPORT	SUPPORT
LISV	SUPPORT	NO SUPPORT
LMAX?	SUPPORT	SUPPORT
LMAXS?	SUPPORT	SUPPORT
LMIN?	SUPPORT	SUPPORT
LMINS?	SUPPORT	SUPPORT
LOGFREQ	SUPPORT	NO SUPPORT
LOGM	SUPPORT	NO SUPPORT
LOGMD	SUPPORT	NO SUPPORT
LOGMP	SUPPORT	NO SUPPORT
LOWLIMI	NO SUPPORT	SUPPORT
MANTRIG	SUPPORT	NO SUPPORT
MARD	SUPPORT	SUPPORT
MARKBUCK	SUPPORT	SUPPORT
MARKCENT	SUPPORT	SUPPORT
MARKCONT	SUPPORT	SUPPORT
MARKCOUP	SUPPORT	SUPPORT
MARKDISC	SUPPORT	SUPPORT
MARKFAUV	SUPPORT	NO SUPPORT
MARKFSTI	SUPPORT	SUPPORT
MARKFVAL	SUPPORT	SUPPORT
MARKL	SUPPORT	SUPPORT
MARKMIDD	SUPPORT	NO SUPPORT
MARKODATA	SUPPORT	SUPPORT
MARKOFF	SUPPORT	SUPPORT
MARKOMEMO	SUPPORT	SUPPORT
MARKPEAD	SUPPORT	NO SUPPORT
MARKREF	SUPPORT	SUPPORT
MARKSPAN	SUPPORT	SUPPORT

COMMAND	87510A	E5100A
MARKSTAR	SUPPORT	SUPPORT
MARKSTIM	SUPPORT	NO SUPPORT
MARKSTOP	SUPPORT	SUPPORT
MARKTIME	SUPPORT	SUPPORT
MARKUNCO	SUPPORT	SUPPORT
MARKZERO	SUPPORT	SUPPORT
MARK{1-8}	SUPPORT	SUPPORT
MAXD?	SUPPORT	NO SUPPORT
MAXPOIN?	NO SUPPORT	SUPPORT
MAXPORT?	NO SUPPORT	SUPPORT
MEAS	SUPPORT	SUPPORT
MEAS1PT?	NO SUPPORT	SUPPORT
MEASA	SUPPORT	SUPPORT
MEASR	SUPPORT	SUPPORT
MEASTAT	SUPPORT	SUPPORT
MENU3	NO SUPPORT	SUPPORT
MODI1	SUPPORT	NO SUPPORT
MONDYEAR	SUPPORT	SUPPORT
MOVADARY	NO SUPPORT	SUPPORT
MULC	NO SUPPORT	SUPPORT
NEGL	SUPPORT	SUPPORT
NEXP	SUPPORT	NO SUPPORT
NEXPK?	SUPPORT	SUPPORT
NEXTNPK?	NO SUPPORT	SUPPORT
NPEAK	NO SUPPORT	SUPPORT
NPEAKLIST?	NO SUPPORT	SUPPORT
NPEAKSORT?	NO SUPPORT	SUPPORT
NUMC	NO SUPPORT	SUPPORT
NUMG	SUPPORT	NO SUPPORT
NUMLMAX?	SUPPORT	SUPPORT
NUMLMIN?	SUPPORT	SUPPORT
NUMLMINMAX?	SUPPORT	NO SUPPORT
OFSD	SUPPORT	NO SUPPORT
OFSL	SUPPORT	NO SUPPORT
OFSZ	SUPPORT	NO SUPPORT
OPEP	SUPPORT	NO SUPPORT
OPTI?	SUPPORT	NO SUPPORT
OSE	SUPPORT	SUPPORT
OSER?	SUPPORT	SUPPORT
OSNT	SUPPORT	SUPPORT
OSPT	SUPPORT	SUPPORT
OSR?	SUPPORT	SUPPORT
OUT1ENVH	SUPPORT	SUPPORT
OUT1ENVL	SUPPORT	SUPPORT
OUT1H	SUPPORT	SUPPORT

COMMAND	87510A	E5100A
OUT1L	SUPPORT	SUPPORT
OUT2ENVH	SUPPORT	SUPPORT
OUT2ENVL	SUPPORT	SUPPORT
OUT2H	SUPPORT	SUPPORT
OUT2L	SUPPORT	SUPPORT
OUT8IO	SUPPORT	SUPPORT
OUTAIO	SUPPORT	SUPPORT
OUTBIO	SUPPORT	SUPPORT
OUTCIO	SUPPORT	SUPPORT
OUTDIO	SUPPORT	SUPPORT
OUTEIO	SUPPORT	SUPPORT
OUTFIO	SUPPORT	SUPPORT
OUTGIO	SUPPORT	SUPPORT
OUTHIO	SUPPORT	SUPPORT
OUTPCALC{01-03}?	SUPPORT	SUPPORT
OUTPCALK?	SUPPORT	NO SUPPORT
OUTPCERR?	SUPPORT	SUPPORT
OUTPCF2	NO SUPPORT	SUPPORT
OUTPCFIL?	SUPPORT	SUPPORT
OUTPDATA?	NO SUPPORT	SUPPORT
OUTPDATAT?	SUPPORT	SUPPORT
OUTPDATM	NO SUPPORT	SUPPORT
OUTPDATTP?	SUPPORT	SUPPORT
OUTPERRO?	SUPPORT	SUPPORT
OUTPFAIP?	SUPPORT	NO SUPPORT
OUTPFBUS?	SUPPORT	NO SUPPORT
OUTPFILT?	SUPPORT	SUPPORT
OUTPFORM?	SUPPORT	SUPPORT
OUTPFORMP?	SUPPORT	SUPPORT
OUTPIFORM?	SUPPORT	SUPPORT
OUTPINP8IO?	SUPPORT	SUPPORT
OUTPINPCIO?	SUPPORT	SUPPORT
OUTPINPDIO?	SUPPORT	SUPPORT
OUTPINPEIO?	SUPPORT	SUPPORT
OUTPIRFORM?	SUPPORT	SUPPORT
OUTPIRTMEM?	SUPPORT	SUPPORT
OUTPITMEM?	SUPPORT	NO SUPPORT
OUTPLIMF?	SUPPORT	NO SUPPORT
OUTPLIML?	SUPPORT	NO SUPPORT
OUTPLIMM?	SUPPORT	NO SUPPORT
OUTPMARK?	SUPPORT	SUPPORT
OUTPMAX?	SUPPORT	SUPPORT
OUTPMEAN?	SUPPORT	SUPPORT
OUTPMEMO?	SUPPORT	SUPPORT
OUTPMEMOT?	SUPPORT	SUPPORT

COMMAND	87510A	E5100A
OUTPMEMTP?	SUPPORT	NO SUPPORT
OUTPMIN?	SUPPORT	SUPPORT
OUTPMINMAX?	SUPPORT	SUPPORT
OUTPMSTA?	SUPPORT	SUPPORT
OUTPMWID?	SUPPORT	SUPPORT
OUTPMWIL?	SUPPORT	SUPPORT
OUTPMWLF?	SUPPORT	SUPPORT
OUTPPEAK?	SUPPORT	NO SUPPORT
OUTPRAW?	NO SUPPORT	SUPPORT
OUTPRESF?	SUPPORT	SUPPORT
OUTPRESO?	SUPPORT	SUPPORT
OUTPRESR?	SUPPORT	SUPPORT
OUTPRFORM?	SUPPORT	SUPPORT
OUTPRFORMP?	NO SUPPORT	SUPPORT
OUTPRTMEM?	SUPPORT	SUPPORT
OUTPRTMEMP?	NO SUPPORT	SUPPORT
OUTPSTIM?	SUPPORT	SUPPORT
OUTPTESS?	SUPPORT	NO SUPPORT
OUTPTITL?	SUPPORT	SUPPORT
OUTPTMEM?	SUPPORT	NO SUPPORT
OUTPTMEMP?	SUPPORT	NO SUPPORT
OUTPTRAC	SUPPORT	SUPPORT
OUTPTRACB	SUPPORT	SUPPORT
OUTPXF2	NO SUPPORT	SUPPORT
OUTPXFIL?	SUPPORT	SUPPORT
PARS	SUPPORT	SUPPORT
PARSMODE	NO SUPPORT	SUPPORT
PARSRANG	NO SUPPORT	SUPPORT
PEADX	SUPPORT	NO SUPPORT
PEADY	SUPPORT	NO SUPPORT
PEAK?	SUPPORT	SUPPORT
PEAKLIST?	NO SUPPORT	SUPPORT
PEAKSORT?	NO SUPPORT	SUPPORT
PHAO	SUPPORT	SUPPORT
PHAS	SUPPORT	SUPPORT
PICIRC	NO SUPPORT	SUPPORT
PLOALL	NO SUPPORT	NO SUPPORT
PLOC	NO SUPPORT	NO SUPPORT
PLODONLY	NO SUPPORT	NO SUPPORT
PLOGRAT	NO SUPPORT	NO SUPPORT
PLOS	NO SUPPORT	NO SUPPORT
PLOT	NO SUPPORT	NO SUPPORT
POIN	SUPPORT	SUPPORT
POLA	SUPPORT	NO SUPPORT
POLE?	SUPPORT	SUPPORT

COMMAND	87510A	E5100A
POLM	SUPPORT	NO SUPPORT
POLMLIN	SUPPORT	NO SUPPORT
POLMLOG	SUPPORT	NO SUPPORT
POLMRI	SUPPORT	NO SUPPORT
PORE	SUPPORT	NO SUPPORT
PORTA	SUPPORT	NO SUPPORT
PORTR	SUPPORT	NO SUPPORT
POSL	SUPPORT	SUPPORT
POWDAUTO	SUPPORT	NO SUPPORT
POWDMANU	SUPPORT	NO SUPPORT
POWDVALU	SUPPORT	NO SUPPORT
POWE	SUPPORT	SUPPORT
POWF	SUPPORT	NO SUPPORT
POWL	SUPPORT	NO SUPPORT
POWO?	SUPPORT	NO SUPPORT
POWS	SUPPORT	SUPPORT
POWU	NO SUPPORT	SUPPORT
PREP	SUPPORT	NO SUPPORT
PRES	SUPPORT	SUPPORT
PRINALL	SUPPORT	SUPPORT
PRIR	NO SUPPORT	SUPPORT
PSOFT	NO SUPPORT	NO SUPPORT
PTABORT	NO SUPPORT	SUPPORT
PTFOVHD	SUPPORT	SUPPORT
PTFR	SUPPORT	SUPPORT
PTFRBW	SUPPORT	NO SUPPORT
PTFRSR	SUPPORT	SUPPORT
PTPARA	NO SUPPORT	SUPPORT
PTPMNUM	SUPPORT	NO SUPPORT
PTRACK	SUPPORT	SUPPORT
PTREPN	NO SUPPORT	SUPPORT
PTRUPD	SUPPORT	NO SUPPORT
PTSTAT	SUPPORT	SUPPORT
PTTRGLMT	SUPPORT	SUPPORT
PTTRGPHS	SUPPORT	SUPPORT
PURG	SUPPORT	SUPPORT
QUAD	NO SUPPORT	NO SUPPORT
RAID	SUPPORT	SUPPORT
RAIISOL	SUPPORT	SUPPORT
RAIRESP	SUPPORT	SUPPORT
REAL	SUPPORT	NO SUPPORT
RECCOFF	SUPPORT	NO SUPPORT
RECCON	SUPPORT	NO SUPPORT
RECD	SUPPORT	SUPPORT
REFP	SUPPORT	SUPPORT

COMMAND	87510A	E5100A
REFV	SUPPORT	SUPPORT
RESAVD	SUPPORT	SUPPORT
RESC	SUPPORT	NO SUPPORT
RESD	SUPPORT	NO SUPPORT
RESPDONE	SUPPORT	SUPPORT
REST	SUPPORT	SUPPORT
RFOPNORM	SUPPORT	NO SUPPORT
RFOPOPEN	SUPPORT	NO SUPPORT
RIGL	NO SUPPORT	NO SUPPORT
RIGU	NO SUPPORT	NO SUPPORT
RPLENV?	SUPPORT	SUPPORT
RPLHEI?	SUPPORT	SUPPORT
RPLLHEI?	SUPPORT	SUPPORT
RPLMEA?	SUPPORT	SUPPORT
RPLMM	NO SUPPORT	SUPPORT
RPLPP?	SUPPORT	SUPPORT
RPLPPS?	SUPPORT	SUPPORT
RPLRHEI?	SUPPORT	SUPPORT
RPLVAL?	SUPPORT	SUPPORT
SADD	SUPPORT	SUPPORT
SAV1	SUPPORT	SUPPORT
SAVC	SUPPORT	NO SUPPORT
SAVCA	SUPPORT	SUPPORT
SAVDA	NO SUPPORT	SUPPORT
SAVDALL	SUPPORT	SUPPORT
SAVDASC	SUPPORT	SUPPORT
SAVDDAT	SUPPORT	SUPPORT
SAVDGRA	NO SUPPORT	NO SUPPORT
SAVDGRAP	NO SUPPORT	SUPPORT
SAVDMNU3	NO SUPPORT	SUPPORT
SAVDSTA	SUPPORT	SUPPORT
SAVEUSEK	SUPPORT	NO SUPPORT
SAVFA	NO SUPPORT	SUPPORT
SAVMA	NO SUPPORT	SUPPORT
SAVRA	NO SUPPORT	SUPPORT
SAVTA	SUPPORT	SUPPORT
SAVTMA	SUPPORT	SUPPORT
SCAC	SUPPORT	NO SUPPORT
SCAFDATA	SUPPORT	SUPPORT
SCAFMEMO	SUPPORT	SUPPORT
SCAL	SUPPORT	SUPPORT
SCAPFULL	NO SUPPORT	NO SUPPORT
SCAPGL	NO SUPPORT	NO SUPPORT
SCAPGU	NO SUPPORT	NO SUPPORT
SCAU	SUPPORT	NO SUPPORT

