

Agilent 4286A RF LCR Meter  
**Programming Manual**

**SERIAL NUMBERS**

This manual applies directly to instruments with serial number prefix JP3KC and above, or whose firmware is version 2.0.



**Agilent Technologies**  
Innovating the HP Way

Agilent Part No. 04286-90027  
Printed in JAPAN December 1999

**Fourth Edition**

---

## Notice

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

This document contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of the Agilent Technologies.

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.

MS-DOS<sup>®</sup> is a U.S. registered trademark of Microsoft Corporation.



---

## 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.

June 1995 .....	First Edition (part number: 04286-90007)
July 1995 .....	Second Edition (part number: 04286-90017)
January 1996 .....	Third Edition (part number: 04286-90027)
December 1999 .....	Fourth Edition (part number: 04286-90027)

---

## 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 CRT are enclosed in **▣**.

# Contents

---

## 1. Introduction

About This Manual . . . . .	1-1
GPIB Commands . . . . .	1-2
Common Commands . . . . .	1-2
Instrument Control Commands . . . . .	1-2
Simple Commands . . . . .	1-2
Program Message Syntax . . . . .	1-3
Command Abbreviations . . . . .	1-3
Upper and Lower Cases . . . . .	1-3
Program Message Terminator . . . . .	1-3
Multiple Messages . . . . .	1-4
Query and Response Message Syntax . . . . .	1-4
Parameters . . . . .	1-4
Variable Types . . . . .	1-4
Command Tree and Compound Header Usage . . . . .	1-6

## 2. SCPI Conformance Information

SCPI Version . . . . .	2-1
GPIB Commands Compatible to SCPI . . . . .	2-1

## 3. Command Reference

Conventions and Definitions . . . . .	3-2
ABORT . . . . .	3-3
CALCulate Subsystem . . . . .	3-4
CALCulate:BEEPper:CONDition {PASS FAIL} . . . . .	3-4
:BEEPper[:STATe] {ON OFF} . . . . .	3-4
CALCulate:COMParator Subsystem . . . . .	3-4
:CLEar . . . . .	3-4
:CONTrol:SEGment <numeric> . . . . .	3-5
:COUNt:CLEar . . . . .	3-5
:COUNt[:STATe] {ON OFF} . . . . .	3-5
:OFFSet {<numeric>} . . . . .	3-6
:SAVE . . . . .	3-6
:SEGment <numeric> . . . . .	3-6
:SEGment:ADD . . . . .	3-7
:SEGment:DELEte . . . . .	3-7
:SEGment:EDIT . . . . .	3-7
:SEGment:LOWer <numeric> . . . . .	3-7
:SEGment:SAVE . . . . .	3-7
:SEGment:UPPer <numeric> . . . . .	3-8
:SREJect:STATe {OFF ON 0 1} . . . . .	3-8
:SREJect:LOWer <numeric> . . . . .	3-8
:SREJect:UPPer <numeric> . . . . .	3-9
:SREJect:LOWer <numeric> . . . . .	3-9
:SREJect:STATe {OFF ON 0 1} . . . . .	3-9
:SREJect:UPPer <numeric> . . . . .	3-10

:STATe {OFF ON 0 1} . . . . .	3-10
CALCulate:FORMat1 {CPD CPQ CPG CPRP CSD CSQ CSRS LPD LPQ LPG LPRP LSD LSQ LSRS RX GB ZTD ZTR YTD YTR} . . . . .	3-11
CALCulate:FORMAT2{OFF ACV ACC} Subsystem . . . . .	3-12
CALCulate:LIMit{1 2} Subsystem . . . . .	3-12
:CLEar . . . . .	3-12
:SAVE . . . . .	3-12
:SEGment <numeric> . . . . .	3-12
:SEGment:ADD . . . . .	3-13
:SEGment:CONTRol[:DATA] <numeric> . . . . .	3-13
:SEGment:DELEte . . . . .	3-13
:SEGment:EDIT . . . . .	3-13
:SEGment:LOWer <numeric> . . . . .	3-14
:SEGment:SAVE . . . . .	3-14
:SEGment:UPPer <numeric> . . . . .	3-14
:STATe {OFF ON 0 1} . . . . .	3-14
CALCulate:RDC:LIMit Subsystem . . . . .	3-15
STATe{OFF ON 0 1} . . . . .	3-15
LIMit:UPPer <numeric> . . . . .	3-15
LIMit:LOWer <numeric> . . . . .	3-15
DATA Subsystem . . . . .	3-17
DATA[:DATA] <array>,{<block> <numeric11>,<numeric12>, . . . ,<numeric n2>} . . . . .	3-17
DATA[:DATA]? BCOU . . . . .	3-18
DATA[:DATA]? BIN . . . . .	3-18
DATA[:DATA]? LFA [1 2] . . . . .	3-18
DATA[:DATA]? LLIS [1 2] . . . . .	3-18
DATA[:DATA]:? {LRES1 LRES2} . . . . .	3-19
DATA[:DATA]:? RDC . . . . .	3-19
DATA[:dATA]? RDCDT . . . . .	3-19
DATA[:DATA]:? RDCL . . . . .	3-19
DATA[:DATA]? SPAR . . . . .	3-20
DATA[:DATA]:VALue? {LRES1 LRES2} <numeric> . . . . .	3-20
DATA:DEFine {OADM SIMP LIMP},{<numeric> DATA DTR } . . . . .	3-20
DATA:DELEte {OADM SIMP LIMP} . . . . .	3-21
DATA:POINt? LFA[1 2] . . . . .	3-21
DIAGnostic Subsystem . . . . .	3-22
DIAGnostic:EREFErence:STATe? . . . . .	3-22
DIAGnostic:FREVIision? . . . . .	3-22
DIAGnostic:INIT:RESult? . . . . .	3-22
DIAGnostic:SERVice Subsystem . . . . .	3-23
DIAGnostic:TEST Subsystem . . . . .	3-23
DISPlay Subsystem . . . . .	3-24
DISP:DATA:PARAMetor{1 2} <numeric> . . . . .	3-24
DISP:DATA:PARAMetor{1 2}:MSD {1-10},<numeric> . . . . .	3-24
DISP:DATA:PARAMetor{1 2}:MSD:ALL <numeric> . . . . .	3-24
DISP:DATA:PARAMetor{1 2}:MSD:AUTO . . . . .	3-25
DISP:DATA:PARAMetor{1 2}:MSD:FIX {ON OFF} . . . . .	3-25
DISPlay[:WINDow]:ALLocation {INSTrument HIHB BASic BSTatus} . . . . .	3-25
DISPlay[:WINDow]:BCOUnT:STATe {ON OFF} . . . . .	3-25
DISPlay[:WINDow]:TEXT{1-10} Subsystem . . . . .	3-26
[:DATA] <string> . . . . .	3-26
:PAGE {UP DOWN <numeric>} . . . . .	3-26
:STATe {OFF ON 0 1} . . . . .	3-27
DISPlay[:WINDow]:TRACe Subsystem . . . . .	3-28

:STATe {OFF ON 0 1} . . . . .	3-28
FORMat Subsystem . . . . .	3-29
FORMat[:DATA] {ASCIi REAL,32 REAL,64 PACKed,32} . . . . .	3-29
HCOPy Subsystem . . . . .	3-30
HCOPy . . . . .	3-30
HCOPy:ABORt . . . . .	3-30
HCOPy:DEFault . . . . .	3-30
HCOPy:DEVice:LANGUage {HPGL PCL} . . . . .	3-30
HCOPy:DEVice:SPEed {1 2} . . . . .	3-31
HCOPy[:IMMediate] . . . . .	3-31
HCOPy:ITEM Subsystem . . . . .	3-31
:ANNotation:STATe {OFF ON 0 1} . . . . .	3-31
:MENU:STATe {OFF ON 0 1} . . . . .	3-32
:TDSTamp:STATe {OFF ON 0 1} . . . . .	3-32
HCOPy:PAGE Subsystem . . . . .	3-32
:DIMensions:FULL . . . . .	3-32
:DIMensions:QUADrant{1-4} . . . . .	3-33
:SCALE {1 FULL UPPer LOWer} . . . . .	3-33
INITiate Subsystem . . . . .	3-34
INITiate:CONTinuous {OFF ON 0 1} . . . . .	3-34
INITiate[:IMMediate] . . . . .	3-34
INITiate[:IMMediate]:AGain:ALL . . . . .	3-34
MMEMory Subsystem . . . . .	3-35
MMEMory:CDIRectory [<string>] . . . . .	3-35
MMEMory:COpy {<string(s)>,<string(m_s)>,<string(d)>,<string(m_d)> } . . . . .	3-35
MMEMory:CREate:DIRectory <string> . . . . .	3-35
MMEMory:DELEte <string(file_name)>[,<string(m)>] . . . . .	3-36
MMEMory:DELEte:CAL . . . . .	3-36
MMEMory:FNAMe:EXTension{1 2} <string> . . . . .	3-36
MMEMory:INITialize <string>,{LIF DOS} . . . . .	3-37
MMEMory:LOAD Subsystem . . . . .	3-37
:CAL . . . . .	3-37
:STATe <string(file_name)>[,<string(m)>] . . . . .	3-37
:TRACe SEL,<string(file_name)>[,<string(m)>] . . . . .	3-37
MMEMory:STORe Subsystem . . . . .	3-38
:CAL:AUTO {ON OFF} . . . . .	3-38
:DIRectory:GRAPhics <string(file_name)>[,<string(m)>] . . . . .	3-38
:DIRectory:TRACe SEL,<string(file_name)>[,<string(m)>] . . . . .	3-38
:ITEM:TRACe:CATalog? . . . . .	3-39
:ITEM:TRACe:DELEte {CCO DATA DTR RAW} . . . . .	3-39
:ITEM:TRACe:SELEct {CCO DATA DTR RAW} . . . . .	3-39
:STATe <string(file_name)>[,<string(m)>] . . . . .	3-40
:TRACe SEL,<string(file_name)>[,<string(m)>] . . . . .	3-40
PROGram Subsystem . . . . .	3-41
PROGram:CATalog? . . . . .	3-41
PROGram[:SELEcted] Subsystem . . . . .	3-41
:DEFine <block> . . . . .	3-41
:DELEte[:SELEcted] . . . . .	3-42
:DELEte:ALL . . . . .	3-42
:EXECute <string> . . . . .	3-42
:MALLocate {<numeric> DEFault} . . . . .	3-43
:NAME <string> . . . . .	3-43
:NUMber <var>,<numeric1>,<numeric2> . . . . .	3-43
:STATe {RUN PAUSe STOP CONTinue} . . . . .	3-44
:STRing <var>,<string1>,<string2> . . . . .	3-44

:WAIT	3-44
PROGRAM:EXPLicit Subsystem	3-45
:DEFine "PROG",<block>	3-45
:DELeTe "PROG"	3-45
:EXECute "PROG",<string>	3-45
:MALLocate "PROG",<numeric> DEFault}	3-45
:NAME "PROG",<string>	3-45
:NUMBer "PROG",<var>,<numeric1>,<numeric2>,...,<numeric n>	3-46
:STATe "PROG",{RUN PAUSE STOP CONTinue}	3-46
:STRing "PROG",<var>,<string1>,<string2>,...,<string n>	3-46
:WAIT "PROG"	3-46
SENSE Subsystem	3-47
SENSE:CORRection1 Subsystem	3-47
:CKIT {APC35 APC7 UDEFined}	3-47
:CKIT:LABel <string>	3-47
:CKIT:SAVE	3-47
:CKIT:STANdard1:C <numeric>	3-47
:CKIT:STANdard1:G <numeric>	3-48
:CKIT:STANdard2:L <numeric>	3-48
:CKIT:STANdard2:R <numeric>	3-48
:CKIT:STANdard3:F <numeric>	3-49
:CKIT:STANdard3:L <numeric>	3-49
:CKIT:STANdard3:Q <numeric>	3-49
:CKIT:STANdard3:R <numeric>	3-49
:COLLect[:ACQuire] {STANdard1 STANdard2 STANdard3 STANdard4}	3-50
:COLLect:FPOints {FIXed USER}	3-50
:COLLect:SAVE	3-51
:EDELay:STATe {OFF ON 0 1}	3-51
:EDELay[:TIME] <numeric>	3-51
[:STATe]?	3-51
:LOAD:TYPE {RL LQF}	3-52
SENSE:CORRection2 Subsystem	3-53
:CKIT[1]:LABel <string>	3-53
:CKIT[1]:SAVE	3-53
:CKIT[1]:STANdard1:C <numeric>	3-53
:CKIT[1]:STANdard1:G <numeric>	3-53
:CKIT[1]:STANdard1[:SELeCt] {LIST LPARAmeter}	3-54
:CKIT[1]:STANdard2:L <numeric>	3-54
:CKIT[1]:STANdard2:R <numeric>	3-54
:CKIT[1]:STANdard2[:SELeCt] {LIST LPARAmeter}	3-54
:CKIT:STANdard3:F <numeric>	3-55
:CKIT:STANdard3:L <numeric>	3-55
:CKIT:STANdard3:Q <numeric>	3-55
:CKIT[1]:STANdard3:R <numeric>	3-55
:CKIT[1]:STANdard3[:SELeCt] {LIST LPARAmeter}	3-56
:COLLect[:ACQuire] STANdard{1-3}	3-56
:COLLect:FPOints {FIXed USER}	3-56
:COLLect:SAVE	3-57
:OPEN[:STATe] {OFF ON 0 1}	3-57
:SHORT[:STATe] {OFF ON 0 1}	3-57
:LOAD[:STATe] {OFF ON 0 1}	3-57
SENSE:LIST Subsystem	3-58
:CLEAr	3-58
:MODIfy:SEGMENT{1-10}:FREQUency <numeric>	3-58
:MODIfy:SEGMENT{1-10}:AVERAge:COUNT <numeric>	3-58



:SAVE	3-58
:SEGment <numeric>	3-58
:SEGment:ADD	3-59
:SEGment:AVERage:COUNT <numeric>	3-59
:SEGment:DELEte	3-59
:SEGment:EDIT	3-59
:SEGment:FREQuency <numeric>	3-59
:SEGment:QUIT	3-60
:SEGment:SAVE	3-60
SENSE:RDC [:STATe] {OFF ON 0 1}	3-60
SENSE:SWEEp:DWELl{1 2} <numeric>	3-60
SOURce Subsystem	3-62
SOURce:{CURRent POWEr}[:LEVel][:IMMEDIATE][:AMPLitude] <numeric>	3-62
SOURce:VOLTage Subsystem	3-62
[:LEVel][:IMMEDIATE][:AMPLitude] <numeric>	3-62
STATus Subsystem	3-63
STATus:INSTrument Subsystem	3-63
:ENABle <numeric>	3-63
[:EVENT]?	3-63
STATus:OPERation Subsystem	3-64
:CONDition?	3-64
:ENABle <numeric>	3-64
[:EVENT]?	3-64
:NTRansition <numeric>	3-65
:PTRansition <numeric>	3-65
STATus:PRESet	3-65
STATus:QUEStionable Subsystem	3-66
:CONDition?	3-66
:ENABle <numeric>	3-66
[:EVENT]?	3-66
SYSTem Subsystem	3-67
SYSTem:BEEPer{1 2}:STATe {OFF ON 0 1}	3-67
SYSTem:COMMunicate:GPIB Subsystem	3-67
:CONTRoller:ADDReSS <numeric>	3-67
:RDEvice{1 2}:ADDReSS <numeric>	3-68
SYSTem:COMMunicate:PARAllel[1] Subsystem	3-68
[:RECEive]:DATA?	3-68
:TRANsmit:DATA <numeric>	3-68
SYSTem:COMMunicate:PARAllel2 Subsystem	3-69
:MODE {0 1 2}	3-69
SYSTem:DATE <numeric(year)>,<numeric(month)>,<numeric(day)>	3-69
SYSTem:DATE:MODE {MDY DMY}	3-69
SYSTem:ERRor?	3-70
SYSTem:FIXTure {NONE 16191 16192 16193 16194 UDEFined}	3-70
SYSTem:FIXTure:DISTance <numeric>	3-70
SYSTem:FIXTure:LABel <string>	3-70
SYSTem:FIXTure:SAVE	3-71
SYSTem:KEY <numeric>	3-71
SYSTem:KLOCK {OFF ON 0 1}	3-71
SYSTem:PRESet	3-71
SYSTem:TIME <numeric(hour)>,<numeric(min)>,<numeric(sec)>	3-72
SYSTem:VERsion?	3-72
TRIGger Subsystem	3-73
TRIGger:EVENT:TYPE {POINT SWEEp}	3-73
TRIGger:SLOPe {POSitive NEGative}	3-73

TRIGger:SOURce {BUS EXTeRnal INTeRnal MANual}	3-73
Common Commands	3-74
*CLS	3-74
*ESE <numeric>	3-74
*ESR?	3-74
*IDN?	3-74
*OPC	3-75
*OPT?	3-75
*PCB <numeric>	3-75
*RST	3-75
*SRE <numeric>	3-76
*STB?	3-76
*TRG	3-76
*TST?	3-76
*WAI	3-76

#### 4. Using Status Reporting System

General Status Register Model	4-1
Event Register	4-2
Enable Register	4-2
Status Byte Register	4-2
Transition Filter and Condition Register	4-3
Status Register Structure	4-4
How to Use the Status Registers in a Program	4-8
Reading an Event Register Directly	4-8
SRQ and Interrupt	4-9

#### A. Manual Changes

Introduction	A-1
Manual Changes	A-1
Change 1	A-2
Serial Number	A-3

#### B. Sample Program

Example 1: BIN sort at a single frequency	B-1
Measurement Conditions	B-1
EXAMPLE 1	B-2
Example 2: Two Measurement frequency points	B-6
Measurement Conditions	B-6
Sample Program List 2	B-7
Example 3: Binary data transfer format	B-11
Measurement Conditions	B-11
Sample Program List 3	B-12
Example 4: BIN sort and Limit test at two frequency points	B-16
Measurement Conditions	B-16
Sample Program List 4	B-17
Example 5: Capacitor measurement	B-21
Measurement Conditions	B-21
Sample Program List 5	B-22

#### Messages

#### Index

## Figures

---

1-1. Program Message Terminators . . . . .	1-3
1-2. Command Tree and Compound Header Usage . . . . .	1-7
4-1. General Status Register Model . . . . .	4-1
4-2. Transition Filter and Condition Register . . . . .	4-3
4-3. Status Register Structure . . . . .	4-4
4-4. SRQ Generation Sequence . . . . .	4-10
A-1. Serial Number Plate . . . . .	A-3

## Tables

---

2-1. IEEE 488.2 Common Commands . . . . .	2-1
2-2. Instrument Control Commands . . . . .	2-2
4-1. Status Bit Definitions of the Status Byte Register . . . . .	4-5
4-2. Status Bit Definitions of the Instrument Event Status Register . . . . .	4-6
4-3. Status Bit Definitions of the Standard Event Status Register . . . . .	4-7
4-4. Status Bit Definitions of the Operation Status Register . . . . .	4-7
A-1. Manual Changes by Serial Number . . . . .	A-1
A-2. Manual Changes by Firmware Version . . . . .	A-1



# Introduction

---

## About This Manual

This manual provides a reference for the General Purpose Interface Bus (GPIB) commands used to control the 4286A RF LCR Meter (LCR meter). These commands are implemented in the LCR meter and can be used by an external controller or the HP Instrument BASIC (option 1C2).

The following is a brief description of each chapter and appendix:

- Chapter 1 provides an introduction to using GPIB commands. If you are not familiar with GPIB command syntax, start with this chapter.
- Chapter 3 explains all the GPIB command functions and their syntax.
- Chapter 4 explains the status reporting system.
- Appendix A shows manual backdate.
- Appendix B shows sample programs.
- Error Messages lists all error messages with an explanation for each error.

---

## References



- You should become familiar with the operation of the LCR meter before you attempt to control it using GPIB commands. See the following documents that are better suited to this task:

*User's Guide* (furnished to the LCR meter)

*Function Reference* (furnished to the LCR meter)

- This manual is not intended to teach the BASIC programming language or techniques. It also does not discuss GPIB theory.

For more information concerning BASIC, see the manual set for the BASIC version being used:

*BASIC Programming Techniques*

*BASIC Language Reference*

For more information concerning HP Instrument BASIC, see the following:

*HP Instrument BASIC User Handbook* (furnished to the LCR meter with option 1C2)

*HP Instrument BASIC User Handbook Supplement* (furnished to the LCR meter with option 1C2)

For more information concerning GPIB operation, see the following:

*BASIC Interfacing Techniques*

*Tutorial Description of the General Purpose Interface Bus*

*Condensed Description of the General Purpose Interface Bus*

- The LCR meter conforms to the Standard Commands for Programmable Instruments (SCPI). For more information concerning SCPI, see the following:

*Beginner's Guide to SCPI.*

---

---

## GPIB Commands

The LCR meter is equipped with the General Purpose Interface Bus (GPIB) remote programming digital interface. The GPIB is Agilent Technologies's hardware, software, documentation, and support for the IEEE 488.1, IEC-625, IEEE 488.2, and JIS-C1901 worldwide standards for interfacing instruments.

The GPIB commands implemented in the 4286A are divided into the following three categories: common commands, instrument control commands, and simple commands.

The 4286A's GPIB commands conform to the Standard Commands for Programmable Instruments (SCPI). SCPI is the new instrument command language for controlling instruments that goes beyond IEEE 488.2 to address a variety of instrument functions in a standard manner.

### Common Commands

Common commands are defined by the IEEE 488.2 standard. All common commands begin with an asterisk (\*).

For example,

```
*CLS
```

### Instrument Control Commands

Instrument control commands are defined by SCPI, and include all measurement functions and some general purpose functions. Instrument control commands consist of subsystems. Each subsystem is a set of commands that roughly corresponds to a functional block inside the instrument.

Instrument control commands have a hierarchical structure, called a **command tree**, that consists of several nodes separated by colons.

For example,

```
CALCulate:EVALuate:BAND:FULL
```

### Simple Commands

Simple commands are LCR meter-specific commands that conform to IEEE 488.2. Each simple command controls some measurement function that is normally programmed by sending multiple instrument control commands. To reduce the number of program lines and make the program simpler, these functions can also be executed by using a simple command instead of the multiple instrument control commands.

---

**Note** All GPIB commands implemented the LCR meter are listed in Chapter 2.



---

## Program Message Syntax

This section explains the construction of program messages. A **program message** is the message that you send from a computer to an instrument. Program messages consist of commands combined with appropriate punctuation and program message terminators.

### Command Abbreviations

- Many instrument control commands have a long and a short form. The short form is obtained by deleting the lower case letters. The LCR meter accepts both forms.

For example, the short form of :INITiate is :INIT and the long form of it is :INITIATE. (The LCR meter does *not* accept anything in between, such as :INITIA.)

- Some commands have a numerical suffix. The numerical suffix can be omitted, and the LCR meter recognizes that a numerical suffix of 1 is implied in this command.

For example in DISPlay:CMAP:COLor{1-14}, the numerical suffix is {1-14}. If you send DISP:CMAP:COL, it is recognized as DISP:CMAP:COL1 (the 1 is implied).

### Upper and Lower Cases

Letter cases (upper and lower) are ignored.

### Program Message Terminator

A program message must end with the **program message terminators**.

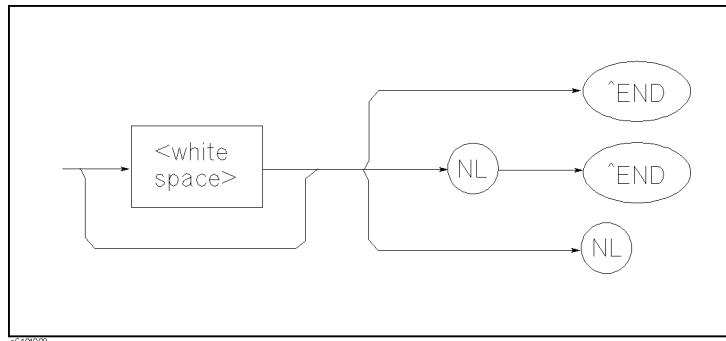


Figure 1-1. Program Message Terminators

<^END> means that End of Identify (EOI) is asserted on the GPIB interface at the same time the preceding data byte is sent.

The HP BASIC OUTPUT statement automatically sends program message terminators after the last data byte.

## Multiple Messages

To send more than one command in the same message, you must separate them with a semicolon(;):

```
SENS:FREQ:STAR 100MAHZ;STOP 1GHZ
```

For more information, see “Command Tree and Compound Header Usage”, later in this chapter.