COMMAND	87510A	E5100A
SCAY	NO SUPPORT	SUPPORT
SDEL	SUPPORT	SUPPORT
SDON	SUPPORT	SUPPORT
SEAL	SUPPORT	SUPPORT
SEALMAX	SUPPORT	SUPPORT
SEALMIN	SUPPORT	SUPPORT
SEAM	SUPPORT	SUPPORT
SEAMAX	SUPPORT	SUPPORT
SEAMEAN	SUPPORT	SUPPORT
SEAMIN	SUPPORT	SUPPORT
SEAOFF	SUPPORT	SUPPORT
SEAPPEAK	SUPPORT	SUPPORT
SEAR	SUPPORT	SUPPORT
SEARSTOR	SUPPORT	SUPPORT
SEATARG	SUPPORT	SUPPORT
SEDI	SUPPORT	SUPPORT
SEET	SUPPORT	SUPPORT
SELD	SUPPORT	NO SUPPORT
SERM?	SUPPORT	NO SUPPORT
SET1PT	NO SUPPORT	SUPPORT
SETCDATE	SUPPORT	SUPPORT
SETCTIME	SUPPORT	SUPPORT
SETZ	SUPPORT	SUPPORT
SINDTMEM	SUPPORT	NO SUPPORT
SING	SUPPORT	SUPPORT
SINSPEAK	SUPPORT	SUPPORT
SMOO	SUPPORT	SUPPORT
SMOOAPER	SUPPORT	SUPPORT
SOUCOFF	SUPPORT	NO SUPPORT
SOUCON	SUPPORT	NO SUPPORT
SPAN	SUPPORT	SUPPORT
SPECRESI	SUPPORT	NO SUPPORT
SPECRESP	SUPPORT	NO SUPPORT
SPECS11A	SUPPORT	NO SUPPORT
SPECS11B	SUPPORT	NO SUPPORT
SPECS11C	SUPPORT	NO SUPPORT
SPLD	SUPPORT	SUPPORT
SRCHFR	NO SUPPORT	SUPPORT
STAN	SUPPORT	NO SUPPORT
STAR	SUPPORT	SUPPORT
STAS	SUPPORT	SUPPORT
STAW	NO SUPPORT	SUPPORT
STDD	SUPPORT	NO SUPPORT
STDT	SUPPORT	NO SUPPORT
STDTARBI	SUPPORT	NO SUPPORT

COMMAND	87510A	E5100A
STDTDELA	SUPPORT	NO SUPPORT
STDTLOAD	SUPPORT	NO SUPPORT
STDTOPEN	SUPPORT	NO SUPPORT
STDTSHOR	SUPPORT	NO SUPPORT
STED	NO SUPPORT	SUPPORT
STIDROUT{1-16}	SUPPORT	SUPPORT
STIMROUT{1-16}	SUPPORT	SUPPORT
STOD{DISK|MEMO}	SUPPORT	SUPPORT
STOP	SUPPORT	SUPPORT
STPSIZE	SUPPORT	SUPPORT
STR	NO SUPPORT	SUPPORT
SWET	SUPPORT	SUPPORT
SWETAUTO	SUPPORT	SUPPORT
SWPT	SUPPORT	SUPPORT
TARL?	SUPPORT	SUPPORT
TARLRP?	SUPPORT	NO SUPPORT
TARSUBL?	NO SUPPORT	SUPPORT
TARSUBR?	NO SUPPORT	SUPPORT
TARR?	SUPPORT	SUPPORT
TERI	SUPPORT	NO SUPPORT
TESC	SUPPORT	NO SUPPORT
TEST	SUPPORT	NO SUPPORT
THRR	SUPPORT	SUPPORT
TIMO	SUPPORT	SUPPORT
TITL	SUPPORT	SUPPORT
TOTIME	SUPPORT	SUPPORT
TRABGE	NO SUPPORT	SUPPORT
TRABLE	NO SUPPORT	SUPPORT
TRACK	SUPPORT	SUPPORT
TRAFDATA	SUPPORT	SUPPORT
TRAFMEMO	SUPPORT	SUPPORT
TRAP	SUPPORT	SUPPORT
TRAR	SUPPORT	SUPPORT
TRIGMEAS	NO SUPPORT	SUPPORT
TRIM	NO SUPPORT	SUPPORT
UPPD	SUPPORT	SUPPORT
UPPELIMI	NO SUPPORT	SUPPORT
VELOFACT	SUPPORT	NO SUPPORT
WIDSIN	SUPPORT	NO SUPPORT
WIDSOUT	SUPPORT	NO SUPPORT
WIDT	SUPPORT	SUPPORT
WIDV	SUPPORT	SUPPORT
WRIT16	SUPPORT	SUPPORT
WRIT24	SUPPORT	SUPPORT

F

Keyword Guide to Porting

The following sections summarize the differences in the GPIB commands of the 87510A and the E5100A/B. This appendix is intended only as a quick reference to the keywords and their compatibility. For detailed information, see the Chapter 8.

The following table is summary of this appendix

87510A	E5100A
Trigger	
SRQ	SING?
SING	TRIM SING
Calibration	
*OPC?	STANC?, RAIRESP?, RAIISO?, CLASS11[A|B|C]?
*OPC?	RESPDONE, RAID? , SAV1
Data Transfer	
FORM2, FORM3, FORM5	No support for data input. (use FORM4)
Meas/Format	
AR, MEASA, MEASR	Full support
LOGM, PHAS, DELA, LINM, LOGMP, LOGMD, REAL, IMAG	Full support
EXPP	FMT EXPP
POLAR	No support
CONVZTRA, CONVZREF, CONVYTRA, CONVYREF	ANAMODE ZREFL, ANAMODE YREFL
Analysis Function	
BINSLINE, BINS, BINOA, BINOB, EDITBINL	No support
EDITLIS1, EDITLIS2	UPELIMI, LOWELIMI
Other	
KEY	No support

Trigger Function

Single Trigger (SRQ)

Use the `SING?` command instead of `SRQ` to generate a single trigger. `SING?` initiates a single measurement sweep and returns 1 when the sweep is completed.

```
OUTPUT @E5100;"SING?"
ENTER @E5100;Tmp$
```

Single Trigger Mode (SING)

Use the `TRIM SING` command instead of `SING` to select a single trigger as the trigger mode.

Calibration

Calibration Measurement (OPC)

Use the `STANC?`, `RAIRESP?`, `RAIISO?`, `CLASS11[A|B|C]?` query commands instead of the `OPC` command (which is used to detect that calibration measurement is completed). These query commands make a calibration measurement and return 1 when the calibration measurement is completed.

Calibration Done (OPC)

Use the `RESPDONE?`, `RAID?` and `SAV1` query commands instead of the `OPC` command (which is used to detect that the calibration coefficient calculation is completed). These query commands calculate and save the calibration coefficient into the internal array and return 1 when the operation is completed.

Data Transfer Command

Binary Transfer Commands (FORM2, FORM3, FORM4)

Use ASCII transfer (`FORM4`) when data is entered from an external controller to the E5100A/B. The E5100A/B does not support binary transfer for data input.

Input Port (AR, MEASA, MEASR)

The following commands are used to select an input port for the 87510A. The AR, MEASA, and MEASR commands are supported (see the following table)

Table F-1.

Equivalent Softkey	87510A	E5100A
A/R	MEAS AR or AR	MEAS AR or AR
B/R	-	MEAS BR
C/R	-	MEAS CR
B/A	-	MEAS BA
C/A	-	MEAS CA
R/A	-	MEAS RA
A/B	-	MEAS AB
C/B	-	MEAS CB
R/B	-	MEAS RB
A/C	-	MEAS AC
B/C	-	MEAS BC
R/C	-	MEAS RC
R	MEAS R or MEASR	MEAS R or MEASR
A	MEAS A or MEASA	MEAS A or MEASA
B	-	MEAS B
C	-	MEAS C

Use the MEAS? query command to confirm the current port.

```
OUTPUT @E5100;MEAS?
ENTER @E5100; Port$
```

Format (LOGM, PHAS, DELA, LINM, EXPP, LOGMP, LOGMD, REAL, IMAG, POLAR)

The following commands are supported :

LOGM, PHAS, DELA, LINM, LOGMP, LOGMD, REAL, IMAG

Use the FMT EXPP command instead of EXPP for selecting Expanded Phase format. EXPP is not supported.

Polar chart format (POLAR) is not supported. (Refer to the following table.)

Table F-2.

Equivalent Softkey	87510A	E5100A
LOG MAG & PHASE	FMT LOGMP or LOGMP	FMT LOGMP or LOGMP
LOG MAG & DELAY	FMT LOGMD or LOGMD	FMT LOGMD or LOGMD
LIN MAG & PHASE	-	FMT LINMP
LIN MAG & DELAY	-	FMT LINMD
REAL & IMAGINARY	-	FMT RIMAG
LOG MAG	FMT LOGM or LOGM	FMT LOGM or LOGM
LIN MAG	FMT LINM or LINM	FMT LINM or LINM
PHASE	FMT PHAS or PHAS	FMT PHAS or PHAS
DELAY	FMT PHAS or PHAS	FMT PHAS or PHAS
REAL	FMT REAL or REAL	FMT REAL or REAL
IMAGINARY	FMT IMAG or IMAG	FMT IMAG or IMAG
EXPANDED PHASE	FMT EXPP or EXPP	FMT EXPP (EXPP is not supported)
|Z| & PHASE z	-	FMT MAGZP
|Y| & PHASE y	-	FMT MAGYP
R-X	-	FMT IMPRX
G-B	-	FMT ADMGB
|Z|	-	FMT MAGZ
|Y|	-	FMT MAGY
PHASE z	-	FMT PHAZ
PHASE y	-	FMT PHAY
R	-	FMT IMPR
G	-	FMT ADMG
B	-	FMT ADMB
X	-	FMT IMPX
POLAR	POLA or FMT POLA	No support

Use the FMT? query command to confirm the current format.

```
OUTPUT @E5100;FMT?
ENTER @E5100; Format$
```

Impedance Conversion Function (CONVZTRA, CONVZREF, CONVYTRA, CONVYREF)

Use the impedance measurement mode ANAMODE instead of the conversion function, even though the E5100A/B supports the CONV commands.

Table F-3.

Impedance Conversion (87510A)	Impedance Measurement Mode (E5100A)
CONVZTRA	ANAMODE ZTRAN
CONVZREF	ANAMODE ZREFL
CONVYTRA	ANAMODE ZTRAN;FMT MAGY
CONVYREF	ANAMODE ZREFL;FMT MAGY

Analysis Commands

BIN sort (BINSLINE, BINS, BINOA, BINOB, EDITBINL)

The E5100A/B does not support the BIN sort function.

Limit Line Edit (EDITLIS1, EDITLIS2)

Use `UPELIMI` and `LOWELIMI` commands to define a limit line. The E5100A/B does not support the `EDITLIS1` and `EDITLIS2` commands of the 87510A.

Other Commands

KEY command

The E5100A/B dose not support the `KEY` command.

G

Option 022

Overview

This option enables you to control metal deposition during a deposition process of crystal resonators.

Metal deposition is used during a manufacturing to adjust the frequency of a crystal resonator. The resonant frequency of a crystal resonator becomes lower as deposition proceeds. E5100A/B OPT. 022, as described in Figure G-1, monitors the phase at a specified frequency f_1 and outputs preset data to the I/O port when the measured phase goes under 0 (or any value you specify as you wish).

E5100A/B OPT. 022, then, moves its focus to the next frequency f_2 and performs another measurement. When the measured phase reaches the specified value again, it outputs different data to the I/O port. This output to the I/O port controls the amount of metal deposition to achieve efficient and accurate metal deposition. This option's function is called the Trap Function because it awaits the phase to match the preset phase at a certain frequency.

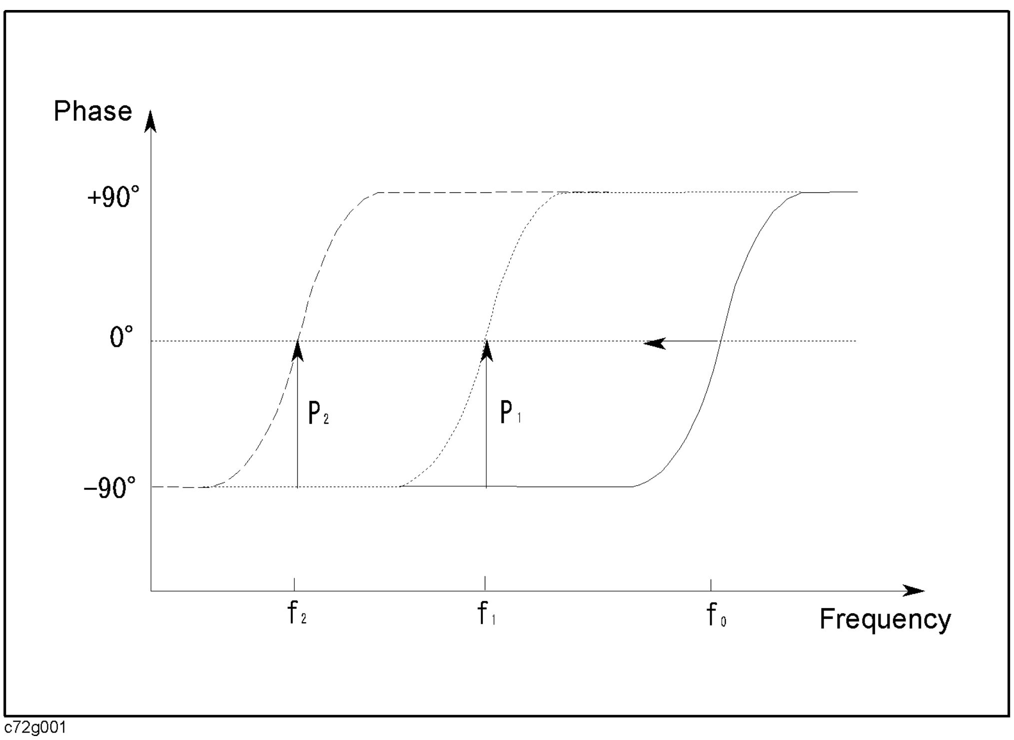


Figure G-1. The Trap Function

Using E5100A/B Option 022

You must use a program to use the trap function to control a deposition process.

A sample program is provided in the sample program disk. This chapter explains how to measure a device using the sample program.

The sample program for the trap function is prepared under the following conditions.

- A π network test fixture is used.
- Characteristic impedance is set at 12.5Ω to support standard π network test fixture.

This sample program measures a device using the trap function on the channel 1. When the program sees the resonant frequency of the resonator matching the final trap frequency, it switches the focus to the channel 2 to perform a frequency sweep and verifies the resonant frequency.

E5100A/B option 022 starts performing the trap function from the measured point at the right of the screen. As deposition proceeds, the resonant frequency of resonator becomes lower and the measured frequency goes from right to left. This is because the function waits for the measured phase to go under 0°(actually the phase value specified by `INPUTRAC`) at the measured point so that it can output certain data to the I/O port and move to the next measured point to repeat the process.

The more detailed description of the program function is discussed for the sample program with its default setup. The default setup is completed between lines 1600 and 1910 in "The Sample Program List". Please refer to this section for the setup. Figure G-2 describes the status of the channel 1 with this setup.

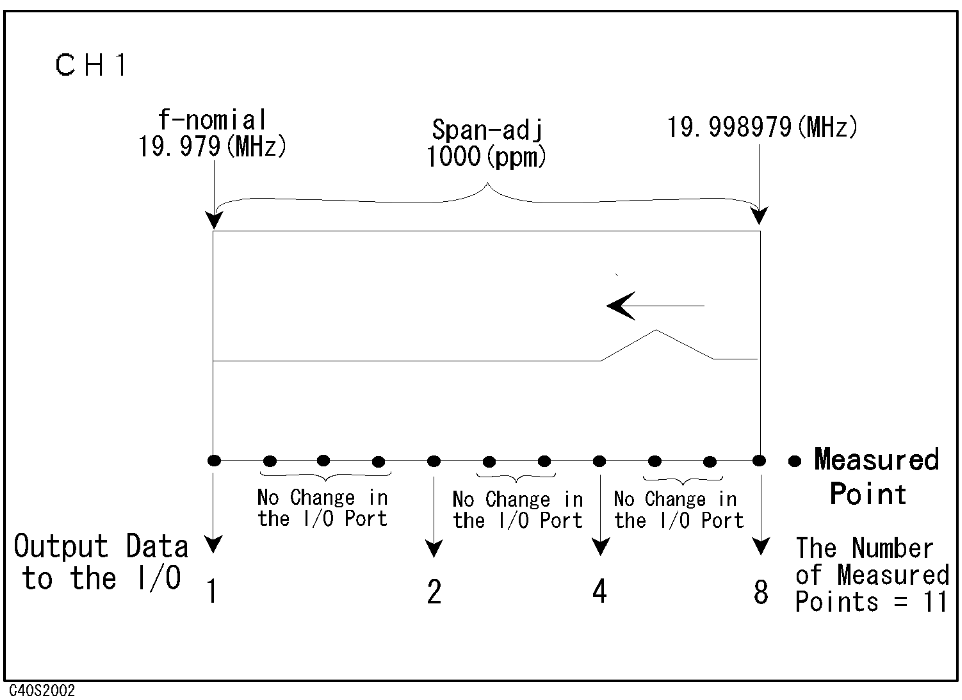


Figure G-2. How E5100A/B option 022 Works with the Default Setup

An Example of Trapping

1. The leftmost frequency on the screen is the stop frequency for the trap function. (19.979MHz)
2. The rightmost frequency on the screen is the start frequency for the trap function. (19.998979 MHz$=$19.979$\times$(1$+$1000E$-$6))
3. A phase is continuously measured at the rightmost frequency on the screen.
4. When deposition proceeds and the phase becomes 0, E5100A/B option 022 outputs 8 to the I/O port.
5. E5100A/B option 022 moves the measured point to the second point from the right (it changes the measured frequency) and measures the phase.
6. Since the output data is not specified for this I/O port, the output data to the I/O port remain unchanged when the phase becomes 0.
7. E5100A/B option 022 goes to the third measured point from the right to perform the trap function.
8. At the 4th measured point from the right, E5100A/B option 022 outputs 4 to the I/O port.
9. E5100A/B option 022 continues to measure at the rest of measured points until it reaches the stop frequency for the trap function.

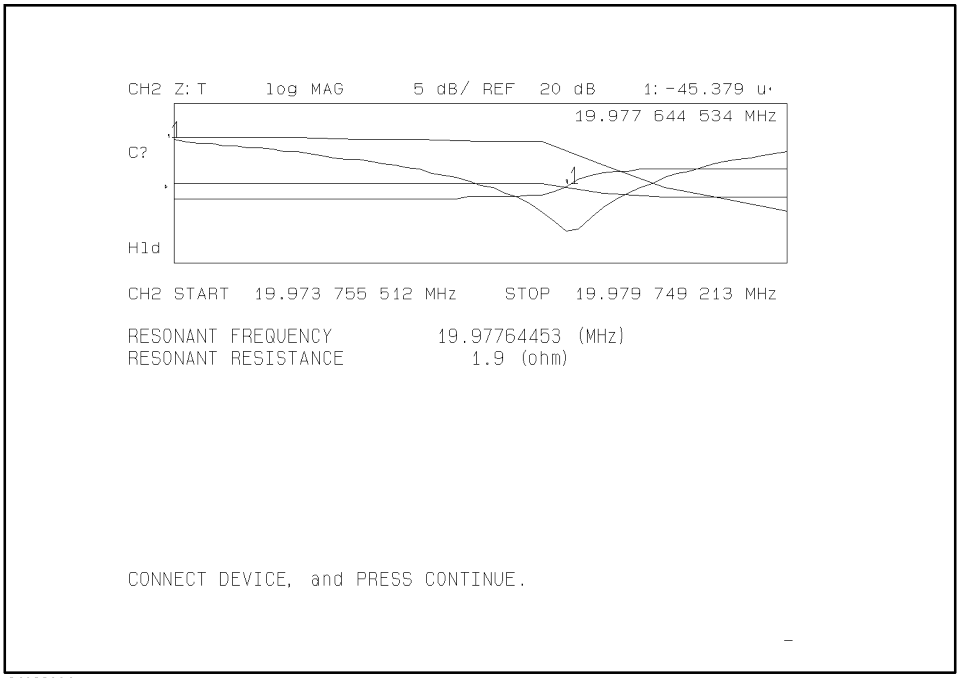
Note

You can specify at which measured point and what data to output to the I/O port

---

As soon as E5100A/B option 022 completes the trap function on the channel 1, it performs a normal frequency sweep on the channel 2 to verify the resonant point. When E5100A/B option 022 completes the sweep, it calculates the resonant frequency and resonant impedance and displays them at the lower half of the screen. (Figure G-3)

There are 4 waveforms on the screen because the measurement screen at the upper half of the screen displays the waveforms of the channel 1 over those of the channel 2.



C40S2004

Figure G-3. The Displayed Measurement Result

Required Parameters

Following parameters are required to execute this program.

- Parameters Required for the Trap Function (Channel 1)
  - □ The stop frequency for the trap function
  - □ The start frequency for the trap function (Input in the ratio (ppm) to the stop frequency for the trap function)
  - □ The IF bandwidth
  - □ The number of measured points
  - □ The phase value to trap at each measured point and data to output when the value matches
- Parameters Specified for a Frequency Sweep (Channel 2)
  - □ The frequency sweep span
  - □ The IF bandwidth
  - □ The number of measured points
  - □ The wait time interval between the completion of the trap function and the start of frequency sweep.

The Sample Program List

This program is contained in the attached sample disk for option 022/023 with the file name **SMPL022.BAS**.

```
490    ASSIGN @E5100 TO 800
500    CLEAR @E5100
510   !
520    DIM Bin(1:20)
530    DIM Dat$[80],Lw$[20],Err$[50],Lmt_flag$(1:20)[5]
540  !
550    OUTPUT @E5100;"PRES"
560    GOSUB Setting
570    GOSUB Power_setup
580    GOSUB Set_ch1
590    GOSUB Pi_cal
591    OUTPUT @E5100;"CALCOPY 1,2"
600    CLEAR SCREEN
610    GOSUB Set_ch1_part2
620    OUTPUT @E5100;"COUC OFF"
630    GOSUB Setup_monitor
640    GOSUB Set_ch2
650    GOSUB Measurement
660  !
670    STOP
680  !
690  !
700  !
710   !  ************  SURBROUTINES  ****************
720   !
730   !  ********* DEFAULT VALUES ********
740   !
750 Default_set:  !
760    Z0=12.5
770    F_nominal=19.979  ! (MHz)
780    Span_adj=1000  ! (ppm)
790    Span_fr=100    ! (ppm)
800    Ifbw1=1000     ! (Hz)
810    Ifbw2=1000     ! (Hz)
820    Nop1=11        ! NOP for CH1
830    Nop2=201        ! NOP for CH2
840    Wait_check=100  ! WAIT TIME for Fr check after EVAPOLATION
850    RETURN
860   !
870   !
880 Default_power:  !
890    Power=10           ! POWER = 10 (uW)
900    R=10             ! CI = 10 (ohm)
910    RETURN
920    !
930    !
940 Lmt_flag:  !
950   !
960   !     Lmt,Flag
970   ! ----------------
980   DATA "0,ON","0000000000000001"  !  1
990   DATA "0,OFF"
1000  DATA "0,OFF"
1010  DATA "0,OFF"
1020  DATA "0,ON","0000000000000010"  !  2
1030  DATA "0,OFF"
1040  DATA "0,OFF"
1050  DATA "0,ON","0000000000000100"  !  4
1060  DATA "0,OFF"
1070  DATA "0,OFF"
```

```
1080  DATA "0,ON","0000000000001000"  !  8
1090 !
1100 !
1110 !
1120 !  ********************************
1130 !
1140 Setting: !
1150 !
1160  GOSUB Default_set
1170 ! OUTPUT @E5100;"OSE 16384;OSNT 16384;*SRE 128"
     ! SET INTERRUPT for DATA TRANSFER between NA and EXT PC
1180  OUTPUT @E5100;"*CLS"
1190  OUTPUT @E5100;"DISAHIHB"
1200  OUTPUT @E5100;"CHAN1;HOLD"
1210  OUTPUT @E5100;"TRAP OFF"
1220  OUTPUT @E5100;"CHAN2;HOLD"
1230  OUTPUT @E5100;"TRAP OFF"
1240  OUTPUT @E5100;"CHAN1"
1250  OUTPUT @E5100;"MULC ON"
1260  OUTPUT @E5100;"COUC ON"
1270  OUTPUT @E5100;"SWPT LINF"
1280  INPUT "Nominal frequency (MHz) ?",F_nominal
1290  F_nominal=F_nominal*1.E+6
1300  INPUT "Adjustment Span (ppm) ?",Span_adj
1310  Span_adj=Span_adj/1.E+6
1320  Span_adj=F_nominal*Span_adj
1330  Startf=F_nominal
1340  Stopf=F_nominal+Span_adj
1350  INPUT "Span for Fr check (ppm) ?",Span_fr
1360  Span_fr=Span_fr/1.E+6
1370  Span_fr=F_nominal*Span_fr
1380  INPUT "IFBW at CH1 (Hz) ?",Ifbw1
1390  INPUT "IFBW at CH2 (Hz) ?",Ifbw2
1400 ! INPUT "NOP for EVAPOLATION at CH1 ? ",Nop1
1410  INPUT "NOP for Fr check at CH2 ? ",Nop2
1420  INPUT "WAIT TIME after EVAPOLATION (msec) ?",Wait_check
1430  Wait_check=Wait_check/1000
1440  EXECUTE "ANAOCH2"
1450  EXECUTE "ANAODATA"
1460  EXECUTE "ANARFULL"
1470  RETURN
1480 !
1490 !
1500 Power_setup: !
1510  GOSUB Default_power
1520  OUTPUT @E5100;"POWU WATT"
1530  OUTPUT @E5100;"PICIRC ON"
1540  INPUT "POWER (uW)",Power
1550  INPUT "RESONANT RESISTANCE (ohm) ",R
1560  Power=Power*1.E-6
1570   !
1580 !
1590 !
1600 Set_ch1:!
1610  OUTPUT @E5100;"CHAN1;ANAMODE ZTRAN;MEAS AR;FMT MAGZP"
1620  OUTPUT @E5100;"HOLD"
1630  OUTPUT @E5100;"CIVAL ";R
1640  OUTPUT @E5100;"POWE ";Power
1650  OUTPUT @E5100;"STAR ";Startf-(Span_fr)*(7/8)  ! SET START FREQ for CAL
1660  OUTPUT @E5100;"STOP ";Stopf
1670  OUTPUT @E5100;"POIN ";Nop1
1680  OUTPUT @E5100;"IFBW ";Ifbw1
1690  OUTPUT @E5100;";DISG OFF"
1700  OUTPUT @E5100;";SWED DOWN"
1710  RETURN
1720 !
```

```
1730 !
1740 Set_ch1_part2:!
1750  OUTPUT @E5100;"STAR ";Startf
1760  RETURN
1770 !
1780 !
1790 Set_ch2:!
1800  OUTPUT @E5100;"CHAN2;ANAMODE ZTRAN;MEAS AR;FMT MAGZP"
1810  OUTPUT @E5100;"HOLD"
1820  ! OUTPUT @E5100;"CIVAL ";R
1830  ! OUTPUT @E5100;"POWE ";Power
1840  ! OUTPUT @E5100;"LINFREQ"
1850  OUTPUT @E5100;"STOP ";Startf+(Span_fr)*(1/8)
      ! SET STOP FREQ for CH2 based on START FREQ at CH1
1860  OUTPUT @E5100;"STAR ";Startf-(Span_fr)*(7/8)
1870  OUTPUT @E5100;"POIN ";Nop2
1880  OUTPUT @E5100;"IFBW ";Ifbw2
1890  OUTPUT @E5100;";DISG OFF"
1900  OUTPUT @E5100;"SETZ ";Z0
1910  RETURN
1920 !
1930 !
1940 !     ********** MONITOR FUNCTION SETUP ***********
1950 !
1960 Setup_monitor:!          MONITOR function setip
1970 !
1980  Start_pt=1              ! Set Start Point
1990  Stop_pt=Nop1            ! Set Stop Point
2000 !
2010  OUTPUT @E5100;"WRIT16" !  Set the Number of PIN of I/O Port
2020  !
2030  RESTORE Lmt_flag
2040  !
2050  FOR I=Start_pt TO Stop_pt
2060    READ Lmt_flag$(I)
2070    OUTPUT @E5100;"INPUTRAC "&VAL$(I)&","&Lmt_flag$(I)
        ! Set I/O output Points
2080    IF Lmt_flag$(I)[3;2]="ON" THEN
2090      READ Lw$
2100      Bin(I)=DVAL(Lw$,2)
2110      OUTPUT @E5100;"INPUTRACB "&VAL$(I)&",1,";Bin(I)
          ! Set I/O output info
2120    ELSE
2130      OUTPUT @E5100;"INPUTRACB "&VAL$(I)&",1,0"
2140    END IF
2150  NEXT I
2160 !
2161  OUTPUT @E5100;"TRABGE 1"
2170  OUTPUT @E5100;"TRAR ";Start_pt,Stop_pt
2180  OUTPUT @E5100;"TRAFMEMO"
2190  OUTPUT @E5100;"TRAP ON"
2200  !
2210  RETURN
2220  !
2230  !
2240  !  ********* MEASUREMENT ***********
2250  !
2260 Measurement:!
2270  !
2280  LOOP
2290    DISP "CONNECT DEVICE, and PRESS CONTINUE."
2300    PAUSE
2310    DISP "MEASURING"
2320    !
2330    OUTPUT @E5100;"UPDD OFF"
2340    OUTPUT @E5100;"CHAN1"
```

```
2350    OUTPUT @E5100;"UPDD ON"
2360    EXECUTE "SING"
2370    !
2380    WAIT Wait_check
2390    !
2400    OUTPUT @E5100;"UPDD OFF"
2410    OUTPUT @E5100;"CHAN2"
2420    OUTPUT @E5100;"UPDD ON"
2430    EXECUTE "SING"
2440    !
2450    GOSUB Analysis
2460    GOSUB Printing
2470  END LOOP
2480    !
2490  RETURN
2500    !
2510    !  ***************************************
2520    !
2530 Analysis: !
2540  EXECUTE "OUTPRESO?"
2550  Ci=READIO(8,0)
2560  Fr=READIO(8,1)
2570  RETURN
2580  !
2590  !
2600 Printing:  !
2610  CLEAR SCREEN
2620  PRINT USING "25A,4D.8D,6A";"RESONANT FREQUENCY  ";Fr/1.E+6;" (MHz)"
2630  PRINT USING "25A,5X,6D.D,6A";"RESONANT RESISTANCE ";Ci;" (ohm)"
2640  RETURN
2650 !
2660 !
2670 !
2680 !   *************** CALIBRATION *******************
2690 !
2700 Pi_cal: !
2710    !
2720  OUTPUT @E5100;";DISAHIHB"
4480  !
4811  DISP "Leave fixture open,then press 'continue'."
4812  PAUSE
4820  OUTPUT @E5100;"CLASS11A;*OPC?"
4830  ENTER @E5100;Opc
4860  !
4871  DISP "Connect short bar ,then press 'continue'."
4872  PAUSE
4880  OUTPUT @E5100;"CLASS11B;*OPC?"
4890  ENTER @E5100;Opc
4920  !
4931  DISP "Connect load,then press 'continue'."
4932  PAUSE
4940  OUTPUT @E5100;"CLASS11C;*OPC?"
4950  ENTER @E5100;Opc
4980  !
5070  OUTPUT @E5100;"SAV1"
5080  DISP ""
5090  RETURN
5100  !
5190  END
```

GPIB Commands of Option 022

BINSIZE␣<*value*>

Specify the number of continuous data outputted to the I/O port by `INPUTRACB`.