## Query and Response Message Syntax

All commands can be queried except the commands described as “no query” in the command reference. To send a query message, add ? after the last command mnemonic.

```
SENS:FREQ:STAR?
```

A query response indicates the current setting of the LCR meter. A response message may contain both commas and semicolons as separators. When a single query command returns multiple values, a comma is used to separate each data item. When multiple queries are sent in the same message, the group of data items corresponding to each query are separated by a semicolon. For example, the fictitious query QUERY1?;QUERY2? might return a response message of:

```
<data1>,<data1>;<data2>,<data2>
```

After the message, <New Line><^END> is always sent as a response message terminator.

## Parameters

There must be a <white space>\* between the last command mnemonic (SOUR:FREQ, in the example below) and the first parameter (100MAHZ, in the example below).

```
SOUR:FREQ 100MAHZ
```

If you send more than one parameter with a single command, each parameter must be separated by a comma.

```
DATA AOFF,2
```

Each command reference contains information about the parameters available for the individual commands. There are parameters that are spelled out (for example OFF, ON, “TR1”) or parameters shown as a word enclosed in < >, that represents some value.

\* <white space> is a white space character (ASCII-encoded byte in the range of 00-09, 0B-20 (0-9, 11-32 decimal) ) or a series of the white space characters.

## Variable Types

The variable parameters used in GPIB commands are of three types: <numeric>, <string>, and <block>.

■ <numeric> represents numeric parameters as follows:

100	integer
100.0	fixed decimal point
1.0E6	floating decimal point
100.	fractional digits optional
-1.23, +235	leading signs allowed
-7.89e-01	use either E or e in exponentials
.5	digits to the left of the decimal point are optional



The LCR meter accepts *<numeric>* parameters in various formats and responds to a particular query in a predefined and fixed format.\*

The LCR meter setting programmed with a numeric parameter can assume a finite number of values, so the LCR meter automatically rounds off the parameter. For example, if you specified the OSC level as 501MV, it would be rounded off to 500MV.

\* The *<numeric>* whose absolute value is less than 1000000 is returned in the fixed decimal point format (If the value is integer, the return format is integer).

The *<numeric>* whose absolute value is, or more than 1000000 is returned in the floating decimal point format.

□ Suffix

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 <sup>1</sup>	OHM ( $\Omega$ )	U : $\mu$ ( micro; $\times 10^{-6}$ )
Admittance <sup>2</sup>	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 ( $^\circ$ ; 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 LCR meter assumes that the default suffix is sent.

- *<string>* is a string parameter that contains ASCII characters. A string must begin with a double quote (ASCII 34 decimal) and end with a double quotation mark. You can include the quotation mark as part of the string by typing it twice without any characters in between the quotation marks. Or, you can avoid typing the quotation marks twice by using a single quotation mark (ASCII 39 decimal).

Example of *<string>* TITLE,

```
OUTPUT @Meter;"DISP:TEXT10 'TITLE'"      using single quote
OUTPUT @Meter;"DISP:TEXT10 ""TITLE""""   using double quote
```

The quote to mark the beginning and end of the string is called the **delimiter**. The query response is the string with double quote delimiters.

- *<block>* is typically used to transfer large quantities of related data in binary format. *<block>* can be sent as the definite length blocks.

General form of block parameters:

```
#<num_digits><num_bytes><data bytes>
```

The single decimal digit\* <num\_digits> specifies how many digits are contained in <num\_bytes>. The decimal number <num\_bytes> specifies how many data bytes will follow in <data bytes>.

Example of <block> 1234567890

```
OUTPUT @Meter;"#2101234567890"
```

(2 means two digits follow, 10 means ten bytes follow.)

\* Digit is an ASCII-encoded byte in the range of 30-39 (48-57 decimal).

---

## Command Tree and Compound Header Usage

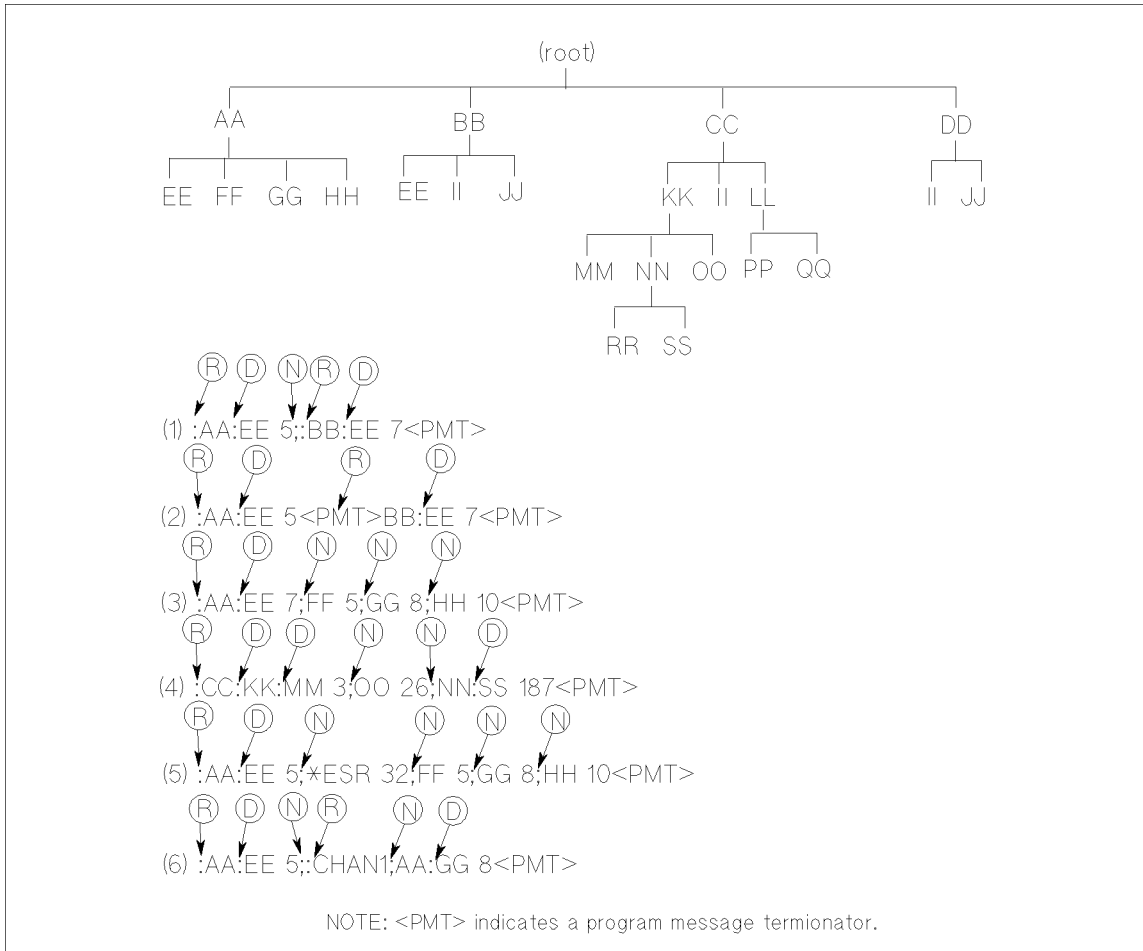
Figure 1-2 presents the instrument control command structure. The top of the command tree is called the **root**. To reach the low-level nodes, you must specify a particular path (like a DOS file directory path). Some of the notable aspects of this organization are:

1. The paths through the tree are not all the same length.
2. The number of sub-nodes under a node is not constant.
3. Node names are reused.

After Power ON or after presetting, the current path is set to the root. The path settings are changed as follows:

<b>Program Message Terminator</b>	A program message terminator (see page 1-4) sets the current path to the root.
<b>Colon (:)</b>	When a colon is placed between two command nodes, the colon moves the current path down one level on the command-tree. When the colon is the first character of a header, it specifies that the following command node is a root-level command.
<b>Semicolon (;)</b>	A semicolon separates two commands in the same message without changing the current path.
<b>Common command</b>	Common commands, such as *RST, are not part of the command tree. The parser interprets them in the same way, whatever the current path setting, and they have no effect on the current path.
<b>Simple Command</b>	Simple commands are root-level commands and they have no sub-node. The current path must be root, before and after the simple command is parsed.

Figure 1-2 shows examples of how to use the colon and semicolon to navigate efficiently through the command tree.



C6401001

**Figure 1-2. Command Tree and Compound Header Usage**

- Ⓡ sets the current path to the root.
- ⓓ moves the current path down one level.
- Ⓝ does not change the current path.

- (1) The leading colon before BB put the parser at the root.
- (2) This example works the same as example (1). A leading colon before BB is unnecessary because BB starts the parser at the root.
- (3) The entire path is not given before FF, GG, and HH.
- (4) OO 26 and NN:SS 187 are assumed to be prefixed by the implied prefix of the immediately previous command, :CC:KK.
- (5) The processing of common commands is unaffected by any previous compound commands. Also, the insertion of the common command does not affect the current path.



## SCPI Conformance Information

---

This chapter provides information about SCPI conformance. It also provides a list of all the GPIB commands implemented in the LCR meter.

---

### SCPI Version

The 4286A conforms to the SCPI 1993 standard.

---

### GPIB Commands Compatible to SCPI

Conventions	Definition
< >	Words or characters enclosed in < > brackets are used to symbolize a variable parameter.
[ ]	Words enclosed in the [ ] brackets can be omitted.
{ }	Several codes are enclosed in the { } brackets and one of these codes can be selected. For example, {OFF ON 0 1} means OFF, ON, 0, or 1. {1-4} means 1, 2, 3, or 4.

Upper case characters represent the command that must appear exactly as shown with no embedded spaces. Lower case characters can be omitted.

The following two tables list the SPCI-compatible commands implemented in the 4286A. Table 2-1 lists the IEEE 488.2 common commands. Table 2-2 lists the instrument control commands.

**Table 2-1. IEEE 488.2 Common Commands**

*CLS	*RST
*ESE <numeric>	*SRE <numeric>
*ESR?	*STB?
*IDN?	*TRG
*OPC	*TST?
*OPT	*WAI
*PCB <numeric>	

**Table 2-2. Instrument Control Commands**

COMMAND	PARAMETER	NOTES
ABORt		[no query]
[CALCulate Subsystem]		
CALCulate		
:BEEPer		
:CONDition	{PASS FAIL}	
[:STATe]	{OFF ON 0 1}	
:COMParator		
:CLEAr		
:CONTRol		
:SEGment		
:COUNt		
:CLEAr		
[:STATe]	{OFF ON 0 1}	
:OFFSet		
:OFFSet		
:SAVE		
:SEGment		
:ADD		
:DELete		
:EDIT		
:LOWer		
:SAVE		
:UPPer		
:SREJect		
:LOWer		
:STATe	{OFF ON 0 1}	
:UPPer		
:STATe	{OFF ON 0 1}	
:FORMat[1]	{CPD CPQ CPG CPRP CSD CSQ CSRS LPD LPQ LPG LPRP LSD  LSQ LSRS RX GB ZTD ZTR YTD YTR}	
:FORMat2		
:LIMit[1 2]		
:CLEAr		
:SAVE		
:SEGment		
:ADD		
:CONTRol[:DATA]		
:DELete		
:EDIT		
:LOWer		
:SAVE		
:UPPer		
:STATe	{OFF ON 0 1}	
:RDC		
:LIMit		
:STATe	{OFF ON 0 1}	
:UPPer		
:LOWer		

**Table 2-2. Instrument Control Commands (continued)**

COMMAND	PARAMETER	NOTES
[DATA Sybssystem]		
DATA		
:DATA		
:DEFne	{ OADM SIMP LIMP},{DATA DTR}	
:DELeTe	{ OADM SIMP LIMP}	
:POINt?		
LFA		
:VALue?		
	:LRES{ 1 2}	
[[:DATA]?		
:BCOU		
:BIN		
:LFA{ 1 2}		
:LLIS{ 1 2}		
:LRES{ 1 2}		
:RDC		
:RDCDT		
:RDCL		
:SPAR		
[DIAGnostic Subsystem]		
DIAGnostic		
:EREFeRence		[query only]
:STATe?		[query only]
:FREVision?		[query only]
:INIT		
:RESult?		[query only]
:SERVice		
:TEST	<numeric>	[no query]
[DISPlay Subsystem]		
DISPlay		
:DATA		
	:PARAMeter{ 1 2}	
	:MSD	{ 1-10}
	:ALL	
	:AUTO	
	:FIX	{ ON OFF}
DISPlay		
[[:WINDow]		
:ALLocation	{ INSTRument HIHB BASic BSTatus}	
:BCOUnt		
	:STATe	{ ON OFF}
:TEXT{ 1-10}		
	[[:DATA]	
	:PAGE	{ UP DOWN}
	:STATe	{ ON OFF 0 1}
:TRACe{ 1}		
	:STATe	{ OFF ON 0 1}