Specify the maximum value in the second parameter of `INPUTRACB`. (Option 022 only)

Parameter Range	0 to 6
Query Response	{*value*}

Examples	`OUTPUT @E5100;"BINSIZE 2"`
	
	`OUTPUT @E5100;"BINSIZE?"`
	`ENTER @E5100;A`

INPUTRAC␣<*value1*>,<*value2*>,<*value3*>

Pass the phase value to trap at a measured point and specify whether data is outputted to the I/O port or not when the measured value reaches the phase value. The data to be outputted to the I/O port is specified by `INPUTRACB`. (No Query,Option 022 only)

Parameter Description	<*value1*> : The measured point number <*value2*> : The trap phase value <*value3*> : I/O output ON/OFF
Examples	`OUTPUT @E5100;"INPUTRAC 10, 0, ON"`

INPUTRACB␣<*value1*>,<*value2*>,<*value3*>

Pass the data outputted to the I/O port when the measured value of phase reaches the limit value specified by the `INPUTRAC` command. Multiple data up to 6 data can be outputted continuously to the I/O port. (No Query,Option 022 only)

Parameter Description	<*value1*> : The measured point number <*value2*> : The order of output I/O data (1 to 6) <*value3*> : Data Query Response

Examples	`OUTPUT @E5100;"INPUTRACB 5, 1, ";DVAL("1010",2)`
	`OUTPUT @E5100;"INPUTRACB 5, 2, ";DVAL("0100",2)`
	`OUTPUT @E5100;"INPUTRACB 5, 3, ";DVAL("0000",2)`

If E5100A/B OPT. 022 is set up by the above commands, the data described in the following figure will be outputted to the I/O port when the measured phase reaches the target value at the measured point 5.

INPUTRACB␣<*value1*>,<*value2*>,<*value3*>

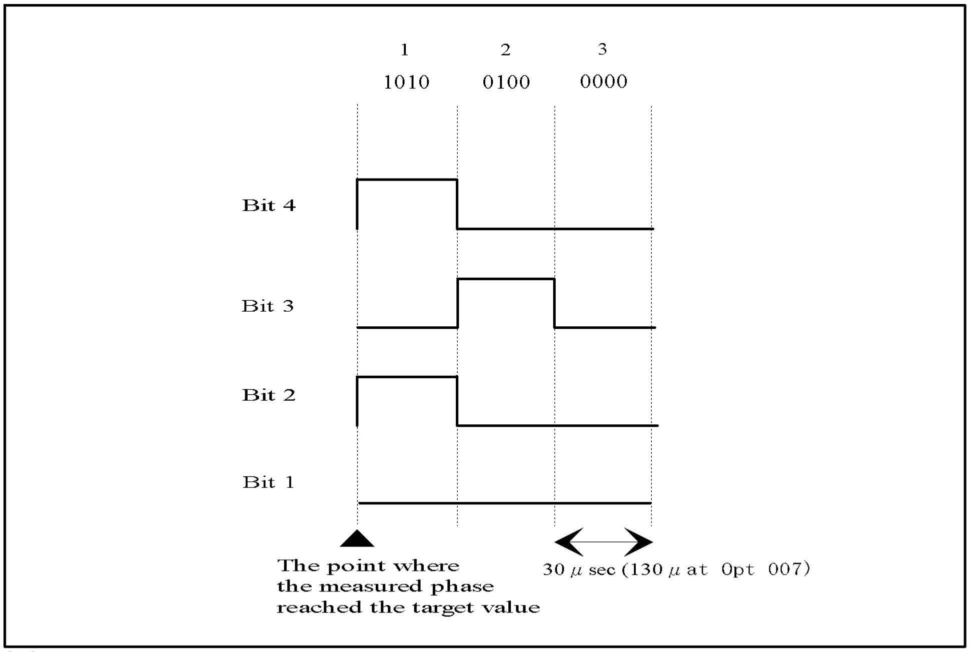


Figure G-4. The Data Output Timing Chart

MEAS1PT? <*value*>

Outputs measurement value at the point number specified by the parameter. When the trap function is turn on (`TRAP ON`), `MEAS1PT` waits to start a measurement until the phase value is in the condition specified by `INPUTRAC` command, and returns the query response.

Parameter Range	1 to NOP
Query Response	{*value*}

Examples	`OUTPUT @E5100;"MEAS1PT? 10000000"`
	`OUTPUT @E5100;"MEAS1PT?"`
	`ENTER @E5100;A`

OUTPTRAC?␣<value>

Outputs the phase value to trap at a measured point and the setting of I/O port at the point specified by the parameter. The phase value and the I/O port setting are specified by `INPUTRAC` command. (Query only, Option 022 only)

Parameter Description	{*value*} : measurement point number
Query Response	{*value*} {0|1} {*value*} : Phase value {1|0} : I/O port setting (ON/OFF)
Examples	`OUTPUT @E5100;"OUTPTRAC? 10"` `ENTER @E5100;A,B`

OUTPTRACB?␣<value1>,<value2>

Outputs the data outputted to the I/O port when the measured value of phase reaches the limit value. The data outputted to the I/O port is specified by `INPUTRACB` command. (Query only,Option 022 only)

Parameter Description	*<value1>* : Measurement point number *<value2>* : Output port number (1 to 6)
Query Response	{*value*} {*value*} : Data
Examples	`OUTPUT @E5100;"INPUTRACB? 10,1"` `ENTER @E5100;A`

TIMO␣{ON|OFF|0|1}

Sets the time limit for the trapping on/off. (Option 022 only)

Parameter Description	`ON` or `1` : ON `OFF` or `0` : OFF
Query Response	{1|0}

Examples	`OUTPUT @E5100;"TIMO ON"`
	
	`OUTPUT @E5100;"TIMO?"`
	`ENTER @E5100;A`

TOTIME␣<value>

Sets the limit time for the trapping. (Option 022 only)

Parameter Range	0 to
Query Response	{*value*} (ms)

Examples	`OUTPUT @E5100;"TOTIME 2000"`
	
	`OUTPUT @E5100;"TOTIME?"`
	`ENTER @E5100;A`

TRABGE

Sets the condition so that the measurement is advanced to the next point when a measurement value (phase value) is greater than or equal to the threshold value specified by the `INPUTRAC` command. (Option 022 only)

TRABGE

Query Response	{1|0}

Examples	`OUTPUT @E5100;"TRABGE"`
	
	`OUTPUT @E5100;"TRABGE?"`
	`ENTER @E5100;A`

TRABLE

Sets the condition so that the measurement is advanced to the next point when a measurement value (phase value) is less than or equal to the threshold value specified by the `INPUTRAC` command. (Option 022 only)

Query Response	{1|0}

Examples	`OUTPUT @E5100;"TRABLE"`
	
	`OUTPUT @E5100;"TRABLE?"`
	`ENTER @E5100;A`

TRAFDATA

Set the trap function on the data trace. (Option 022 only)

Query Response	{0|1}

Examples	`OUTPUT @E5100;"TRAFDATA"`
	
	`OUTPUT @E5100;"TRAFDATA?"`
	`ENTER @E5100;A`

TRAFMEMO

Set the trap function on the sub trace. (Option 022 only)

Query Response	{0|1}

Examples	`OUTPUT @E5100;"TRAFMEMO"`
	
	`OUTPUT @E5100;"TRAFMEMO?"`
	`ENTER @E5100;A`

TRAP␣{OFF|ON|0|1}

Set the trap function on/off. (Option 022 only)

Parameter Description	`OFF` or `0` : Set the trap function off. `ON` or `1` : Set the trap function on.
Query Response	{0|1}

Examples	`OUTPUT @E5100;"TRAP ON"`
	
	`OUTPUT @E5100;"TRAP?"`
	`ENTER @E5100;A`

TRAR␣*<value1>*,*<value2>*

Set the start and the end points for the partial sweep for the trap function.

If the start point and the end point are not between 1 to the value specified by `Number of Point`, E5100A/B option 022 will perform a sweep within the possible range. Use this function to reserve wider sweep range so that you can change the sweep range by simply changing this setup. This way, you can save more measurement time than changing the whole sweep setup. (Option 022 only)

Parameter Description	*<value1>* : The start point for the partial sweep *<value2>* : The end point for the partial sweep
Query Response	{*value1*}{*value2*}

Examples	`OUTPUT @E5100;"TRAR 10, 20"`
	
	`OUTPUT @E5100;"TRAR?"`
	`ENTER @E5100;A,B`

WRIT16

Set the bit width of the data outputted to the I/O port while performing the trap function to 16 bit. The port F is used. (Option 022 only)

Query Response	{0|1}

Examples	`OUTPUT @E5100;"WRIT16"`
	
	`OUTPUT @E5100;"WRIT16?"`
	`ENTER @E5100;A`

WRIT24

Set the bit width of the data outputted to the I/O port while performing the trap function to 24 bit. The port H is used. (Option 022 only)

Query Response	{0|1}

Examples	`OUTPUT @E5100;"WRIT24"`
	
	`OUTPUT @E5100;"WRIT24?"`
	`ENTER @E5100;A`

H

Option 023

Overview

This option enables you to measure the drive level characteristics of the resonant frequency (Fr) and the resonant impedance (CI) of crystal resonators quickly and precisely.

To realize a quick measurement of resonant characteristics, this option adopts the Phase Tracking. The phase tracking will control the source signal to trace a specific resonant phase value (generally 0°) to calculate the frequency and the impedance at the phase. The drive level characteristic measurement, the function's most typical usage, realizes a very quick measurement because it measures only the resonant frequency and the resonant impedance as it sweeps the drive level. E5100A option 023 will display the result as described in Figure H-1 where X-axis representing drive level and Y-axis representing resonant frequency and resonant impedance. Instead of an absolute value, a relative value based on the resonant frequency at the minimum level or the nominal resonant frequency is used as the resonant frequency.

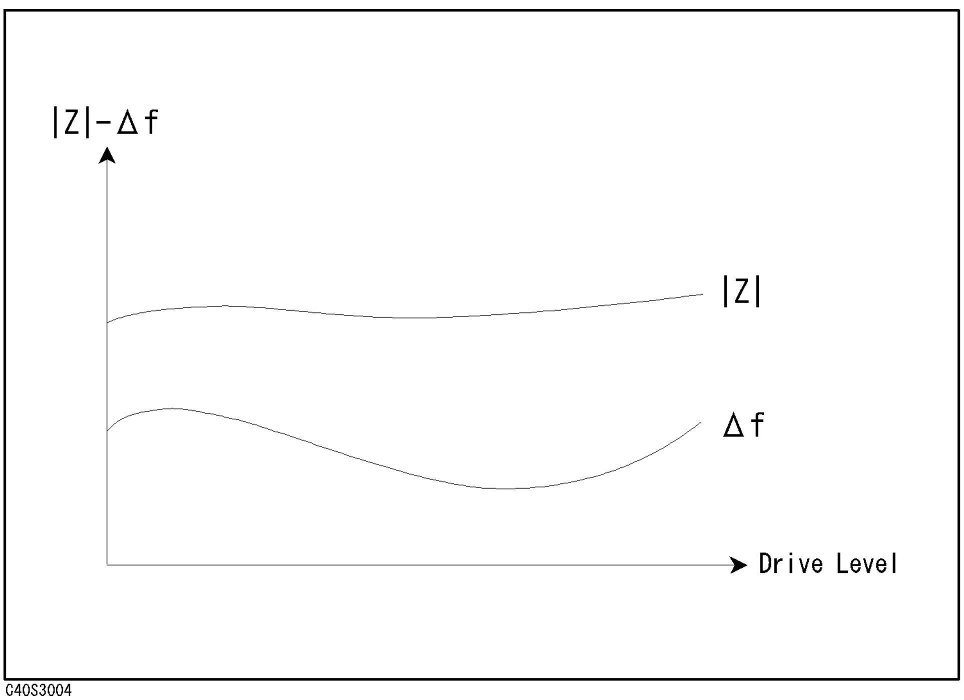
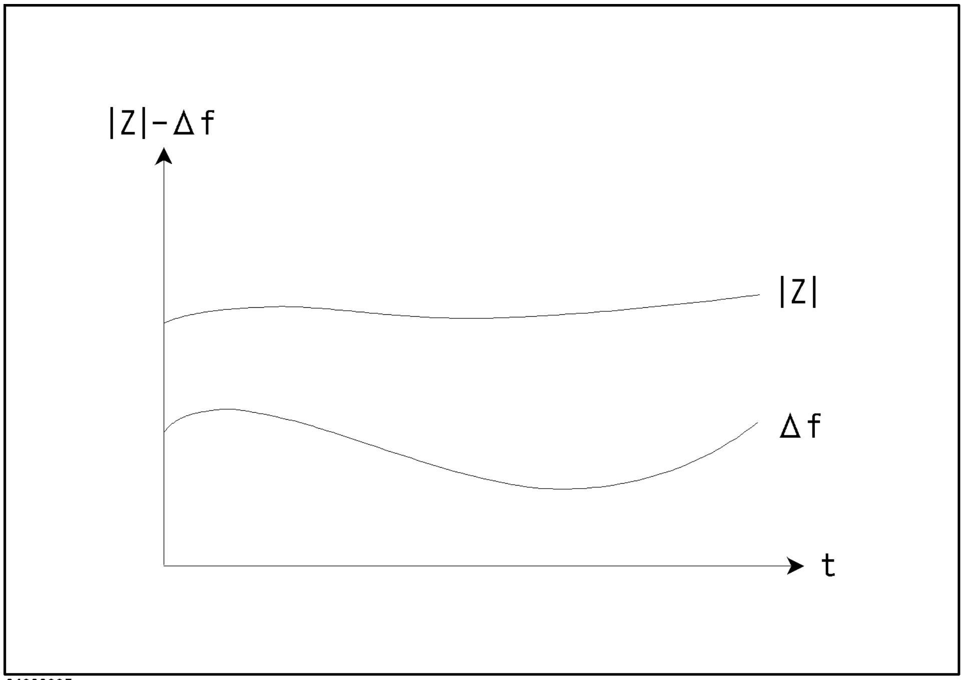


Figure H-1. Measuring Drive Level Characteristics

You can also measure the aging characteristics of the resonant frequency and the resonant impedance by setting the drive level fixed and performing the phase tracking. One application of this aging characteristic measurement may be that you can change the temperature around a resonator with time to measure characteristics of temperature. Instead of an absolute value, a relative value based on the resonant frequency at the minimum level or the nominal resonant frequency is used as the resonant frequency.