**Table 2-2. Instrument Control Commands (continued)**

COMMAND	PARAMETER	NOTES
[FORMat Subsystem] FORMat [:DATA]	{ASCIi REAL,32 REAL,64 PACKed,32} (ASCIi: ASCII data format REAL,32: IEEE 32-bit floating point format REAL,64: IEEE 64-bit floating point format PACKed,32: IEEE 32-bit floating point format for DOS )	
[HCOPY Subsystem] HCOPY :ABORt :DEFault :DEVice :CMAP :LANGuage :SPEEd [:IMMediate] :ITEM :ANNotation :STATe :MENU :STATe :TDSTamp :STATe :PAGE :DIMensions :FULL :QUADrant{1-4} :SCALe	{HPGL PCL} {1 2}	[no query] [no query]    [no query]       [no query] [no query]
[INITiate Subsystem] INITiate :CONTinuous [:IMMediate] :AGAIIn :ALL	{OFF ON 0 1}	[no query] [no query]



**Table 2-2. Instrument Control Commands (continued)**

COMMAND	PARAMETER	NOTES
[MMEMory Subsystem]		
MMEMory		
:CDIRectory	[<string>]	[no query]
:COPY	{<string(s)>,<string(m_s)>,<string(d)>,<string(m_d)>} <sup>1</sup>	[no query]
:CREate		
:DIRectory	<string>	[no query]
:DELeTe	<string(file_name)>[,<string(m)>]	[no query]
:CAL		
:FNAMe		
:EXTension{1 2}	<string> (1:graphics, 2:data)	
:INITialize	<string>,{LIF DOS}	[no query]
:LOAD		
:CAL		
:STATe	<string(file_name)>[,<string(m)>]	
:TRACe	SEL,<string(file_name)>[,<string(m)>]	
:STORe		
:CAL		
:AUTO	{ON OFF}	
:DINTerchange		
:GRAPhics	<string(file_name)>[,<string(m)>]	
:TRACe	SEL,<string(file_name)>[,<string(m)>]	
:ITEM		
:TRACe		
:CATalog?		[query only]
:DELeTe	{CCO DATA DTR MTR RAW}	[no query]
:SELeCt	{CCO DATA DTR MTR RAW}	[no query]
:STATe	<string(file_name)>[,<string(m)>]	
:TRACe	SEL,<string(file_name)>[,<string(m)>]	

1 (s) means the source file, and (d) means the destination file.

(m\_s) means the source mass storage, and (m\_d) means the destination mass storage.

**Table 2-2. Instrument Control Commands (continued)**

COMMAND	PARAMETER	NOTES
[PROGram Subsystem] (Option 1C2 only)		
PROGram		
:CATalog?		[query only]
[[:SELected]		
:DEFine	<block>	
:DELete		
[:SELected]		[no query]
:ALL		[no query]
:EXECute	<string>	[no query]
:MALLocate	{<numeric> DEFAult}	
:NAME	<string>	
:NUMBer	<var>,<numeric1>,<numeric2>, ... ,<numeric n>	
:STATe	{RUN PAUSE STOP CONTInue}	
:STRing	<var>,<string1>,<string2>, ... ,<string n>	
:WAIT		
:EXPLicit		
:DEFine	"PROG",<block>	
:DELete	"PROG"	[no query]
:EXECute	"PROG",<string>	[no query]
:MALLocate	"PROG",{<numeric> DEFAult}	
:NAME	"PROG",<string>	
:NUMBer	"PROG",<var>,<numeric1>,<numeric2>, ... ,<numeric n>	
:STATe	"PROG",{RUN PAUSE STOP CONTInue}	
:STRing	"PROG",<var>,<string1>,<string2>, ... ,<string n>	
:WAIT	"PROG"	

**Table 2-2. Instrument Control Commands (continued)**

COMMAND	PARAMETER	NOTES
[SENSe Subsystem]		
SENSe		
:CORRection1		
:CKIT	{APC35 APC7 UDEFined}	
:LABel	<string>	
:SAVE		[no query]
:STANdard1		
:C	<numeric>	
:G	<numeric>	
:STANdard2		
:L	<numeric>	
:R	<numeric>	
:STANdard3		
:F	<numeric>	
:L	<numeric>	
:Q	<numeric>	
:R	<numeric>	
:COLLect		
[:ACQuire]	{STANdard1 STANdard2 STANdard3 STANdard4}	
:FPOints	{FIXed USER}	
:SAVE		
:EDELay		
:STATe	{OFF ON 0 1}	
[:TIME]	<numeric>	
[:STATe]?		[query only]
:LOAD		
:TYPE	{RL LQF}	
:CORRection2		
:CKIT[1]		
:LABel	<string>	
:SAVE		
:STANdard1		
:C	<numeric>	
:G	<numeric>	
[:SELect]	{LIST LPARAmeter}	
:STANdard2		
:L	<numeric>	
:R	<numeric>	
[:SELect]	{LIST LPARAmeter}	
:STANdard3		
:F	<numeric>	
:L	<numeric>	
:Q	<numeric>	
:R	<numeric>	
[:SELect]	{LIST LPARAmeter}	
:COLLect		
[:ACQuire]		
STANdard{1-7}		
:FPOints	{FIXed USER}	
:SAVE		
:OPEN	{OFF ON 0 1}	
:SHORT	{OFF ON 0 1}	
:LOAD	{OFF ON 0 1}	

**Table 2-2. Instrument Control Commands (continued)**

COMMAND	PARAMETER	NOTES
SENSe		
:LIST		
:CLEar		[no query]
:MODIFy		
:SEGMent{1-10}		
:FREQuency		
:AVERage		
:COUNt		
:SAVE		[no query]
:SEGMent	< numeric >	
:ADD		[no query]
:AVERage		
:COUNt	< numeric >	
:DELete		[no query]
:EDIT		[no query]
:FREQuency		
:QUIT		[no query]
:SAVE		[no query]
:RDC		
[:STATe]	{ON OFF 0 1}	
:SWEep		
:DWELL{1 2}	< numeric >	
[SOURce Subsystem]		
SOURce		
:CURRent		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]		
:POWer		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]		
:VOLTage		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]		
[STATus Subsystem]		
STATus		
:INSTrument		
:ENABle	< numeric >	
[:EVENT]?		[query only]
:OPERation		
:CONDition?		[query only]
:ENABle	< numeric >	
[:EVENT]?		[query only]
:NTRansition	< numeric >	
:PTRansition	< numeric >	
:PRESet		[no query]
:QUESTionable		
:CONDition?		[query only]
:ENABle	< numeric >	
[:EVENT]?		[query only]

**Table 2-2. Instrument Control Commands (continued)**

COMMAND	PARAMETER	NOTES
[SYSTEM Subsystem]		
SYSTEM		
:BEEPPer{1 2}		
:STATe	{OFF ON 0 1}	
:COMMunicate		
:GPIB		
:CONTRoller		
:ADDRes	<numeric>	
:RDEVice{1 2}		
:ADDRes	<numeric>	
:PARAllel[1]		
[:RECEive]		
:DATA?		[query only]
:TRANsmit		
:DATA	<numeric>	
:PARAllel2		
:MODE	{0 1 2}	
:DATE	<numeric>, <numeric>, <numeric>	
:MODE	{MDY DMY}	
:ERRor?		[query only]
:FIXTure	{NONE 16191 16192 16193 16453 16454 UDEFined}	
:DISTance	<numeric>	
:LABel	<string>	
:SAVE		[no query]
:KEY	<numeric>	
:KLOCK	{OFF ON 0 1}	
:PRESet		[no query]
:TIME	<numeric(hour)>, <numeric(min)>, <numeric(sec)>	
:VERSion?		[query only]
[TRIGGER Subsystem]		
TRIGGER		
:EVENT		
:TYPE	{POINT SWEep}	
:SLOPe	{POSitive NEGative}	
:SOURce	{BUS EXTernal INTernal MANual}	



## Command Reference

---

This chapter provides a reference for the GPIB commands of the LCR meter. Use this information as a reference to the syntax requirements and general function of the individual commands.

The chapter is organized as follows:

- Instrument control commands
- Common commands
- Simple commands

The instrument control commands, common commands, and simple commands are listed in alphabetical order. See the *Function Reference* for the details of each function.

# Conventions and Definitions

The following conventions and definitions are used to describe the commands.

① → **CALCulate Subsystem**

The CALCulate subsystem controls postacquisition data processing as follows:

1. Controls the maker and equivalent circuit evaluation function (CALCulate:EVALuate).
2. Selects the measurement parameter (CALCulate:FORMat, CALCulate:FORMat:UNIT:ANGLE, CALCulate:MATH1).
3. Controls the limit test function (CALCulate:LIMit).
4. Controls trace math function (CALCulate:MATH2).

② → **CALCulate:EVALuate**

**Subsystem**

This subsystem is used to set the marker and equivalent circuit evaluation function.

Related Command      DISPlay[:WINDow]:TRACe{1-17}:MARKer{1-8}

③ → **:BAND:FULL[:STATE] {OFF|ON|0|1}**

④ → CALCulate:EVALuate:BAND:FULL[:STATE] {OFF|ON|0|1}

⑤ → Sets the partial search of the marker search function ON or OFF. (PART SRCH ON off under [Search](#))

Parameter	Description
OFF or 0	Partial search ON.
ON or 1	Partial search OFF.

Query Response is {0|1}.

①	<p>Section header</p> <p>The GPIB commands are divided into the following sections.</p> <p>Sections which are top of the command tree (The sections are provided in alphabetical order.)</p> <p>Common commands section</p> <p>Simple commands section</p>
②	<p>Lower level subsystem and its description</p> <p>Describes a lower level subsystem. This subsystem contains functions that are grouped together.</p>
③	<p>Command nodes and required parameter</p> <p>Upper case bold characters represent the command that must appear exactly as shown with no embedded spaces. Lower case bold characters can be omitted.</p> <p>Words or characters enclosed in &lt; &gt; brackets are used to symbolize a parameter type.</p> <p>Words enclosed in the [ ] brackets can be omitted.</p> <p>Several codes are enclosed in the { } brackets and one of these codes can be selected. For example, {OFF ON 0 1} means OFF, ON, 0, or 1. {1-4} means 1, 2, 3, or 4.</p> <p>For example, this command usage is:<sup>1</sup></p> <pre>OUTPUT @Hp4286;"CALC:EVAL:BAND:FULL ON"</pre>
④	<p>Command spelled out from the root</p>
⑤	<p>Command description and parameter description</p>

<sup>1</sup> In this manual, example statements use @Hp4286 as the GPIB address. Therefore, @Hp4286 must be assigned as the LCR meter GPIB address in a preceding program statement.



---

## ABORt

ABORt

Reset the trigger system and place all trigger sequences in the idle state. (**SWEEP: HOLD** under **Trigger Mode**);No query)

Related Command      INITiate:CONTinuous {OFF|ON|0|1}  
                          INITiate[:IMMediate]

Example                To abort the measurement and hold the sweep immediately,

```
OUTPUT @Hp4286"INIT:CONT OFF"  
OUTPUT @Hp4286"ABORt"
```

---

## CALCulate Subsystem

The CALCulate subsystem controls post acquisition data processing as follows:

1. Controls the BIN sort function (CALCulate:COMParator).
2. Selects the measurement parameter (CALCulate:FORMat).
3. Controls the limit test function (CALCulate:limit{1|2}).

### **CALCulate:BEEPer:CONDition {PASS|FAIL}**

CALCulate:BEEPer:CONDition {PASS|FAIL}

Sets comparator beeper. (BEEP PASS FAIL under Test Setup)

Parameter	Description
PASS	Beeps when comparator result passes. (PASS)
FAIL	Beeps when comparator result fails. (FAIL)

Query response is {PASS|FAIL}.

Example

```
OUTPUT @Hp4286;"CALC:BEEP:COND PASS"
OUTPUT @Hp4286;"CALC:BEEP ON"

OUTPUT @Hp4286;"CALC:BEEP:COND?"
ENTER @Hp4286;A,B
```

### **:BEEPer[:STATE] {ON|OFF}**

CALCulate:BEEPer:CONDition {PASS|FAIL}

Sets comparator beeper ON or OFF. (BEEP: OFF under Test Setup)

## CALCulate:COMParator

## Subsystem

---

This subsystem is used to set the comparator function, including its ON/OFF setting, limit mode, limit value, and segment configuration.

### **:CLEAr**

CALCulate:COMParator:CLEAr

Clears all BIN counts. (EDIT BIN TABLE CLEAR LIST YES under Test Setup); No query)

**:CONTRol:SEGMENT <numeric>**

CALCulate:COMParator:CONTRol:SEGMENT <numeric>

Set segment number.

Parameter	Range	Unit
<numeric>	1 to 10	

Query Response is <numeric>.

Example                    OUTPUT @Hp4286;"CALC:COMP:CONT:SEGM 1"  
                               OUTPUT @Hp4286;"CALC:COMP:CONT:SEGM?"  
                               ENTER @Hp4286;A

**:COUNT:CLEAr**

CALCulate:COMParator::COUNT:CLEAr

Set BIN counter to 0. (CLEAR BIN COUNT under **Test Setup**);No Query)

Example                    OUTPUT @Hp4286;"CALC:COMP:COUN:CLE"

**:COUNT[:STATe] {ON|OFF}**

CALCulate:COMParator:COUNT[:STATe] {ON|OFF}

Set BIN Count function ON or OFF. (BIN COUNT on OFF under **Test Setup** )

Parameter	Description
ON	BIN Count ON
OFF	BIN Count OFF

Query response is {1|0}

Example                    OUTPUT @Hp4286;"CALC:COMP:COUN ON"  
                               OUTPUT @Hp4286;"CALC:COMP:COUN?"  
                               ENTER @Hp4286;A

## CALCulate:COMParator

### :OFFSet {<numeric>}

CALCulate:COMParator:OFFSet {<numeric>}

Add or or subtract the offset value to/from amplitude of BIN sort. (AMPLITUDE OFFSET under (System))

Parameter	Range or Description	Unit
<numeric>	-1x10 <sup>9</sup> to 1x10 <sup>9</sup>	Y-axis unit

Query response is <numeric>

Example  
OUTPUT @Hp4286;"CALC:COMP:OFFS 0"  
OUTPUT @Hp4286;"CALC:COMP:OFFS?"  
ENTER @Hp4286;A

### :SAVE

CALCulate:COMParator:SAVE

Completes editing the BIN table. (EDIT BIN SORT under (Test Setup); No query)

### :SEGMENT <numeric>

CALCulate:COMParator:SEGMENT <numeric>

Specifies the BIN count to edit. (EDIT BIN SORT under (Test Setup))

Parameter	Description
<numeric>	Segment number, 1 to 9.
No parameter	Current segment

Query Response is <numeric>.

Related Command      CALCulate:COMParator:SEGMENT:EDIT

Example  
To edit a segment,  
OUTPUT @Hp4286;"CALC:COMP:SEGM 1"  
OUTPUT @Hp4286;"CALC:COMP:SEGM:EDIT"  
OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 100"  
OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 80"  
OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"  
Query,  
OUTPUT @Hp4286;"CALC:COMP:SEG?"  
ENTER @Hp4286;A

**:SEGMENT:ADD**

CALCulate:COMParator:SEGMENT:ADD

Adds a new count to the end of the BIN counts. (EDIT BIN TABLE ADD under **Test Setup**); No query)

**:SEGMENT:DELEte**

CALCulate:COMParator:SEGMENT:DELEte

Deletes a BIN count. (EDIT BIN TABLE DELETE under **Test Setup**); No query)

**:SEGMENT:EDIT**

CALCulate:COMParator:SEGMENT:EDIT

Start editing the BIN count. (EDIT BIN TABLE EDIT under **Test Setup**); No query)

Related Command      CALCulate:COMParator:SEGMENT

**:SEGMENT:LOWer <numeric>**

CALCulate:COMParator:SEGMENT:LOWer &lt;numeric&gt;

Sets the lower limit value for the BIN count. (EDIT BIN TABLE EDIT LOWER LIMIT under **Test Setup**)

Parameter	Range	Unit
<numeric>	$-1 \times 10^9$ to $1 \times 10^9$	y-axis unit

Query Response is <numeric>.

```
Example                    OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 0"
                          OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW?"
                          ENTER @Hp4286;A
```

**:SEGMENT:SAVE**

CALCulate:COMParator:SEGMENT:SAVE

Terminates defining BIN counts. (EDIT BIN TABLE DONE under **Test Setup**); No query)

Related Command      CALCulate:COMParator:SEGMENT  
                          CALCulate:COMParator:SEGMENT:EDIT

### **:SEGMent:UPPer <numeric>**

CALCulate:COMParator:SEGMent:UPPer

Sets the upper limit value for a BIN count. (EDIT BIN TABLE EDIT UPPER LIMIT under **Test Setup**)

Parameter	Range	Unit
<numeric>	-1×10 <sup>9</sup> to 1×10 <sup>9</sup>	y-axis unit

Query Response is <numeric>.

Example                                    OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 0"  
  OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP?"  
  ENTER @Hp4286;A

### **:SREJect:STATe {OFF|ON|0|1}**

CALCulate:COMParator:SREJect:STATe {OFF|ON|0|1}

Sets the comparator secondary rejection parameter ON or OFF.

Parameter	Description
OFF or 0	Secondary rejection parameter OFF
ON or 1	Secondary rejection parameter ON

Query Response is {0|1}.

Example                                    OUTPUT @Hp4286;"CALC:COMP:SREJ:STAT ON"  
  OUTPUT @Hp4286;"CALC:SREJ:STAT?"  
  ENTER @Hp4286;A

### **:SREJect:LOWer <numeric>**

CALCulate:COMParator:SREJect:LOWer <numeric>

Sets the lower limit value for the secondary rejection parameter.

Parameter	Range	Unit
<numeric>		y-axis unit

Query Response is <numeric>.

Example                                    OUTPUT @Hp4286;"CALC:COMP:SREJ:LOW 0"  
  OUTPUT @Hp4286;"CALC:COMP:SREJ:LOW?"  
  ENTER @Hp4286;A

**:SREJect:UPPer <numeric>**

CALCulate:COMParator:SREJect:UPPer &lt;numeric&gt;

Sets the upper limit value for the secondary rejection parameter.

Parameter	Range	Unit
<numeric>		y-axis unit

Query Response is &lt;numeric&gt;.

Example                    OUTPUT @Hp4286;"CALC:COMP:SREJ:UPP 10"  
                               OUTPUT @Hp4286;"CALC:COMP:SREJ:UPP?"  
                               ENTER @Hp4286;A

**:SREJect:LOWer <numeric>**

CALCulate:COMParator:SREJect:LOWer &lt;numeric&gt;

Set lower limit of the secondary parameter of the comparator function.

(SEC REJECT MENU LOWER LIMIT under (Test Setup) )

Parameter	Range	Unit
<numeric>	$-1 \times 10^9$ to $1 \times 10^9$	Y-axis unit

Query response is &lt;numeric&gt;.

Example                    OUTPUT @Hp4286;"CALC:COMP:SREJ:LOW 0"  
                               OUTPUT @Hp4286;"CALC:COMP:SREJ:LOW?"  
                               ENTER @Hp4286;A

**:SREJect:STATe {OFF|ON|0|1}**

CALCulate:COMParator:SREJect:STATe {OFF|ON|0|1}

Set the comparator of the scndary parameter ON or OFF. (SEC REJECT on OFF under (Test Setup) )

Parameter	Description
OFF or 0	Secondary parameter OFF
ON or 1	Secondary parameter ON

Query response is {0|1}.

Example                    OUTPUT @Hp4286;"CALC:COMP:SREJ:STAT ON"  
                               OUTPUT @Hp4286;"CALC:COMP:SREJ:STAT?"  
                               ENTER @Hp4286;A

## CALCulate:COMParator

### **:SREJect:UPPer** <numeric>

CALCulate:COMParator:SREJect:UPPer

Set the upper parameter of the secondary paramter of the comparator function.  
(**SEC REJECT MENU UPPER LIMIT** under **Test Setup**)

Parameter	Range	Unit
<numeric>	-1x10 <sup>9</sup> to 1x10 <sup>9</sup>	Y-axis unit

Query response is <numeric>.

Example                                    OUTPUT @Hp4286;"CALC:COMP:SREJ:UPP 0"  
    OUTPUT @Hp4286;"CALC:COMP:SREJ:UPP?"  
    ENTER @Hp4286;A

### **:STATe** {OFF|ON|0|1}

CALCulate:COMParator:STATe {OFF|ON|0|1}

Sets the comparator function ON or OFF. (**BIN SORT ON off** under **Test Setup**)

Parameter	Description
OFF or 0	Comparator OFF
ON or 1	Comparator ON

Query Response is {0|1}.

Example                                    OUTPUT @Hp4286;"CALC:COMP:STAT ON"  
    OUTPUT @Hp4286;"CALC:COMP:STAT?"  
    ENTER @Hp4286;A



**CALCulate:FORMat1** {CPD|CPQ|CPG|CPRP|CSD|CSQ|CSRS|LPD|LPQ|LPG|LPRP|LSD|LSQ|LSRS|RX|GB|ZTD|ZTR|YTD|YTR}

Set the combination of measurement parameters. It formats vector measurement data, Z (impedance) and Y (admittance), into scalar measurement parameters, such as  $|Z|$ ,  $\theta$ , inductance, capacitance, and so on.

Parameter	Description	
	Primary parameter	Secondary parameter
CPD	Equivalent parallel capacitance	Dissipation factor
CPQ		Quality factor
CPG		Conductance
CPRP	Equivalent series capacitance	Equivalent parallel resistance
CSD		Dissipation factor
CSQ		Quality factor
CSRS	Equivalent parallel inductance	Equivalent series resistance
LPD		Dissipation factor
LPQ		Quality factor
LPG	Equivalent series inductance	Conductance
LPRP		Equivalent parallel resistance
LSD		Dissipation factor
LSQ	Resistance	Quality factor
LSRS		Equivalent series resistance
RX		Reactance
GB	Conductance	Suceptance
ZTD	$ Z$ (impedance)	$\theta$ (deg)
ZTR		$\theta$ (rad)
YTD	$ Y$ (admittance)	$\theta$ (deg)
YTR		$\theta$ (rad)

Query Response is

{CPD|CPQ|CPG|CPRP|CSD|CSQ|CSRS|LPD|LPQ|LPG|LPRP|LSD|LSQ|LSRS|RX|GB|ZTD|ZTR|YTD|YTR}.

Related Command

CALCulate:MATH1[:EXPRession]:NAME  
 CALCulate:MATH1:STATe  
 DISPlay[:WINDow]:TRACe{1-21}:GRATicule:FORMat  
 DISPlay[:WINDow]:TRACe{1-21}:GRATicule:Y:SPACing

Example

To display the admittance chart,

```
OUTPUT @Hp4286;"CALC:MATH1:NAME ADM;STAT ON"
OUTPUT @Hp4286;"CALC:FORM COMP"
OUTPUT @Hp4286;"DISP:TRAC:GRAT:FORM ADM"
```

Query,

```
OUTPUT @Hp4286;"CALC:FORM?"
ENTER @Hp4286;A$
```

## CALCulate:FORMat

### CALCulate:FORMAT2{OFF|ACV|ACC}

### Subsystem

Displays the voltage or current level applied to the DUT at the marker point. (AC-V or AC-I under Monitor)

Parameter	Description
OFF	Displays the measurement data (OFF; level monitor OFF).
ACV	Displays the AC voltage level (ACV).
ACC	Displays the AC current level (ACC).

Query response is {"OFF"|"ACV"|"ACC"}.

Related Command      DATA[:DATA]? MON

### CALCulate:LIMit{1|2}

### Subsystem

This subsystem is used to set the limit test function. Use LIMit1 to set the function for the primary parameter, and LIMit2 for the secondary parameter.

#### :CLEar

Calculate:limit{1|2}:CLEar

Clears all segments in the limit line. (CLEAR LIST YES under System); No query)

#### :SAVE

Calculate:limit{1|2}:SAVE

Completes editing the limit table. (DONE under System); No query)

#### :SEGMENT <numeric>

Calculate:limit{1|2}:SEGMENT <numeric>

Specifies which limit segment in the table to edit. (SEGMENT under System)

Parameter	Description
<numeric>	Segment number, 1 to 9.
No parameter	Current segment

Query Response is <numeric>.