C40S3005

Figure H-2. The Aging Characteristic Measurement

Measuring Drive Level Characteristics using Phase Tracking Function

You must use a program to use the phase tracking to measure drive level characteristics or aging characteristics (the temperature characteristic measurement). This program is contained in the attached sample disk for option 022/023 with the file name **SMPL023.BAS**. This chapter explains how to measure these characteristics using a sample program.

The following procedure shows how to set and measure the drive level characteristics using the phase tracking function. GPIB command are shown in parentheses.

1. Measurement Settings for Phase Tracking

- Select the ratio measurement (`MEAS AR`)
- Select the impedance measurement mode (`ANAMODE ZTRAN`)
- Select the power sweep (`SWPT POWE`)
- Select the order base display (`LISDOBASE`)
- Select Z-ΔF format (`FMT MAGZDF`)

2. Setting for Using π Network Test Fixture

The following settings are required when a π Network Test Fixture is used.

- Set Z0 (SETZ)
- Select Watt as the power unit (POWU WATT)
- Select using π network test fixture (PICIRC ON)
- Set CI value (CIVAL)

In the sample program, Characteristic impedance is set at 12.5Ω to support a standard π test fixture.

3. Settings for the Drive Level Measurement

The following settings are required for measuring the drive level characteristics.

- Make a drive level table and input it as the sweep table. (INPUSTIM)

The result will be similar to Figure G-3 because both the up sweep and the down sweep measurements are performed at the same time for a drive level.

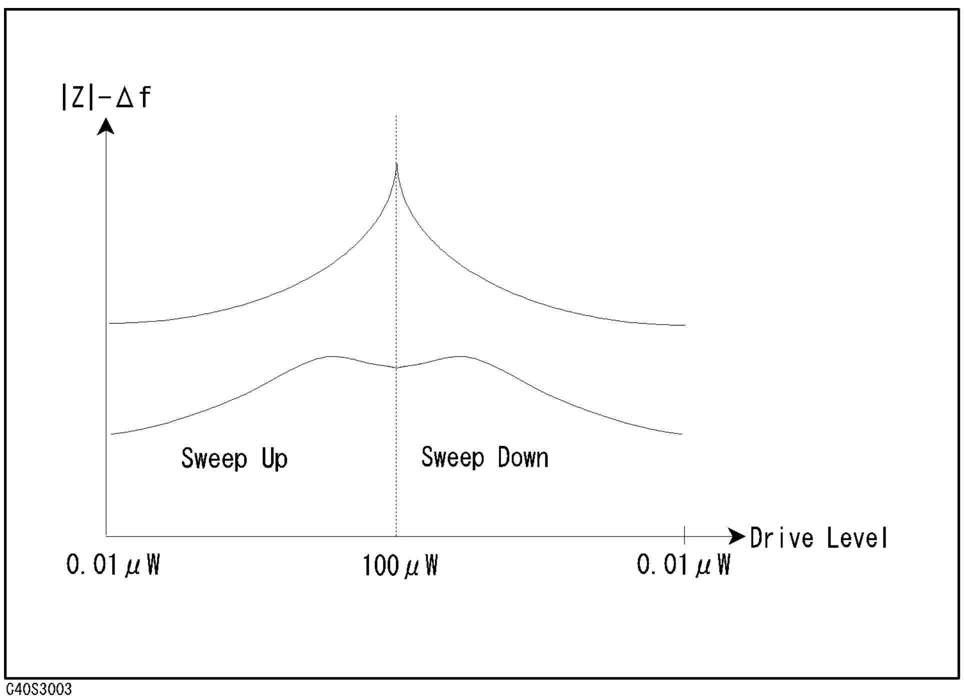


Figure H-3. An Example of the Measurement Result

Note

Set enough power (For example, SPAN = 0 dBm, CENTER = 0 dBm) and set the CW frequency as you measure DUT when you execute calibration measurement.

If the calibration is measured under the low power level, measurement results are not stable.

4. Settings Parameters for Tracking Function

The following process is required for the tracking function.

- Set the start frequency for tracking (PTFR, unit:Hz)
- Set the range from the start frequency for tracking to search the resonant point F_r (PTFRSR, unit:Hz)
- Set the range value for a phase (PTTRGLMT, unit:deg.)
- Set the measurement abort ON/OFF when the phase is over the limit (PTABORT)
- Define the phase at the resonant point. (PTTRGPHS)
- Set the number of tracking for each measurement point (PTREPN)

5. Searching for Resonant Point

E5100A/B searches the range specified by PTFRSR command for the resonant point, which is defined by PTTRGHS. When the resonant point is measured, E5100A/B automatically sets the test signal level to the same level as START and uses the calibration data at the start point. Then Get F_r, CI, and the tracking parameter at the resonant point. (SRCHFR?)

6. Tracking Measurement

To start the tracking, the following procedure is required.

- Set the phase tracking ON (PTRACK)
- Set the reference frequency (PTFR)
- Set the tracking parameter given by the SRCHFR? query response (PTPARA)
- Check the status of phase tracking (PTSTAT?)

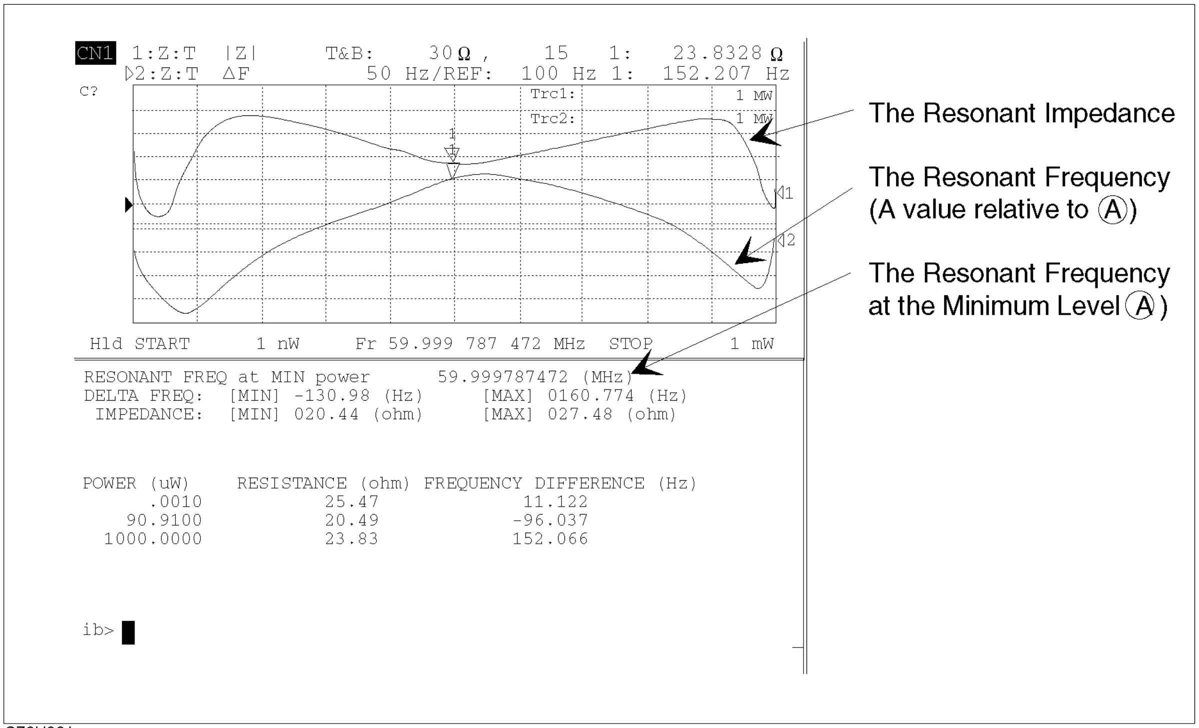
7. The Measurement Result

This program gives you the following result.

Figure G-2 describes the result screen. The upper half of the screen shows the measurement and the lower half shows the resonant frequency at the minimum drive level and resonant frequencies and resonant impedance of typical drive levels (in [W]) are displayed.

Values along the horizontal axis on the screen (drive levels) are displayed in the unit specified by the marker. The data trace indicates resonant impedance and the sub trace indicates resonant frequency (Δf).

When E5100A option 023 displays a resonant frequency (Δf), it displays the value relative to the reference frequency specified by PTFR command.



C72H001

Figure H-4. A Displayed Result

Sample Program List

```
490    CLEAR SCREEN
500   ! MAIN *****************************************************
510   !
520    GOSUB Constants
530    GOSUB Setting
540    GOSUB Cal_setting
550    GOSUB Pi_cal
560    CLEAR SCREEN
570    GOSUB Set_watt
580    GOSUB Set_watt_list
590    GOSUB Set_phase_track
600    GOSUB Start_loop
610    STOP
620   !
630   ! DECLARE CONSTANTS *****************************************
640 Constants:  !
650    Z0=12.5
660   !
670    Ci=10           ! FOR dBm<->Watt CONVERSION
680    W_min=.01       ! START POWER [uW]
690    W_max=100       ! STOP POWER [uW]
700    Ifbw=500       ! IFBW FOR TRACKING
710    Tgt_phas=0      ! TARGET PHASE FOR TRACKING
720    Track_num=2     ! NUMBER OF TIMES TO MEASURE
730    P_nop=50        ! POWER STEP
740    Tgt_limit=8     !
750   !
760    Srch_rng=1000   ! ppm
770    Srch_mode=2      ! SRCHFR Mode (NORMAL)
780    Srch_wait=0      ! SRCHFR Wait Time
790    Fr_trg=4.E+7    ! Target Fr
800   !
```

```
810    RETURN
820   !
830   ! INSTRUMENT SETUP *****************************************
840 Setting:  !
850    ASSIGN @E5100 TO 800
860    OUTPUT @E5100;"PRES"
870    OUTPUT @E5100;"CHAN1;MEAS AR;ANAMODE ZTRAN"
880    OUTPUT @E5100;"HOLD"
890    OUTPUT @E5100;"COUC OFF"
900    OUTPUT @E5100;"IFBW ";Ifbw
910    OUTPUT @E5100;"POIN ";P_nop*2
920    OUTPUT @E5100;"FMT MAGZDF"
930    OUTPUT @E5100;"MULC OFF;SPLD ON;DISAHIHB"
940    OUTPUT @E5100;"SETZ ";Z0
950    RETURN
960   !
970 Cal_setting: !
980    OUTPUT @E5100;"SWPT LINF"
990    OUTPUT @E5100;"CENT ";Fr_trg
1000   OUTPUT @E5100;"SPAN ";Srch_rng*1.E-6*Fr_trg
1010   OUTPUT @E5100;"POWE ";0
1020   RETURN
1030  !
1040 Pi_cal: !
1050   OUTPUT @E5100;"PTRACK OFF"
1060   PRINT TABXY(0,11);"PERFORM PI-CAL"
1070   INPUT "CONNECT OPEN, THEN PRESS ENTER",Tmp$
1080   OUTPUT @E5100;"CALI ONEP;CLASS11A?"
1090   ENTER @E5100;Tmp
1100   INPUT "CONNECT SHORT, THEN PRESS ENTER",Tmp$
1110   OUTPUT @E5100;"CLASS11B?"
1120   ENTER @E5100;Tmp
1130   INPUT "CONNECT LOAD, THEN PRESS ENTER",Tmp$
1140   OUTPUT @E5100;"CLASS11C?"
1150   ENTER @E5100;Tmp
1160   OUTPUT @E5100;"SAV1?"
1170   ENTER @E5100;Tmp
1180   RETURN
1190  !
1200 Set_watt: !
1210   OUTPUT @E5100;"SWPT POWE"
1220   OUTPUT @E5100;"LISDOBASE;SWED UP"
1230   OUTPUT @E5100;"ATTW .002"
1240   OUTPUT @E5100;"POWU WATT;PICIRC ON"
1250   OUTPUT @E5100;"CIVAL ";Ci
1260   RETURN
1270  !
1280  ! SETTING POWER LIST (WATT) *********************************
1290 Set_watt_list: !
1300   ALLOCATE Pwr(1:2*P_nop)
1310   P_min=W_min*1.E-6
1320   P_max=W_max*1.E-6
1330   OUTPUT @E5100;"STAS ";P_min,P_max
1340   P_step=(P_max-P_min)/(P_nop-1)
1350   K=2*P_nop
1360   FOR I=1 TO P_nop
1370     P=P_min+P_step*(I-1)
1380     Pwr(I)=P
1390     Pwr(K)=P
1400     K=K-1
1410   NEXT I
1420   OUTPUT @E5100;"STIM";Pwr(*)
1430   OUTPUT @E5100;"CWFREQ ";Fr_trg
1440   RETURN
1450  !
1460  ! PHASE TRACKING SETTING ************************************
```

```
1470 Set_phase_track: !
1480   OUTPUT @E5100;"PTFR ";Fr_trg
1490   OUTPUT @E5100;"PTFRSR ";Srch_rng*Fr_trg*1.E-6
1500   OUTPUT @E5100;"PTTRGLMT ";Tgt_limit
1510   OUTPUT @E5100;"PTABORT ON"
1520   OUTPUT @E5100;"PTTRGPHS ";Tgt_phas
1530   OUTPUT @E5100;"PTREPN ";Track_num
1540   OUTPUT @E5100;"PTRACK ON"! TRACKING MODE ON
1550   RETURN
1560  !
1570  ! SEARCH "ACTUAL" FR ****************************************
1580 Start_loop: !
1590   LOOP
1600     INPUT "CONNECT DEVICE, THEN PRESS ENTER",Dum$
1610     BEEP
1620     OUTPUT @E5100;"SRCHFR? ";Srch_mode,Srch_wait
1630     ENTER @E5100;Srched_fr,Ci,Pt_param
1640     IF Srched_fr=-1 THEN
1650       PRINT TABXY(0,10),"Fr Search FAILED  "
1660       BEEP
1670       BEEP
1680       GOTO Prompt
1690     END IF
1700  !
1710  ! DLD CHARACTERISTICS MEASUREMENT ***************************
1720  !
1730     OUTPUT @E5100;"PTFR ";Srched_fr
1740     OUTPUT @E5100;"PTPARA ";Pt_param
1750     OUTPUT @E5100;"SING?"
1760     ENTER @E5100;Dummy
1770  !
1780     OUTPUT @E5100;"PTSTAT?"
1790     ENTER @E5100;Result
1800     IF Result=0 THEN
1810       BEEP
1820       BEEP
1830       PRINT TABXY(0,10),"Phase Track FAILED"
1840       GOTO Prompt
1850     END IF
1860  !
1870     GOSUB Analysis
1880  !
1890     GOSUB Printing
1900  !
1910 Prompt:    !
1920   END LOOP
1930  !
1940 Analysis: ! DATA ANALYSIS
1950   OUTPUT @E5100;"ANARFULL"
1960   OUTPUT @E5100;"ANAOMEMO"
1970   OUTPUT @E5100;"OUTPMINMAX?"
1980   ENTER @E5100;Min_df,Mindf_p,Max_df,Maxdf_p
1990   OUTPUT @E5100;"ANAODATA"
2000   OUTPUT @E5100;"OUTPMINMAX?"
2010   ENTER @E5100;Min_z,Min_zp,Max_z,Max_zp
2020   RETURN
2030  !
2040 Printing: ! PRINTING ROUTINE
2050   PRINT TABXY(0,1)
2060   PRINT USING "27A,5X,3D.9D,X,5A";"RESONANT FREQ at MIN power",
       Srched_fr*1.E-6,"(MHz)"
2070   PRINT USING "11A,2X,6A,X,4Z.2D,X,6A,3X,6A,X,4Z.3D,X,5A";
       "DELTA FREQ:","[MIN]",Min_df,"(Hz)","[MAX]",Max_df,"(Hz)"
2080   PRINT USING "X,10A,2X,5A,2X,3Z.2D,X,5A,5X,5A,2X,3Z.2D,X,5A";
       "IMPEDANCE:","[MIN]",Min_z,"(ohm)","[MAX]",Max_z,"(ohm)"
2090   !
```

```
2100   PRINT TABXY(0,6)
2110   PRINT USING "5A,X,4A,4X,10A,X,5A,2X,20A,X,4A";"POWER","(uW)",
       "RESISTANCE","(ohm)","FREQUENCY DIFFERENCE","(Hz)"
2120   FOR I=1 TO 3
2130     Point=10^(I-1)
2140     OUTPUT @E5100;"OUTPFORMP? ";Point
2150     ENTER @E5100;Z,Df
2160     PRINT USING "X,4D.4D,11X,3D.2D,12X,4D.3D";Pwr(Point)*1.E+6,Z,Df
2170   NEXT I
2180   PRINT TABXY(0,10),"                       "
2190  !
2200   RETURN
2210  !
2220  END
```