Related Command      Calculate:limit{1|2}:SEGMENT:EDIT

Example

To edit a segment,

```
OUTPUT @Hp4286;"CALC:LIM:SEGM 1"
OUTPUT @Hp4286;"CALC:LIM:SEGM:EDIT"
OUTPUT @Hp4286;"CALC:LIM:SEGM:CONT 10MAHZ"
OUTPUT @Hp4286;"CALC:LIM:SEGM:UPP 100"
OUTPUT @Hp4286;"CALC:LIM:SEGM:LOW 80"
OUTPUT @Hp4286;"CALC:LIM:SEGM:SAVE"
```

### :SEGMent:ADD

Calcculate:limit{1|2}:SEGMent:ADD

Adds a new segment to the end of the limit table. (ADD under System); No query)

### :SEGMent:CONTRol[:DATA] <numeric>

Calcculate:limit{1|2}:SEGMent:CONTRol[:DATA] <numeric>

Sets the stimulus value of a segment. (STIMULUS under System)

Parameter	Range	Unit
<numeric>	1000000 (= 1 M) to 1x 10 <sup>9</sup> (= 1 G)	Hz (frequency sweep)

Query Response is <numeric>.

Example

```
OUTPUT @Hp4286;"CALC:LIM1:SEGM:CONT 1.E6"
OUTPUT @Hp4286;"CALC:LIM1:SEGM:CONT?"
ENTER @Hp4286;A
```

### :SEGMent:DELeTe

Calcculate:limit{1|2}:SEGMent:DELeTe

Deletes a limit testing segment. (DELETE under System); No query)

### :SEGMent:EDIT

Calcculate:limit{1|2}:SEGMent:EDIT

Start editing the segment. (EDIT under System);No query)

Related Command

Calcculate:limit{1|2}:SEGMent

### **:SEGMENT:LOWer <numeric>**

Calculate:limit{1|2}:SEGMENT:LOWer <numeric>

Sets the lower limit value for the segment. (LOWER LIMIT under (System))

Parameter	Range	Unit
<numeric>	$-1 \times 10^9$ to $1 \times 10^9$	y-axis unit

Query Response is <numeric>.

Example

```
OUTPUT @Hp4286;"CALC:LIM1:SEGM:LOW 0"
OUTPUT @Hp4286;"CALC:LIM1:SEGM:LOW?"
ENTER @Hp4286;A
```

### **:SEGMENT:SAVE**

Calculate:limit{1|2}:SEGMENT:SAVE

Terminates a limit segment definition. (DONE under (System); No query)

Related Command

```
Calculate:limit{1|2}:SEGMENT
Calculate:limit{1|2}:SEGMENT:EDIT
```

### **:SEGMENT:UPPer <numeric>**

Calculate:limit{1|2}:SEGMENT:UPPer

Sets the upper limit value for a limit testing segment. (UPPER LIMIT under (System))

Parameter	Range	Unit
<numeric>	$-1 \times 10^9$ to $1 \times 10^9$	y-axis unit

Query Response is <numeric>.

Example

```
OUTPUT @Hp4286;"CALC:LIM1:SEGM:UPP 0"
OUTPUT @Hp4286;"CALC:LIM1:SEGM:UPP?"
ENTER @Hp4286;A
```

### **:STATe {OFF|ON|0|1}**

Calculate:limit{1|2}:STATe {OFF|ON|0|1}

Sets the limit testing ON or OFF. (LIMIT TEST ON off under (System))

Parameter	Description
OFF or 0	Limit testing OFF
ON or 1	Limit testing ON

Query Response is {0|1}.

Example                    OUTPUT @Hp4286;"CALC:LIM1:STAT ON"  
                               OUTPUT @Hp4286;"CALC:LIM1:STAT?"  
                               ENTER @Hp4286;A

**CALCulate:RDC:LIMit****Subsystem**

This subsystem is used to set the contact check function.

**STATe{OFF|ON|0|1}**

CALCulate:RDC:LIMit:STATe {OFF|ON|0|1}

Sets the contact check ON or OFF. (RDC LIMIT ON OFF under **Contact Check**)

Parameter	Description
OFF or 0	Contact check OFF
ON or 1	Contact check ON

Query Response is {0|1}.

Example                    OUTPUT @Hp4286;"CALC:RDC:LIM:STAT ON"  
                               OUTPUT @Hp4286;" CALC:RDC:LIM:STAT?"  
                               ENTER @Hp4286;A

**LIMit:UPPer <numeric>**

CALCulate:RDC:LIMit:UPPer

Sets the upper value of a contact check. (RDC UPPER LIMIT under **Contact Check**)

Parameter	Range	Unit
<numeric>	0 to 10 k	Ω

Query Response is <numeric>.

Example                    OUTPUT @Hp4286;"CALC:RDC:LIM:UPP 100"  
                               OUTPUT @Hp4286;" CALC:RDC:LIM:UPP?"  
                               ENTER @Hp4286;A

**LIMit:LOWer <numeric>**

CALCulate:RDC:LIMit:LOWer

Sets the lower value of a contact check. (RDC LOWER LIMIT under **Contact Check**)

Parameter	Range	Unit
<numeric>	0 to 10 k	Ω

Query Response is <numeric>.

## **CALCulate:RDC**

Example

```
OUTPUT @Hp4286;"CALC:RDC:LIM:LOW 100"  
OUTPUT @Hp4286;" CALC:RDC:LIM:LOW?"  
ENTER @Hp4286;A
```

## DATA Subsystem

The DATA subsystem is used to send or get the following values:

- The parameters used for display offset.
- The raw data arrays, data arrays, memory arrays, and fixture compensation standard arrays, which are complex data.
- The limit test results.
- The level monitor data array.
- The stimulus array.
- The stimulus, data, and memory of the specified point.

**DATA[:DATA] <array>, {<block>|<numeric11>, <numeric12>, ... , <numeric n2>}**

DATA[:DATA] <array>\_i {<block>|<numeric11>\_i, <numeric12>\_i, ..., <numeric n1>\_i, <numeric n2>\_i}

Send data arrays.

Parameter	Description
<array>	CCO{11-13}, CMP{1-3}, DATA, RAW, OADM, SIMP, LIMP, MON, DTR

Where,

CCO11, CCO12, CCO13	: Calibration coefficient arrays, A <sub>1</sub> , B <sub>1</sub> , C <sub>1</sub> at normal-vl range (real-imaginary)
CMP1, CMP2, CMP3	: Compensation coefficient arrays, A <sub>3</sub> , B <sub>3</sub> , C <sub>3</sub> (real-imaginary) <sup>1</sup>
DATA	: Data array (real-imaginary)
RAW	: Raw data array (real-imaginary)
OADM	: Fixture compensation open standard array (real-imaginary)
SIMP	: Fixture compensation short standard array (real-imaginary)
LIMP	: Fixture compensation load standard array (real-imaginary)
MON	: Level monitor array (scalar) <sup>2</sup>
DTR	: Data trace array

1 Before executing the query command, set the compensation function ON.

2 When the level monitor function is OFF, the value is not updated. Before query, set the level monitor ON.

Parameter	Description
<block>	Binary data transfer format
<numeric11>, ... , <numeric n2>	ASCII data transfer format (real-imaginary)
<numeric1>, ... , <numeric n>	ASCII data transfer format (scalar)

The array dimension is defined as:

CCO{11-13}, CMP{1-3}, DATA, RAW	: n (the number of measurement points) × 2 (real-imaginary)
OADM, SIMP, LIMP	: n (the number defined by DATA:DEFine) × 1 (scalar)
MON	: n (the number of measurement points) × 1 (scalar)

Query response is:

{<block>|<numeric11>, <numeric12>, ... , <numeric n1>, <numeric n2>} (real-imaginary).

{<block>|<numeric1>, ... , <numeric n>} (scalar).

## DATA[:DATA]

Example                    DIM A(1:201,1:2)  
  
                          OUTPUT @Hp4286;"DATA? CC011"  
  
                          ENTER @Hp4286;A(\*)

## DATA[:DATA]? BCOU

DATA[:DATA]? BCOU

Returns the BIN count result. (Query only)

## DATA[:DATA]? BIN

DATA[:DATA]? BIN

Returns the BIN sort test results. (Query only)

Query response is {0|1-9}.

Where,

-1	No test
0	Out of BIN
1-9	The BIN number in which the measurement failed.
10	AUX BIN

## DATA[:DATA]? LFA [1|2]

DATA[:DATA]? LFA [1|2]

Returns the limit test result of the fail points. (Query only)

Query response is  $\langle \text{numeric1}(\text{stim}) \rangle, 0, \langle \text{numeric1}(\text{upp}) \rangle, \langle \text{numeric1}(\text{low}) \rangle, \langle \text{numeric2}(\text{stim}) \rangle, \dots, \langle \text{numeric } n(\text{low}) \rangle$ . (ASCII data transfer format)

Where,

$\langle \text{numeric1}(\text{stim}) \rangle, \dots, \langle \text{numeric } n(\text{stim}) \rangle$	: Stimulus at each fail point.
$\langle \text{numeric1}(\text{upp}) \rangle, \dots, \langle \text{numeric } n(\text{upp}) \rangle$	: Upper limit value at each fail point.
$\langle \text{numeric1}(\text{low}) \rangle, \dots, \langle \text{numeric } n(\text{low}) \rangle$	: Lower limit value at each fail point.

The array dimension is defined as,

n (the number of limit test fail points)  $\times$  4 (stimulus, 0, upper limit value, lower limit value)

## DATA[:DATA]? LLIS [1|2]

DATA[:DATA]? LLIS

Returns the stimulus value, the limit test result, and limit values of all measurement points. (Query only)

Query response is  $\langle \text{numeric1}(\text{stim}) \rangle, \{-1|0|1\}, \langle \text{numeric1}(\text{upp}) \rangle, \langle \text{numeric1}(\text{low}) \rangle, \langle \text{numeric2}(\text{stim}) \rangle, \dots, \langle \text{numeric } n(\text{low}) \rangle$ . (ASCII data transfer format)



## DATA[:DATA]?

Where,

$\langle \text{numeric1}(\text{stim}) \rangle, \dots, \langle \text{numeric } n(\text{stim}) \rangle$  : Stimulus at each measurement point.  
 $\{-1|0|1\}$  : -1 means NO TEST, 0 means FAIL, and 1 means PASS.  
 $\langle \text{numeric1}(\text{upp}) \rangle, \dots, \langle \text{numeric } n(\text{upp}) \rangle$  : Upper limit value at each point.  
 $\langle \text{numeric1}(\text{low}) \rangle, \dots, \langle \text{numeric } n(\text{low}) \rangle$  : Lower limit value at each point.

The array dimension is defined as,

n (the number of the measurement points)  $\times$  4 (stimulus,  $\{-1|0|1\}$ , upper limit value, lower limit value)

### DATA[:DATA]:? {LRES1|LRES2}

DATA[:DATA]? {LRES1|LRES2}

Returns the limit test result of all measurement points. (Query only)

Query response is  $\{-1|0|1\}, \{-1|0|1\}, \{-1|0|1\}, \dots, \{-1|0|1\}$

Where,

-1	No test
0	Fail
1	Pass

### DATA[:DATA]:? RDC

DATA[:DATA]? RDC

Returns the Rdc measurement value. (Query only)

Query response is  $\langle \text{numeric} \rangle$ .

### DATA[:dDATA]? RDCDT

DATA[:dDATA]? RDCDT Returns the contact check result, the Rdc measurement value and the data trace array. (Query only)

Query response is  $\{-1|0|1\}, \langle \text{numeric}(\text{Rdc}) \rangle, \langle \text{numeric1}(\text{primary}) \rangle, \langle \text{numeric1}(\text{secondary}) \rangle, \dots, \langle \text{numeric } n(\text{primary}) \rangle, \langle \text{numeric } n(\text{secondary}) \rangle$

where

-1	No test
0	Fail
1	Pass

$\langle \text{numeric}(\text{Rdc}) \rangle$  : Rdc measurement value  
 $\langle \text{numeric1}(\text{primary}) \rangle, \dots, \langle \text{numeric } n(\text{primary}) \rangle$  : primary parameter measurement value  
 $\langle \text{numeric1}(\text{secondary}) \rangle, \dots, \langle \text{numeric } n(\text{secondary}) \rangle$  : secondary parameter measurement value

When the limit result is 0, all query responses are 0.

### DATA[:DATA]:? RDCL

DATA[:DATA]? RDCL

Returns the contact check result. (Query only)

Query response is  $\{-1|0|1\}$

-1	No test
0	Fail
1	Pass

## DATA[:DATA]? SPAR

DATA[:DATA]? {MON|SPAR}

Returns the stimulus array (scalar). (Query only)

Query response is {<block>|<numeric1>,<numeric2>,...,<numeric n>}.

Where,

*<block>* : Binary data transfer format  
*<numeric1>*, ... ,*<numeric n>* : ASCII data transfer format

The array dimension is defined as,

n (the number of the measurement points) × 1

## DATA[:DATA]:VALue? {LRES1|LRES2} <numeric>

DATA[:DATA]? {LRES1|LRES2}

Returns the limit test result at the measurement point specified by the parameter. (Query only)

Query response is {-1|0|1}

Where,

-1	No test
0	Fail
1	Pass

## DATA:DEFine {OADM|SIMP|LIMP},{<numeric>|DATA|DTR }

DATA:DEFine {OADM|SIMP|LIMP},{<numeric>|DATA|DTR}

Sets the fixture compensation standard arrays.

Parameter	Description
OADM	Fixture compensation open standard array
SIMP	Fixture compensation short standard array
LIMP	Fixture compensation load standard array

Parameter	Range or Description
<numeric>	1 to number of measurement points. Sets the number of data points that are sent by DATA[:DATA] {OADM SIMP LIMP}. The fixture compensation standard array is set by DATA[:DATA] {OADM SIMP LIMP}.
DATA	Copies data array to the fixture compensation standard arrays.
DTR	Copies data trace array to the fixture compensation standard arrays.

Query response is <numeric>.

```
Example      OUTPUT @Hp4286;"DATA:DEF OADM,DATA"  
            OUTPUT @Hp4286;"DATA:DEF? OADM"  
  
            ENTER @Hp4286;A
```

**DATA:DELEte {OADM|SIMP|LIMP}**

DATA:DELEte {OADM|SIMP|LIMP}

Clears the fixture compensation standard array. (No query)

After clear, SENSE:CORRection2:CKIT[1]:STANDard{1-3}[:SElect] LPARameter is selected.

Parameter	Description
OADM	Clears the fixture compensation open standard array.
SIMP	Clears the fixture compensation short standard array.
LIMP	Clears the fixture compensation load standard array.

**DATA:POINT? LFA[1|2]**

DATA:POINT? LFA[1|2]

Returns the number of the limit test fail points. (Query only)

Query response is <numeric>.

---

## DIAGnostic Subsystem

This subsystem provides the LCR meter's hardware and firmware tests, and the service related functions for maintenance and repair.

### DIAGnostic:EREFerence:STATe?

DIAGnostic:EREFerence:STATe?

Queries whether the external reference signal is connected to the external reference input on the rear panel. (Query only)

Query response is {0|1}.

Where,

0 : Not connected.  
1 : Connected.

Example                    OUTPUT @Hp4286;"DIAG:EREF:STAT?"  
                          ENTER @Hp4286;A

### DIAGnostic:FREVision?

DIAGnostic:FREVision?

Returns the model number (4286A) and the firmware revision (for example, REV1.00). (Query only)

Query response is <string>.

Example                    OUTPUT @Hp4286;"DIAG:FREV?"  
                          ENTER @Hp4286;A\$

### DIAGnostic:INIT:RESult?

DIAGnostic:INIT:RESult?

Returns the power-on test result. (Query only)

Query response is {PASS|FAIL}.

Where,

PASS : Power-on test passed.  
FAIL : Power-on test failed.

Example                    OUTPUT @Hp4286;"DIAG:INIT:RES?"  
                          ENTER @Hp4286;A\$

**DIAGnostic:SERVice**

**Subsystem**

---

This subsystem provides the required functions for maintenance and repair. This manual does not describe the DIAGnostic:SERVice subsystem. See the *Service Manual* for the command and function descriptions. ( SERVICE under (System) SERVICE MENU )

**DIAGnostic:TEST**

**Subsystem**

---

This subsystem provides the self-tests for maintenance and repair. This manual does not describe the DIAGnostic:TEST subsystem. See the *Service Manual* for the command and function descriptions. ( TESTS under (System) SERVICE MENU )

---

## DISPlay Subsystem

### DISP:DATA:PARAMeter{1|2} <numeric>

DISP:DATA:PARAMeter{1|2} <numeric>

Set decimal places of display. (PRI DIGIT, SEC DIGIT under **Display**)

Parameter	Description
1	Primary Parameter
2	Secondary Parameter

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"DISP:DATA:PARA1 3"  
                          OUTPUT @Hp4286;"DISP:DATA:PARA1?"  
                          ENTER @Hp4286;A

### DISP:DATA:PARAMeter{1|2}:MSD {1-10},<numeric>

DISP:DATA:PARAMeter{1|2}:MSD {1-10},<numeric>

Set number of digits to be displayed for each segment.

(SEG 1 , SEG 2 , SEG 3 , ... , SEG 10  under **Display**)

Node	Description
PAR1	Primary Parameter
PAR2	Secondary Parameter

Parameter	Description
1 to 10	Segment Number

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"DISP:DATA:PARA1:MSD 1,1.OE-3"  
                          OUTPUT @Hp4286;"DISP:DATA:PARA1:MSD? 1"  
                          ENTER @Hp4286;A

### DISP:DATA:PARAMeter{1|2}:MSD:ALL <numeric>

DISP:DATA:PARAMeter{1|2}:MSD:ALL <numeric>

Set number of digits to be displayed of all segments. (**MODIFY ALL MSD** under **Display**)

Node	Description
PAR1	Primary Parameter
PAR2	Secondary Parameter

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"DISP:DATA:PARA1:MSD:ALL 1.OE-3"  
                          OUTPUT @Hp4286;"DISP:DATA:PARA1:MSD:ALL?"  
                          ENTER @Hp4286;A

**DISP:DATA:PARAMeter{1|2}:MSD:AUTO**

DISP:DATA:PARAMeter{1|2}:MSD:AUTO

Set number of digits automatically. (AUTO SETUP PRI MSD, AUTO SETUP SEC MSD under **Display**); No Query)

Node	Description
PAR1	Primary Parameter
PAR2	Secondary Parameter

Query response is <numeric>.

Example                            OUTPUT @Hp4286;"DISP:DATA:PARA1:MSD:ALL"

**DISP:DATA:PARAMeter{1|2}:MSD:FIX {ON|OFF}**

DISP:DATA:PARAMeter{1|2}:MSD:FIX {ON|OFF}

Set the fixed decimal point function ON or OFF. (FIXED MSD on OFF under **Display**)

Node	Description
PAR1	Primary Parameter
PAR2	Secondary Parameter

Query response is {1|0}.

Example                            OUTPUT @Hp4286;"DISP:DATA:PARA1:MSD:FIX ON"  
                                       OUTPUT @Hp4286;"DISP:DATA:PARA1:MSD:FIX?"  
                                       ENTER @Hp4286;A

**DISPlay[:WINDow]:ALLocation {INSTrument|HIHB|BASic|BSTatus}**

DISPlay[:WINDow]:ALLocation {INSTrument|HIHB|BASic|BSTatus}

Selects the display allocation mode. (Option 1C2 only) (DISPLAY ALLOCATION under **Display**)

Parameter	Description
INSTrument	All instrument.
HIHB	Half Instrument and half HP Instrument BASIC.
BASic	All HP Instrument BASIC.
BSTatus	HP Instrument BASIC status.

Query response is {INST|HIHB|BAS|BST}.

Example                            OUTPUT @Hp4286;"DISP:ALL HIHB"  
                                       OUTPUT @Hp4286;"DISP:ALL?"  
                                       ENTER @Hp4286;A\$

**DISPlay[:WINDow]:BCOUnt:STATe {ON|OFF}**

DISPlay[:WINDow]:BCOUnt:STATe {ON|OFF}

Set BIN counter display ON of OFF. (B-CNT DISP ON OFF under **Test Setup**)

parameter	Description
ON	BIN counter display ON
OFF	BIN counter display OFF

Query response is {1|0}.

## DISPlay[:WINDow]:BCOU:STAT {ON|OFF}

Example                    OUTPUT @Hp4286;"DISP:BCOU:STAT ON"  
                          OUTPUT @Hp4286;"DISP:BCOU:STAT?"  
                          ENTER @Hp4286;A

## DISPlay[:WINDow]:TEXT{1-10}

## Subsystem

This subsystem is used to display the following information.

Node	Description
:TEXT1	Tabular listing of all the measured data points and their current values.
:TEXT2	Tabular listing of the key parameters for both channels.
:TEXT3	Tabular listing of the calibration kit definition.
:TEXT4	Tabular listing of the compensation standard definitions.
:TEXT5	List sweep table.
:TEXT6	List sweep table.
:TEXT7	Limit testing table. (Primary limit)
:TEXT8	Limit testing table. (Secondary limit)
:TEXT9	Bin table.
:TEXT10	Title.

## [[:DATA] <string>

DISPlay[:WINDow]:TEXT{1-10}[:DATA] <string>

Send the string to the following display area. (LABEL, TITLE, HEADLINE, FOOTNOTE under (System))

Node	Description
:TEXT10	The title area on the display of destination channel (TITLE)

Parameter	Description
<string>	up to 53 characters

Query response is <string>.

Example                    OUTPUT @Hp4286;"DISP:TEXT10 ""TITLE"""  
                          OUTPUT @Hp4286;"DISP:TEXT10?"  
                          ENTER @Hp4286;A\$

## :PAGE {UP|DOWN|<numeric>}

DISPlay[:WINDow]:TEXT{1-10}:PAGE {UP|DOWN|<numeric>}

Selects a page of a tabular list, which is defined as follows:



**DISPlay[:WINDow]:TEXT{1-10}**

Node	Description
:TEXT1	Displays a tabular listing of all the measured data points and their current values. (LIST VALUES under <a href="#">Copy</a> )
:TEXT2	Provides a tabular listing on the display of the key parameters for both channels. (OPERATING PARAMETERS under <a href="#">Copy</a> )
:TEXT3	Shows the tabular listing of the calibration kit definition. (CAL KIT DEFINITION under <a href="#">Copy</a> )
:TEXT4	Provides the tabular listing of the compensation standard definitions. (COMPEN KIT DEFINITION under <a href="#">Copy</a> )
:TEXT5	Displays the list sweep table by the “Start & Stop” format. (DISPLAY: LIST under <a href="#">Copy</a> LIST SWEEP TABLE)
:TEXT6	Displays the list sweep table by the “Center & Span” format. (DISPLAY: LIST under <a href="#">Copy</a> LIST SWEEP TABLE)
:TEXT7	Displays the limit testing table by the “Upper & Lower” format. (DISPLAY: LIST under <a href="#">Copy</a> LIMIT TEST TABLE)
:TEXT8	Displays the limit testing table by the “Mid & Delta” format. (DISPLAY: LIST under <a href="#">Copy</a> LIMIT TEST TABLE)
:TEXT9	Displays the BIN sort table. (DISPLAY: LIST under <a href="#">Copy</a> BIN SORT TABLE)

Parameter	Description
UP	Displays the next page of information in a tabular listing. (NEXT PAGE under <a href="#">Copy</a> )
DOWN	Displays the previous page of information in a tabular listing. (PREV PAGE under <a href="#">Copy</a> )
<numeric>	Defines the displayed page number

Query response is <numeric>.

Related Command      DISPlay[:WINDow]:TEXT{1-10}:STATe

Example                      OUTPUT @Hp4286;"DISP:TEXT3:PAGE 1"  
                                   OUTPUT @Hp4286;"DISP:TEXT3:STAT ON"  
                                   OUTPUT @Hp4286;"DISP:TEXT3:PAGE UP"

**:STATe {OFF|ON|0|1}**

DISPlay[:WINDow]:TEXT{1-10}:STATe {OFF|ON|0|1}

Displays a tabular list.

Node	Description
:TEXT{1-10}	See “DISPlay[:WINDow]:TEXT{1-10}:PAGE {UP DOWN <numeric>}” command.

Parameter	Description
OFF or 0	Do not display the item.
ON or 1	Displays the item.

Query response is {0|1}.

**DISPlay[:WINDow]:TEXT{1-10}**

**DISPlay[:WINDow]:TRACe**

**Subsystem**

---

This subsystem is used to control the display.

**:STATe {OFF|ON|0|1}**

DISPlay[:WINDow]:TRACe:STATe {OFF|ON|0|1}

Sets the screen display ON or OFF. (MEAS DISP ON off under (Display))

## FORMat Subsystem

**FORMat[:DATA] {ASCIi|REAL,32|REAL,64|PACKed,32}**

FORMat[:DATA] {ASCIi|REAL,32|REAL,64|PACKed,32}

Sets the format to transfer data via GPIB.