Measuring Aging Characteristics

You can measure the aging characteristics by setting the drive level fixed and performing the phase tracking. You can also read out the time value using the marker time mode. Basically, the measurement setting is almost same as that of the drive level characteristics measurement except for the signal level is constant.

Compensation of Sweep Time for Aging Characteristics

The actual measurement time contents not only the sweep time but also the overhead time to process signal during measurement. PTFOVHD adds the overhead time to the marker time value as to compensate the measurement time value.

The following list shows how to use PTFOVHD command.

Sub-routine for Measuring Aging Characteristics

```
770 Meas_time:  !
780     OUTPUT @E5100;"PTRACK ON"
790     OUTPUT @E5100;"SWETAUTO"
800     T1=TIMEDATE
810     OUTPUT @E5100;"SING?"
820     ENTER @E5100;Dummy
830     Meas_time=TIMEDATE-T1
840     OUTPUT @E5100;"SWET?"
850     ENTER @E5100;Anal_sweep_t
860     Pt_ovhd=((Meas_time-.005)/Track_num-Anal_sweep_t)/Nop
870     OUTPUT @E5100;"PTFOVHD ";Pt_ovhd
880     OUTPUT @E5100;"MARKTIME ON"
890     RETURN
```

Notes on the Phase Tracking

When the Phase Tracking Fails

Followings are problems and solutions caused during the phase tracking

- The resonator cannot keep up with a sweep because there are not enough drive level measurement points.

  Solution — Increase the number of measurement points and repeat a measurement
  Increase the number of tracking at each point (PTREPN)

- The actual resonant frequency is out of the sweep span.

  Solution — Check the resonant frequency and repeat a measurement.

- The resonator cannot keep up with a sweep because it is too fast.

  Solution — Set the sweep time slower on the front panel and repeat a measurement.
  Increase the number of tracking at each point (PTREPN)

- A tangent response is observed during a sweep.

  Solution — Set the attenuator switching time slower and repeat a measurement.

- E5100A can not find the resonance point (F_r)

  Solution — Change the search mode of SRCHFR? to 3 or 4.

Increase the number of parameter of PTFRSR.

The above list shows the possible causes of problems. You may want to investigate the cause by setting the display to "Z-Phase" to check at which point tracking exceeded the range and failed.

Limitations on the Phase Tracking

The following is a list of limitations on a measurement when the phase tracking is ON (PTRACK ON).

- The time specified by the SWEEP TIME and the actual sweep time differ.

GPIB Commands for Option 023

FMT␣MAGZDF

Sets the format as Z-Δ format for tracking measurement. (Option 023 only)

Examples	`OUTPUT @E5100;"FMT MAGZDF"`

PTABORT␣{ON|OFF}

Sets the measurement abort ON/OFF when the phase value is over the limit during the tracking. (Option 023 only)

Parameter Description	ON : Abort a measurement when the tracking is failed OFF : Continue a measurement even the tracking is failed
Query Response	{1|0}

Examples	`OUTPUT @E5100;"PTABORT ON"`
	
	`OUTPUT @E5100;"PTABORT?"`
	`ENTER @E5100;A`

PTFOVHD␣<*value*>

Input the parameters required to display the time base at the temperature characteristic measurement (the aging measurement). Refer to "Compensation of Sweep Time for Aging Characteristics" for inputting. (Option 023 only)

Parameter Range	0 to 1 sec
Query Response	{*value*}

PTFR␣<*value*>

Input the start frequency for tracking. (Option 023 only)

Parameter Range	10 kHz to 300 MHz
Query Response	{*value*}

Examples	`OUTPUT @E5100;"PTFR 199 kHz"`
	
	`OUTPUT @E5100;"PTFR?"`
	`ENTER @E5100;A`

PTFRSR␣<*value*>

Sets the range for searching for F_r. (Option 023 only)

Parameter Range	0 Hz to 100 kHz
Query Response	{*value*} (Hz)

Examples	`OUTPUT @E5100;"PTFRSR 5000"`
	
	`OUTPUT @E5100;"PTFRSR?"`
	`ENTER @E5100;A`

PTPARA␣<*value*>

Sets the tracking parameter. The tracking paramter is given by the SRCHFR? command query. (Option 023 only)

Parameter Range	−1000 to 1000
Query Response	{*value*}

Examples	`OUTPUT @E5100;"SRCHFR?''`
	`ENTER @E5100;Fr,Ci,Param`
	`OUTPUT @E5100;"PTPARA";Param`
	
	`OUTPUT @E5100;"PTPARA?"`
	`ENTER @E5100;A`

PTRACK␣{OFF|ON}

Set the phase tracking ON/OFF. (Option 023 only)

PTRACK␣{OFF|ON}

Parameter Description	`OFF` : The phase tracking is OFF `ON` : The phase tracking is ON
Query Response	{0|1}

Examples	`OUTPUT @E5100;"PTRACK ON"`
	
	`OUTPUT @E5100;"PTRACK?"`
	`ENTER @E5100;A`

PTREPN␣<*value*>

Sets the number of tracking on each point. (Option 023 only)

Parameter Range	1 to 1,000,000
Query Response	{*value*}

Examples	`OUTPUT @E5100;"PTREPN 5"`
	
	`OUTPUT @E5100;"PTREPN?"`
	`ENTER @E5100;A`

PTSTAT?

Returns the status of the phase tracking. (Query only,Option 023 only)

Query Response	{0|1} 0 : Error encountered during phase tracking 1 : The phase tracking was successful.
Examples	`OUTPUT @E5100;"PTSTAT?"` `ENTER @E5100;A`

PTTRGLMT␣<*value*>

Defines the range value for a phase, which is used for the phase tracking. (Option 023 only)

Parameter Range	0° to 180°
Query Response	{*value*}

Examples	`OUTPUT @E5100;"PTTRGLMT 8"`
	
	`OUTPUT @E5100;"PTTRGLMT?"`
	`ENTER @E5100;A`

PTTRGPHS␣<*value*>

Defines the phase at the resonant point. (Option 023 only)

Parameter Range	−180° to 180°
Query Response	{*value*}

Examples	`OUTPUT @E5100;"PTTRGPHS 0"`
	
	`OUTPUT @E5100;"PTTRGPHS?"`
	`ENTER @E5100;A`

SRCHFR?␣{1|2|3|4|5|6},<*value*>

Searches for the resonance frequency (F_r). (Query only, Option 023 only)

Parameter Description	Search mode: Rough (High speed) 1 ↔ 6 Finer (Slow) <*value*> : Waiting time during searching (sec)
Query Response	{*value1*}, {*value2*},{*value3*} {*value1*} : F_r [Hz] {*value2*} : CI [Ω] {*value3*} : Tracking parameter
Examples	`OUTPUT @E5100;"SRCHFR? 2,0"` `ENTER @E5100;A,B,C`

I

Partial Sweep

The partial sweep function means a fucntion can sweep only a part of E5100A/B's sweep span. Although it takes some time to change stimulus value usualy, you don't need change stimulus value. So you can get fast measurement on some measurement which are often changed stimulus value.

Commands for the partial sweep function.

To know more information, see Chapter 8.

PARSMODE	Turn On/Off the partial sweep function.
PARSRANG	Define the partial sweep range using with measurement points.
ANAPSTIMP?	Exchange a frequency value to a measurement point.

Sample Program

This is a simple program of partial sweep.

```
100 ASSIGN @E5100 TO 800
110 OUTPUT @E5100;"POIN 201"
120 OUTPUT @E5100;"STAR 99E6"
130 OUTPUT @E5100;"STOP 101E6"
140 OUTPUT @E5100;"SING?"
150 ENTER @E5100;Dummy
160 !
170 OUTPUT @E5100;"ANASTIMP? 99.8E6"        !Exchange the start frequency
180 ENTER @E5100;Start_p                     !value to a meas. point.
190 OUTPUT @E5100;"ANASTIMP? 100.2E6"        !Exchange the stop frequency
200 ENTER @E5100;Stop_p                      !value to a meas. point.
210 !
220 OUTPUT @E5100;"PARSRANG ";Start_p,Stop_p !Define the partial sweep range.
230 OUTPUT @E5100;"PARSMODE ON"              !Partial sweep ON
240 OUTPUT @E5100;"SING?"
250 ENTER @E5100;Dummy
260 END
```

Figure I-1. Sample Program : Partial Sweep

J

File Transfer Function

Overview

This appendix describes how to use the file transfer function, showing you the sample programs.

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

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

  For example, you can transfer the files saved on the E5100A/B storage deveice, such as graphic files or data array (ASCII format) files, to the external controller. Then, you can output the graphic file transferred from a printer connected to the external controller, or manage the data array (ASCII format) trasferred using a software running on the external controller.

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

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

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

  Preparation:

  Use the keys on the front panel to establish the setting required for your measurement. Store it on the storage device of the E5100A/B, then transfer the file to the external controller, and store it on an external storage device. Repeat this procedure for all of the settings required for your measurement.

  Measurement:

  Choose a necessary setting file from those stored and transfer it to the E5100A/B using the external controller. Then, recall the file to set the E5100A/B for the measurement and perform the measurement using the GPIB commands.

The storage device of the E5100A/B allows you to handle files listed below in the DOS format. For DOS format files, both binary files and ASCII files can be transferred.

- Binary files
  - Instrument settings and internal data array (ALL)
  - Instrument settings (STATE ONLY)
  - Internal data array (DATA ONLY (BINARY))

- Graphic images (SAVE GRAPHICS)

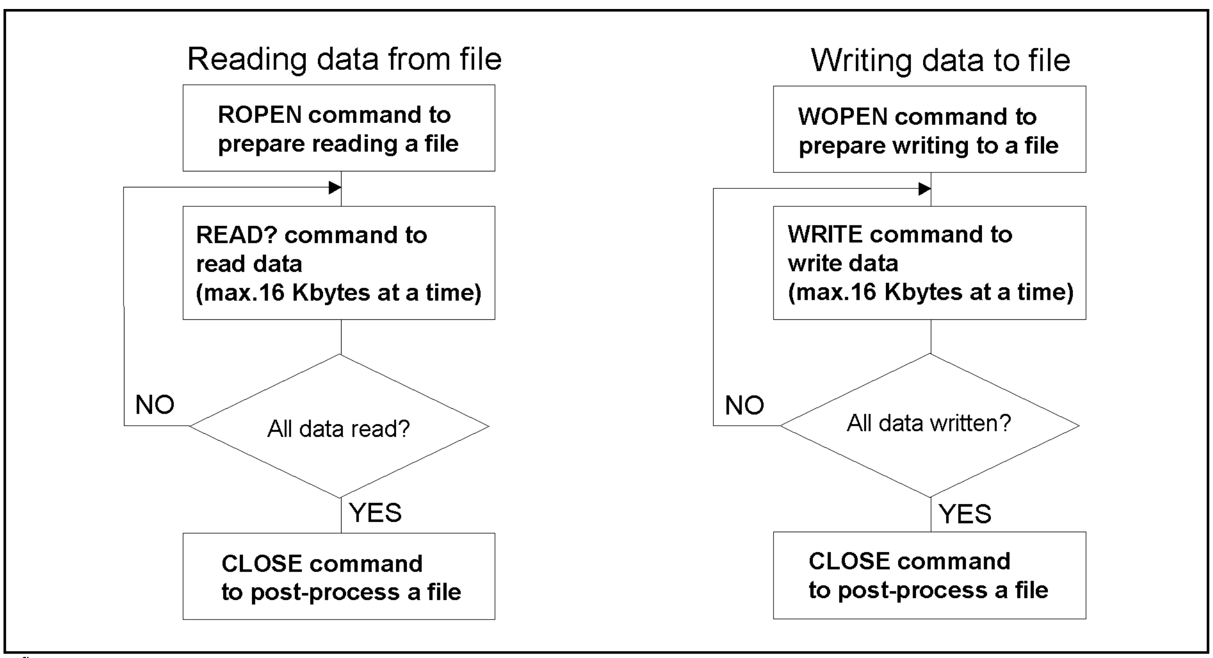
- ASCII files
  - Internal data arrays (DATA ONLY (ASCII))
  - HP instrument BASIC programs

Commands for the File Transfer Function.

For details, see Command Reference.

CLOSE	Makes a file access-disabled
READ?	Reads data from a file that has been read-enabled using the ROPEN command
ROPEN	Makes a file read-enabled
WOPEN	Makes a file write-enabled
WRITE	Writes data in a file that has been write-enabled using the WOPEN command

Used in combination with the commands as shown in Figure J-1 to read/write the data.



rwflow

Figure J-1. Procedure of executing commands to read/write data

The data which are read or written are applied to the fixed length block format defined in IEEE488.2 (Figure J-2).

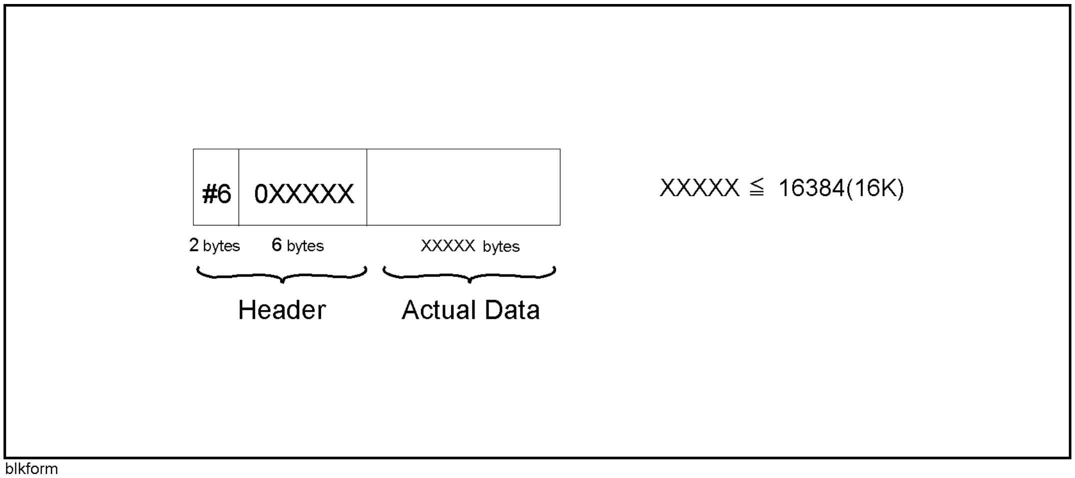


Figure J-2. Fixed length block format

Sample Programs

The following sample programs are related to the file transfer function.

- File Transfer from E5100A/B to External Controller
- File Transfer from External Controller to E5100A/B
- Displaying List of Files in Current Directory

File Transfer from E5100A/B to External Controller

This program transfers a specified file in the current directory (memory disk in the sample program) of the E5100A/B to the current directory of the storage device (A drive in the sample program) connected to the external controller, giving a file name you desire.

When executed, this program first prompts you to enter a source file name, as shown below. Enter the name of a file you want to transfer.

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

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

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

Disk

This program is contained in the attached sample disk with the file name **FILE1_E** (for the external controller). The program for Instrument BASIC is not provided.

```
110  !
120  !  File transfer (Instrument -> Controller)
130  !
```

```
140  DIM Src_file$[50],Dst_file$[50]
150  ASSIGN @Agte5100 TO 717
160  OUTPUT @Agte5100;"*rst"
170  !
180  MASS STORAGE IS "a:\
190  OUTPUT @Agte5100;"STODMEMO"
200  !
210  PRINT "  ENTER SOURCE FILE NAME ON INSTRUMENT ?        ";
220  INPUT Src_file$
230  PRINT Src_file$
240  !
250  PRINT "  ENTER DESTINATION FILE NAME ON CONTROLLER ?   ";
260  INPUT Dst_file$
270  PRINT Dst_file$
280  !
290  Copy_from_instr(@Agte5100,Src_file$,Dst_file$)
300  !
310  END
320  !
330  !  copy_from_instrument
340  !
350  SUB Copy_from_instr(@Agte5100,Src_file$,Dst_file$)
360      DIM Len$[6],Img$[32],Dmy$[2]
370      !
380      ON ERROR GOTO Skip_purge
390      PURGE Dst_file$
400 Skip_purge:  OFF ERROR
410      CREATE Dst_file$,1
420      ASSIGN @Dst_file TO Dst_file$
430      !
440      CLEAR @Agte5100
450      OUTPUT @Agte5100;"CLES"
460      OUTPUT @Agte5100;"ROPEN """;Src_file$;""""
470      IF FNCheck_error(@Agte5100,"<CPFI: ropen>")=-1 THEN SUBEXIT
480      !
490      LOOP
500          OUTPUT @Agte5100;"READ?"
510          ENTER @Agte5100 USING "#,2A";Dmy$
520          ENTER @Agte5100 USING "#,6A";Len$
530          Block_size=VAL(Len$)
540          !
550          IF Block_size=0 THEN
560              ENTER @Agte5100 USING "%,A";Dmy$
570              ASSIGN @Dst_file TO *
580              OUTPUT @Agte5100;"CLOSE"
590              SUBEXIT
600          END IF
610          !
620          ALLOCATE Dat$[Block_size]
630          Img$="#,"&VAL$(Block_size)&"A"
640          ENTER @Agte5100 USING Img$;Dat$
650          ENTER @Agte5100 USING "%,A";Dmy$
660          OUTPUT @Dst_file USING Img$;Dat$
670          DEALLOCATE Dat$
680          !
```

```
690          IF FNCheck_error(@Agte5100,"<CPFI: block read>")=-1 THEN SUBEXIT
700      END LOOP
710  SUBEND
720  !
730  !  Instrument Error Check
740  !
750  DEF FNCheck_error(@Agte5100,Str$)
760      DIM Err$[64]
770      OUTPUT @Agte5100;"OUTPERRO?"
780      ENTER @Agte5100;Err$
790      IF Err$<>"0,""No error""" THEN
800          PRINT "ERROR: ";Str$;"   ";Err$
810          RETURN -1
820      ELSE
830          RETURN 0
840      END IF
850  FNEND
```

Lines 180 to 190 sets the current directory of the external controller to A drive and sets the current directory of the E5100A/B to the memory disk. You can set the current directory of the E5100A/B to the internal flexible disk using the **STODDISK** command. The A drive in the external controller may not be detected under a certain environment of the external controller, so change the drive depending on the situation.
Lines 210 to 270 accept the entry of the source file name and the destination file name.
Line 290 calls the subprogram to transfer a file from the E5100A/B to the external controller.
Lines 380 to 420 prepare for writing to the destination file.
Lines 440 to 470 prepare for reading the source file to the external controller.
Line 500 executes the query command to read data.
Lines 510 to 530 read the part indicating the length of the fixed length block data (see Figure J-2) to obtain the length of the data to be transferred.
Lines 550 to 600 check the data length. If the data length is 0, the transfer process is terminated.
Depending on the data length obtained in lines 630 to 650, the program adjusts the format and reads the data part.
Line 660 writes the data to the destination file.
The maximum length of data transferred at a time is 16 Kbytes. Therefore, if the size of the source file is greater than 16 Kbytes, the transfer routine, lines 500 to 690, is repeated until transferring all of the data is completed.
Lines 750 to 850 provide a function to check that no error has occurred in the E5100A/B.

File Transfer from External Controller to E5100A/B

This program transfers a specified file in the current directory of the storage device (A drive in the sample program) connected to the external controller to the current directory of the selected storage device (memory disk in the sample program) of the E5100A/B, giving a file name you desire.

This program, when executed, first prompts you to enter a source file name, as shown below. Enter the name of a file you want to transfer.

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

Next, the program prompts you to enter the size of the source file as shown below (in this example, SAMPLE.STA has been entered as the source file name). Enter the size correctly in bytes.

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

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

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

Disk

This program is contained in the attached sample disk with the file name **FILE2_E** (for the external controller). The program for Instrument BASIC is not provided.

```
110  !
120  !  File transfer (Controller -> Instrument)
130  !
140  DIM Src_file$[50],Dst_file$[50]
150  ASSIGN @Agte5100 TO 717
160  OUTPUT @Agte5100;"*rst"
170  !
180  MASS STORAGE IS "a:\"
190  OUTPUT @Agte5100;"STODMEMO"
200  !
210  PRINT "  ENTER SOURCE FILE NAME ON CONTROLLER ?        ";
220  INPUT Src_file$
230  PRINT Src_file$
240  !
250  PRINT "  ENTER SOURCE FILE SIZE ?                      ";
260  INPUT Src_size
270  PRINT Src_size
280  !
290  PRINT "  ENTER DESTINATION FILE NAME ON INSTRUMENT ?   ";
300  INPUT Dst_file$
310  PRINT Dst_file$
320  !
330  Copy_to_instr(@Agte5100,Src_file$,Src_size,Dst_file$)
340  !
350  END
360  !
370  !  copy_to_instrument
380  !
390  SUB Copy_to_instr(@Agte5100,Src_file$,Src_size,Dst_file$)
400      DIM Img$[32]
410      Max_bsize=16384
420      !
430      ASSIGN @Src_file TO Src_file$
440      !
450      CLEAR @Agte5100
460      OUTPUT @Agte5100;"CLES"
470      OUTPUT @Agte5100;"WOPEN """;Dst_file$;""""
480      IF FNCheck_error(@Agte5100," <CPTI: wopen>")=-1 THEN SUBEXIT
```

```
490      Xfr_done=0
500      !
510      LOOP
520          SELECT (Src_size-Xfr_done)
530              CASE >Max_bsize
540                  Block_size=Max_bsize
550              CASE 0
560                  ASSIGN @Src_file TO *
570                  OUTPUT @Agte5100;"CLOSE"
580                  SUBEXIT
590              CASE ELSE
600                  Block_size=(Src_size-Xfr_done)
610          END SELECT
620          Xfr_done=Xfr_done+Block_size
630          !
640          ALLOCATE Dat$[Block_size]
650          !
660          Img$="#,"&VAL$(Block_size)&"A"
670          ENTER @Src_file USING Img$;Dat$
680          !
690          Img$="8A,ZZZZZZ,"&VAL$(Block_size)&"A"
700          OUTPUT @Agte5100 USING Img$;"WRITE #6",Block_size,Dat$,END
710          DEALLOCATE Dat$
720          IF FNCheck_error(@Agte5100," <CPTI: block write>")=-1 THEN SUBEXIT
730      END LOOP
740  SUBEND
750  !
760  !  Instrument Error Check
770  !
780  DEF FNCheck_error(@Agte5100,Str$)
790      DIM Err$[64]
800      OUTPUT @Agte5100;"OUTPERRO?"
810      ENTER @Agte5100;Err$
820      IF Err$<>"0,""No error""" THEN
830          PRINT "ERROR: ";Str$;"   ";Err$
840          RETURN -1
850      ELSE
860          RETURN 0
870      END IF
880  FNEND
```

Lines 180 to 190 sets the current directory of the external controller to A drive and sets the current directory of the E5100A/B to the memory disk. You can set the current directory of the E5100A/B to the internal flexible disk using the **STODDISK** command. The A drive in the external controller may not be detected under a certain environment of the external controller, so change the drive depending on the situation.

Lines 210 to 310 accept the entry of the source file name and its size and the destination file name.

Line 330 calls the subprogram to transfer a file from the external controller to the E5100A/B.

Lines 470 to 480 prepare for writing the file to the destination storage device.

Lines 520 to 610 calculate the length of the data that has not been transferred based on the source file size previously entered and the length of the data that has been already transferred. If the length of the remaining data does not exceed 16 Kbytes, it is set as the transfer data length; otherwise, 16 Kbytes is set as the transfer data length. Note that, if the length of the data not transferred is 0 at this time, the transfer process is terminated.

Lines 660 to 670 read data, whose amount is specified by the transfer data length, from the source file.

Lines 690 to 700 write data to the destination file in the fixed length block format (see Figure J-2).

The maximum length of data transferred at a time is 16 Kbytes. Therefore, if the size of the source file is greater than 16 Kbytes, the transfer routine, lines 520 to 720, is repeated until transferring all of the data is completed.

Lines 780 to 880 provide a function to check that no error has occurred in the E5100A/B.

Note To transfer a file from the external storage device to the E5100A/B, you must check the file size (number of bytes) in advance .

Displaying List of Files in Current Directory

This program displays the list of the files in the current directory of the E5100A/B.

Disk This program is contained in the attached sample disk with the file name **FILE3_E** (for the external controller). The program for Instrument BASIC is not provided.

```
110  !
120  !  File list
130  !
140  ASSIGN @Agte5100 TO 717
150  OUTPUT @Agte5100;"*rst"
160  !
170  Dir_instr(@Agte5100)
180  !
190  END
200  !
210  !  Dir_instr
220  !
230  SUB Dir_instr(@Agte5100)
240      DIM Stor_dev$[5],File_name$[13]
250      !
260      Stor_dev$="MEMO"
270      !
280      OUTPUT @Agte5100;"STOD";Stor_dev$
290      PRINT "["&Stor_dev$&"]: "
300      PRINT "Size[byte]    File Name"
310      PRINT "------------------------"
320      OUTPUT @Agte5100;"FNUM?"
330      ENTER @Agte5100;File_count
340      IF File_count>=1 THEN
350          FOR I=1 TO File_count
360              OUTPUT @Agte5100;"FNAME? ";I
370              ENTER @Agte5100;File_name$
380              OUTPUT @Agte5100;"FSIZE? """&File_name$&""""
```

```
390             ENTER @Agte5100;File_size
400             PRINT USING "XX,DDDDDD,XXXX,K";File_size,File_name$
410          NEXT I
420       END IF
430   SUBEND
```

Line 170 calls the subprogram to display the list of the files in the current directory.
Lines 260 to 290 sets the current directory to the instrument's memory disk and display it.
Lines 320 to 330 check the number of the files in the current directory.
Lines 340 to 420 check the file name and size of each file and display them, if there are any files in the current directory.

K

Measuring Load Resonance Frequency (FL) Using a CL Board with Fixed C

Introduction

When a load capacitance (CL) is connected in series or in parallel to a crystal resonator, the lower of the two frequencies with zero phase will be generally higher than that when no load capacitance is connected. This lower frequency is called load resonance frequency (FL) and varies depending on the CL value. When a crystal resonator is employed in a physical electronic circuit, CL will always be connected to the resonator. Therefore, FL is a critical parameter.

When the crystal resonator FL needs to be evaluated for trading, each of the parties involved in the trade must measure the FL by connecting the same CL to the resonator. For example, it has been common to measure the CL with an LCR meter while adjusting it with the trimmer. However, conventional measurement procedures resulted in varying FL values depending on the accuracy of the LCR meter and the manner in which the FL is measured. Another drawback to conventional methods is that the CL set with the trimmer could vary due, for example, to vibrations, thus producing different evaluations in relation to the same resonator among traders.

The measurement procedure presented here uses a specially designed command to determine the FL by connecting a CL board having a precisely measured fixed C and calculate the FL for the desired (or target) CL through corrections. This procedure can minimize errors caused, for example, by improper trimmer adjustments and stray capacitance.

The 41900A π network text fixture and the 41901A SMD π network text fixture come with a CL board with a fixed load capacitance. Each of these CL values is precisely measured for utmost accuracy in determining the FL for the desired CL value.

Basic Program Flow

The Figure K-1 shows the basic program flow.

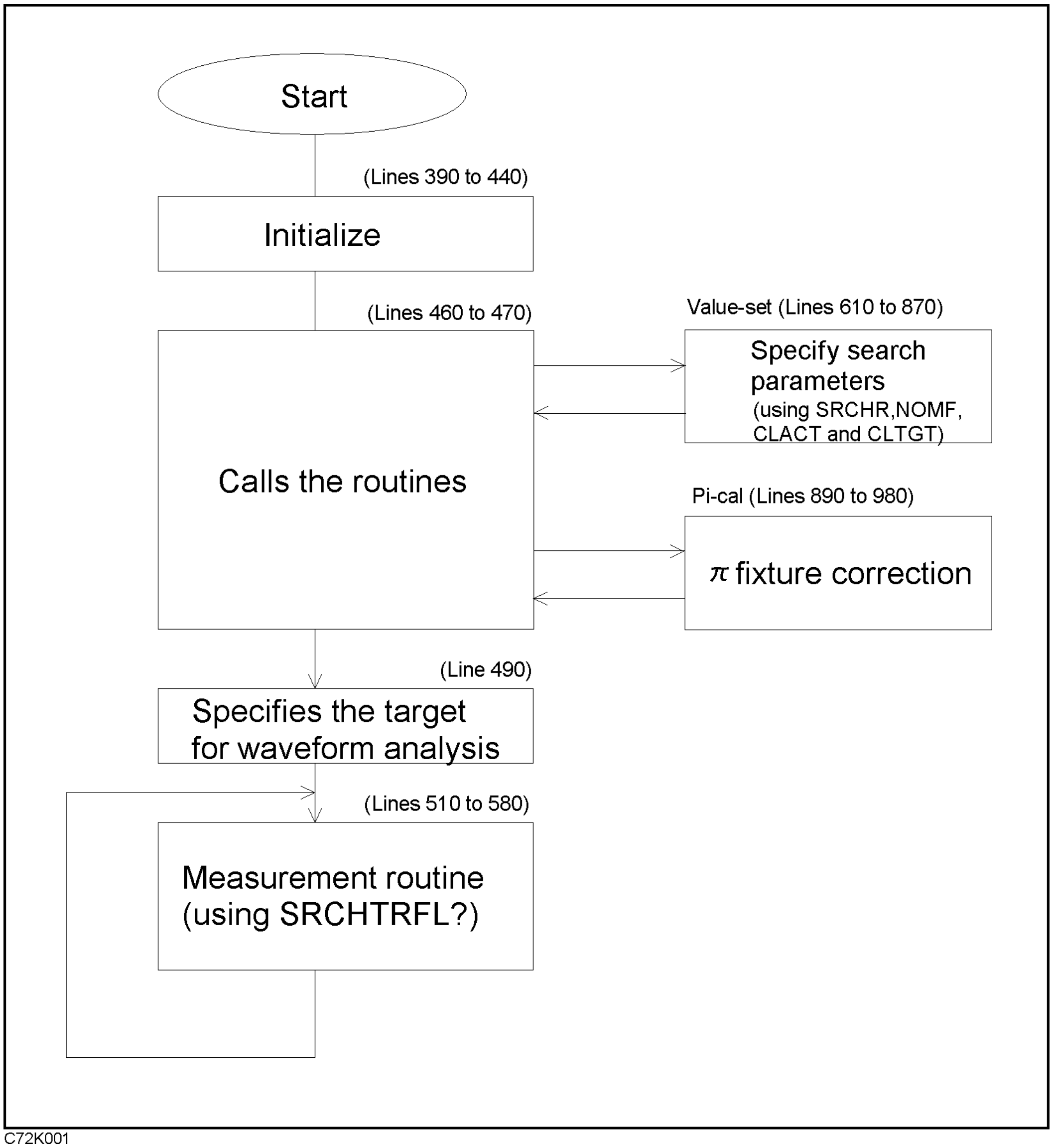


Figure K-1.

Commands

Commands shown in the Table K-1 are used to measure the load resonance frequency (FL) of crystal resonators using a CL board with a fixed C. See the description of commands in the corresponding chapter for more information.

Table K-1. Commands

Command	Function
SRCHTRFL?	Use this command to measure through search the load resonance frequency (FL) of a sample with the known CL and determines the FL for the desired (target) CL through corrections.
SRCHR	Use this command to specify in [Hz] the target range of frequencies for measurement through search using the SRCHTRFL? command.
NOMF	Use this command to specify in [Hz] the nominal FL of the sample for measurement through search using the SRCHTRFL? command.
CLACT	Use this command to specify in [F] the actual CL connected to the sample for measurement through search using the SRCHTRFL? command.[1]
CLTGT	Use this command to specify in [F] the target CL for measurement through search using the SRCHTRFL? command.

1 The 41900A π network text fixture and the 41901A SMD π network text fixture come with a CL board with a fixed CL. Therefore, simply specify this CL using this command.

Note Be sure to specify necessary search parameters using each of the `SRCHR`, `NOMF`, `CLACT`, and `CLTGT` commands before executing the `SRCHTRFL?` command. Note that you can specify search parameters in any order.

Program Listing

Note

- This program file is stored under the name of **SRCHTRFL.BAS** on the attached sample program disk (Agilent Part Number E5100-61002). This program is designed for use with the HP Instrument BASIC.
- The program listing shown below does not include comments given before line 390 in the sample program. Note also that the comment shown to the right of each program line is not included in the sample program.

```
390   ASSIGN @E5100 TO 800             ! Specifies E5100A/B address.
400   OUTPUT @E5100;"PRES"             ! Initializes the settings.
410   OUTPUT @E5100;"ANAMODE ZTRAN"    ! Selects the impedance (transfer) mode.
420   OUTPUT @E5100;";DISAALLB"        ! Selects the entire screen for BASIC.
430   OUTPUT @E5100;"PICIRC ON"        ! Selects the π fixture mode.
440   DIM Ma$[16],Un$[2]               ! Specifies the character string
                                       ! variable size.
450   !
460   GOSUB Value_set                  ! Calls the routine to specify search
                                       ! parameters.
470   GOSUB Pi_cal                     ! Calls the π fixture correction routine.
480   !
490   OUTPUT @E5100;"ANAOCH1;ANARFULL;ANAODATA" ! Specifies the target for
                                       ! waveform analysis.
500   !                           (Channel 1, entire display range, data trace)
510 Meas:                              ! Starts the measurement routine.
520   INPUT "Connect the DEVICE with CL board",B$ ! Suggests that the CL board
                                       ! be connected to the sample.
530         OUTPUT @E5100;"SRCHTRFL? ";Search_mode,W_t ! Measures the FL.
540         ENTER @E5100;Fl_trg1        ! Obtains measurement results from
                                       ! the E5100A/B.
550   Ma$="FL at target CL"            ! Assigns a value to the character string.
560   Un$="Hz"                         ! Assigns a value to the character string.
570   PRINT USING "16A,9D,2A";Ma$,Fl_trg1,Un$ ! Displays the measurement results
                                       ! on the screen.
580   GOTO Meas                        ! Returns to the top of the measurement
                                       ! routine to restart measurement.
590   STOP
600   !
610 Value_set:                         ! Starts the routine to specify search
                                       ! parameters.
620   INPUT "Nominal frequency?[MHz]",Nf1 ! Inputs the nominal FL of the sample in [MHz].
630        Nf=Nf1*1.E+6                ! Converts the nominal FL into that in [Hz].
640        OUTPUT @E5100;"NOMF";Nf     ! Specifies the nominal FL.
650   INPUT "Search range?[ppm]",Sr1   ! Inputs in [ppm] the frequency range
                                       ! to be searched.
660        Sr=Sr1*Nf1                  ! Converts the range into that in [Hz].
670        OUTPUT @E5100;"SRCHR ";Sr   ! Specifies the frequency range.
680   INPUT "Nominal CI?[ohm]",Nci     ! Inputs the nominal CI in [Ω].
690        OUTPUT @E5100;"CIVAL";Nci   ! Specifies the nominal CI.
700   INPUT "Power?[uW]",Pow1          ! Inputs the test signal power in [μW].
710        Pow=Pow1*1.E-6              ! Converts the power into that in [W].
720        OUTPUT @E5100;"POWU WATT"   ! Specifies W as the unit for signal power.
730        OUTPUT @E5100;"POWE";Pow    ! Specifies the signal power.
740   INPUT "Actual capacitance?[pF]",Act1 ! Inputs the actually connected CL in [pF].
750        Act=Act1*1.E-12             ! Converts the actual CL into that in [F].
760        OUTPUT @E5100;"CLACT";Act   ! Specifies the actual CL.
770   INPUT "Target capacitance?[pF]",Trg1 ! Inputs the target CL in [pF].
780        Trg=Trg1*1.E-12             ! Converts the target CL into that in [F].
790        OUTPUT @E5100;"CLTGT";Trg   ! Specifies the target CL.
800   !
810   Search_mode=3                    ! Specifies the default search mode.
```

```
820   W_t=0                            ! Specifies the default wait time per
                                       ! frequency.
830   INPUT "Search mode?[1-6]",Search_mode ! Inputs the search mode (intervals
                                       !  at which the frequency changes).
840   INPUT "Waiting time?[ms]",W_t1    ! Inputs the wait time in [msec].
850   W_t=W_t1*1.E-6                   ! Converts the wait time to that in [sec].
860   !
870   RETURN
880   !
890 Pi_cal:                            ! Starts the π fixture correction routine.
900   OUTPUT @E5100;";CALI ONEP"        ! Specifies the 1-port 3-term
                                       ! correction method.
910   INPUT "Leave the terminal OPEN, and press[return].",Cal$ ! Instructs that
                                       ! the terminal be left open.
920   OUTPUT @E5100;";CLASS11A"         ! Selects OPEN for the S11 correction
                                       ! standard class.
930   INPUT "Connect SHORT, and press[return]",Cal$ ! Instructs that the
                                       ! terminal be shorted.
940   OUTPUT @E5100;";CLASS11B"         ! Selects SHORT for the S11
                                       ! correction standard class.
950   INPUT "Connect LOAD, and press[return]",Cal$ ! Instructs that the load
                                       ! be connected to the terminal.
960   OUTPUT @E5100;";CLASS11C"         ! Selects LOAD for the S11 correction
                                       ! standard class.
970   OUTPUT @E5100;";SAV1"             ! Calculates and stores the error facto
                                       ! for the 3-term correction.
980   RETURN
990   END
```

Index

A

ABORT , 1-4
accessing calibration coefficient arrays , 3-5
active channel , 2-3
active controller , 1-4
ASCII data transfer , 3-7

B

binary data transfer , 3-8

C

calibration coefficient , 3-5
clearing register , 5-10
*CLS , 5-10
condition register , 5-3
controller , 1-4

D

data array, 3-4
data arrays , 3-1
data header , 3-9
data processing flow , 3-1
data transfer methods , 3-7
device selector , 1-3, **1-5**
display allocation , 1-6
DOS , 1-8

E

enable register , 5-2
event register , 5-2
external controller , 1-7

F

File Transfer Function, J-1
formatted data array , 3-5

G

general status register model , 5-1
getting data from the analyzer , 3-10
GPIB , 1-4, 1-9
GPIB address
  setting , 1-6, 1-7
GPIB address , **1-5**
GPIB cable , 1-7
GPIB command , 2-1
GPIB control , 2-1

H

HP BASIC , 1-9
HP Instrument BASIC , 1-5

I

IEEE 32-bit floating point format , 3-8
IEEE 488 bus , 1-4
IEEE 64-bit floating point format , 3-8
initializing module , 1-2
Instrument BASIC
  preparing , 1-6
Instrument BASIC , 1-6
Instrument BASIC editor , 1-8
Interrupt , 5-10

L

LIF , 1-8

M

manual changes , A-1
manual overview , 1-1
memory trace , 3-6
module , 1-2
MS-DOS format , 3-9
MSS , 5-2

O

option UKR , 1-6
OUTPUT , 2-1

P

PASS CONTROL , 1-4
preparation for operation , 1-6
Program Module
  detecting sweep end using SRQ and interrupt , 5-10
  getting data from analyzer using binary transfer , 3-10
  getting data trace array , 3-5
  getting memory array data , 3-6
  getting raw data array , 3-4
  initialize for the external controller , 1-3
  initialize module for Instrument BASIC , 1-2
  presetting the analyzer , 2-1

querying active channel setting , 2-2
reading an event register , 5-9
retrieving data from the analyzer using ASCII transfer , 3-8
sending data to analyzer by ASCII transfer , 3-8
sending multiple commands in a line , 2-1
Setting data to the data array, 3-4
setting the active channel , 2-3
setting the Measurement and Format , 2-3
setting trace array data , 3-6
program module , 1-2
PROGram subsystem command, 8-111

Q

query , 2-2
query response , 2-2

R

raw data , 3-4
reading an event register directly , 5-9
related documentation , 1-9
remote mode , 1-7
required instrument , 1-1
RQS , 5-2, 5-10

S

sample program
loading , 1-8
reading on PC , 1-8
sample program disk , 1-8
SCPI , 1-9, 8-111
select code , 1-3
serial number, A-2
serial poll , 5-2
service request , 5-2
SPOLL , 5-2
SRQ , 5-2, **5-10**
SRQ generation , 5-10
status byte register , 5-2
status register
using , 5-9
status register structure , 5-4
status reporting system , 5-1
*STB , 5-2
system controller , 1-4

T

transition filter , 5-3

U

unformatted data , 3-4

REGIONAL SALES AND SUPPORT OFFICES

For more information about Agilent Technologies test and measurement products, applications, services, and for a current sales office listing, visit our web site: http://www.agilent.com/find/tmdir. You can also contact one of the following centers and ask for a test and measurement sales representative. *11/29/99*

United States:
Agilent Technologies
Test and Measurement Call Center
P.O.Box 4026
Englewood, CO 80155-4026
(tel) 1 800 452 4844

Canada:
Agilent Technologies Canada Inc.
5150 Spectrum Way
Mississauga, Ontario
L4W 5G1
(tel) 1 877 894 4414

Europe:
Agilent Technologies
Test & Measurement
European Marketing Organization
P.O.Box 999
1180 AZ Amstelveen
The Netherlands
(tel) (31 20) 547 9999

Japan:
Agilent Technologies Japan Ltd.
Call Center
9-1, Takakura-Cho, Hachioji-Shi,
Tokyo 192-8510, Japan
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Latin America:
Agilent Technologies
Latin American Region Headquarters
5200 Blue Lagoon Drive, Suite #950
Miami, Florida 33126
U.S.A.
(tel) (305) 267 4245
(fax) (305) 267 4286

Australia/New Zealand:
Agilent Technologies Australia Pty Ltd
347 Burwood Highway
Forest Hill, Victoria 3131
(tel) 1-800 629 485 (Australia)
(fax) (61 3) 9272 0749
(tel) 0 800 738 378 (New Zealand)
(fax) (64 4) 802 6881

Asia Pacific:
Agilent Technologies
24/F, Cityplaza One, 1111 King's Road,
Taikoo Shing, Hong Kong
(tel) (852)-3197-7777
(fax) (852)-2506-9284