Parameter	Description
ASCIi	ASCII transfer format
REAL,32	IEEE 32-bit floating point format
REAL,64	IEEE 64-bit floating point format
PACKed,32	DOS format

Query Response is {ASC|REAL,32|REAL,64|PACK,32}.

At \*RST, ASCIi is selected.

**Related Command**      The FORMat[:DATA] command selects the format of the data sent by the following commands:

```
DATA[:DATA]
DATA[:DATA]?
DATA[:DATA]:VALue?
TRACe[:DATA]
TRACe[:DATA]:VALue?
```

**Example**

```
OUTPUT @Hp4286;"FORM REAL,32"
OUTPUT @Hp4286;"FORM?"
ENTER @Hp4286;A$
```

---

## HCOPY Subsystem

The HCOpy subsystem controls the hard copy functions (printing and plotting).

### HCOPY

HCOpy

See "HCOPY[:IMMediate]".

### HCOPY:ABORt

HCOpy:ABORt

Aborts a plot or print in progress. (**COPY ABORT** under **Copy**); No query)

Example                    OUTPUT @Hp4286;"HCOP:ABOR"

### HCOPY:DEFault

HCOpy:DEFault

Sets the plotting parameters to their default values. (**DEFAULT SETUP** under **Copy**); No query)

Example                    OUTPUT @Hp4286;"HCOP:DEF"

### HCOPY:DEvIce:LANGuage {HPGL|PCL}

HCOpy:DEvIce:LANGuage {HPGL|PCL}

Selects hard copy function (printing or plotting) and selects the output device (printer or plotter).

Parameter	Description
HPGL	Output to plotter
PCL	Output to printer

Query Response is {HPGL|PCL}.

Related Command        HCOpy[:IMMediate]

Example                    To plot out,  
                              OUTPUT @Hp4286;"HCOP:DEV:LANG HPGL"  
                              OUTPUT @Hp4286;"HCOP"

                              Query,  
                              OUTPUT @Hp4286;"HCOP:DEV:LANG?"  
                              ENTER @Hp4286;A\$

**HCOPY:DEVIce:SPEEd {1|2}**

HCOPY:DEVIce:SPEEd {1|2}

Sets the plotting speed to fast or slow. (PLOT SPEED [ ] under (Copy))

Parameter	Description
1	Slow plot speed (for plotting directly on transparencies).
2	Fast plot speed (for normal plotting).

Query Response is {1|2}.

**HCOPY[:IMMEDIATE]**

HCOPY[:IMMEDIATE]

Executes printing or plotting. (PRINT [ ] or PLOT under (Copy);No query)

Related Command HCOPY:DEVIce:LANGUage {HPGL|PCL}

**HCOPY:ITEM**

**Subsystem**

This subsystem is used to select the display hard copy object.

**:ANNotation:STATe {OFF|ON|0|1}**

HCOPY:ITEM:ANNotation:STATe {OFF|ON|0|1}

Select items to be plotted from the items displayed on the screen.(PLOT: ALL ,DATA&GRTCL ,DATA ONLY under (Copy))

Parameter	Description
OFF or 0	Data only
ON or 1	All items on the screen excepte for the softkey labels

Query response is {0|1}.

Example  
 OUTPUT @Hp4286;"HCOP:ITEM:ANN:STAT ON"  
 OUTPUT @Hp4286;"HCOP:ITEM:ANN:STAT?"  
 ENTER @Hp4286;A

## HCOPY:ITEM

**:MENU:STATe** {OFF|ON|0|1}

HCOPY:ITEM:MENU:STATe {OFF|ON|0|1}

Sets the LCR meter to plot the softkey label when plotting the display.

Parameter	Description
OFF or 0	Do not plot softkey label.
ON or 1	Plot the softkey label.

Query Response is {0|1}.

Example                    OUTPUT @Hp4286;"HCOP:ITEM:MENU:STAT ON"  
                          OUTPUT @Hp4286;"HCOP:ITEM:MENU:STAT?"  
                          ENTER @Hp4286;A

**:TDSTamp:STATe** {OFF|ON|0|1}

HCOPY:ITEM:TDSTamp:STATe {OFF|ON|0|1}

Turns printing or plotting time and date (the time stamp function) ON or OFF.

(COPY TIME ON off under **Copy**)

Parameter	Description
OFF or 0	Time stamp function OFF.
ON or 1	Time stamp function ON.

Query Response is {0|1}.

Example                    OUTPUT @Hp4286;"HCOP:ITEM:TDST:STAT ON"  
                          OUTPUT @Hp4286;"HCOP:ITEM:TDST:STAT?"  
                          ENTER @Hp4286;A

## HCOPY:PAGE

## Subsystem

This subsystem is used to set the plot size.

**:DIMensions:FULL**

HCOPY:PAGE:DIMensions:FULL

Selects the full-size plot setting. (xx FULL xx PAGE under **Copy** SELECT QUADRANT;

No query)

Example                    OUTPUT @Hp4286;"HCOP:PAGE:DIM:FULL"

**:DIMensions:QUADrant{1-4}**

HCOPY:PAGE:DIMensions:QUADrant{1-4}

Selects the quadrant plot setting. (SELECT QUADRANT under (Copy); No query)

Node	Description
:QUADrant1	Upper right (-x RIGHT -- UPPER)
:QUADrant2	Upper left (x- LEFT -- UPPER)
:QUADrant3	Lower left (-- LEFT x- LOWER)
:QUADrant4	Lower right (-- RIGHT -x LOWER)

Example                                    OUTPUT @Hp4286;"HCOP:PAGE:DIM:QUAD2"

**:SCALE {1|FULL|UPPer|LOWer}**

HCOPY:PAGE:SCALE {1|FULL|UPPer|LOWer}

Selects the scale size for plotting. (SCALE: FULL , UPPER GRATICULE , LOWER GRATICULE under (Copy))

Parameter	Description
1 or FULL	Normal full size.
UPPer	Upper graticule of the split display to the user-defined P1 and P2.
LOWer	Lower graticule of the split display to the user-defined P1 and P2.

Query Response is {FULL|UPP|LOW}.

Example                                    OUTPUT @Hp4286;"HCOP:PAGE:SCAL FULL"  
     OUTPUT @Hp4286;"HCOP:PAGE:SCAL?"  
     ENTER @Hp4286;A\$

---

## INITiate Subsystem

The INITiate subsystem is used to control the initiation of the trigger system. To trigger a measurement, the trigger sequence should be in the waiting-for-trigger state (that is, the trigger system should be initiated).

Related Command            **ABORt**  
                              **TRIGger:EVENT** TYPE {POINT|SWEep}  
                              **TRIGger:SLOPe** {POSitive|NEGative}  
                              **TRIGger:SOURce** {BUS|EXTernal|INTernal|MANual}

### INITiate:CONTinuous {OFF|ON|0|1}

INITiate:CONTinuous {OFF|ON|0|1}

Initiate the trigger system continuously.

Parameter	Description
OFF or 0	After measurement, the trigger sequence moves to the idle state.
ON or 1	After measurement, the trigger sequence moves to the wait-for-trigger state (continuously initiated).

Query Response is {0|1}.

Example                      To abort the measurement and hold the sweep immediately,

```
OUTPUT @Hp4286;"INIT:CONT OFF"  
OUTPUT @Hp4286;"ABOR"
```

Query statement,

```
OUTPUT @Hp4286;"INIT:CONT?"  
ENTER @Hp4286;A
```

### INITiate[:IMMediate]

INITiate[:IMMediate]

Initiate trigger system. (**SING** under **Trigger**);No query)

This command execution can cause an error if the trigger sequence is not in the idle state. To place it in the idle state, execute the **ABORt** command.

Example                      To trigger a measurement,

```
OUTPUT @Hp4286;"TRIG:SOUR INT"  
OUTPUT @Hp4286;"INIT:CONT OFF"  
OUTPUT @Hp4286;"ABOR"  
OUTPUT @Hp4286;"SENS:SWE:COUN 1"  
OUTPUT @Hp4286;"INIT"
```

After measurement, the LCR meter goes to trigger hold.

### INITiate[:IMMediate]:AGain:ALL

INITiate[:IMMediate]:AGain:ALL

Aborts the sweep in progress to exit to the idle state and initiates the trigger system again.

(**MEASURE RESTART** under **Trigger**); No query)



## MMEMory Subsystem

The MMEMory subsystem controls the save/recall function.

### MMEMory:CDIRectory [<string>]

MMEMory:CDIRectory [<string>]

Changes the current directory of a DOS format disk. (CHANGE DIRECTORY under **Save**); No query)

Parameter	Description
<string>	Directory path.

Example                                    OUTPUT @Hp4286;"MMEM:CDIR ""..""

### MMEMory:COPY {<string(s)>,<string(m\_s)>,<string(d)>,<string(m\_d)> }

MMEMory:COPY {<string(s)>,<string(m\_s)>,<string(d)>,<string(m\_d)> }

Copies files. (COPY FILE under **Save**); No query)

Parameter	Description
<string(s)>	Source file name.
<string(m_s)>	Source mass storage name ("DISK" or "MEMORY"). <sup>1</sup>
<string(d)>	Destination file name, up to 8 characters. In addition to the file name, the extension may be sent. Add appropriate extension if the destination mass storage is "MEMORY".
<string(m_d)>	Destination mass storage name. ("DISK" or "MEMORY")

<sup>1</sup> "DISK" for the built-in flexible disk drive; "MEMORY" for the RAM disk memory.

Example

OUTPUT @Hp4286;"MMEM:COPY ""DAT1.TXT"",""MEMORY"",""DAT1.TXT"",""DISK""

### Note



You cannot copy the file if the source mass storage format (DOS or LIF) is different from that of mass storage.

### MMEMory:CREate:DIRectory <string>

MMEMory:CREate:DIRectory <string>

Creates a new directory in a DOS format disk. (CREATE DIRECTORY under **Save**); No query)

Parameter	Description
<string>	Directory name, up to 8 characters. In addition to the directory name, the extension (up to 3 character) may be sent.

Example                                    OUTPUT @Hp4286;"MMEM:CRE:DIR ""DATA""

OUTPUT @Hp4286;"MMEM:CRE:DIR ""INDUCTOR.DIR""

## MMEMory:DElete <string(file\_name)>[,<string(m)>]

MMEMory:DElete <string(file\_name)>[,<string(m)>]

Removes the file. (PURGE FILE under (SAVE); No query)

Parameter	Description
<string(file_name)>	File name with extension.
<string(m)>	"DISK" for the flexible disk drive. "MEMORY" for the RAM disk memory.

Example                                    OUTPUT @Hp4286;"MMEM:DEL ""TEST\_S""

## MMEMory:DElete:CAL

MMEMory:DElete:CAL Delete the calibration and compensation data file CALREC\_C, which is automatically saved. (PURGE CALREC under (Save/Recall) ; No Query)

Example                                    OUTPUT @Hp4286;"MMEM:DEL:CLAL"

## MMEMory:FNAME:EXTension{1|2} <string>

MMEMory:FNAME:EXTension{1|2} <string>

Changes the extension of DOS format files.

Node	Description
:EXTension1	Changes the extension of HP-GL files for DOS format. The extension is automatically attached to the file name when an HP-GL file is saved. The factory setting is ".HPG". The modified extension is kept in SRAM. When SRAM was backup, the file is kept even if power is turned OFF. (GRAPHICS [ ] under (Save))
:EXTension2	Changes the extension of an ASCII data file for DOS format. The extension is automatically attached to the file name when an ASCII data file is saved. The factory setting is ".TXT". The modified extension is kept in SRAM. When SRAM was backup, the file is kept even if the power is turned OFF. (ASCII DATA [.TXT] under (Save))

Parameter	Description
<string>	Up to 3 characters.

Query response is <string>.

Example                                    OUTPUT @Hp4286;"MMEM:FNAME:EXT1 ""HPG""  
   OUTPUT @Hp4286;"MMEM:FNAME:EXT2 ""TXT""  
   OUTPUT @Hp4286;"MMEM:FNAME:EXT2?"  
   ENTER @Hp4286;A\$

**MMEMory:INITialize** <string>,{LIF|DOS}

MMEMory:INITialize &lt;string&gt;,{LIF|DOS}

Initializes the disk in the flexible disk drive or the RAM disk memory. (INITIALIZE under Save); No query)

Parameter	Description
<string>	"DISK" for the flexible disk drive. "MEMORY" for the RAM disk memory.
LIF	LIF format.
DOS	DOS format.

Example                                    OUTPUT @Hp4286;"MMEM:INIT ""DISK"" ,DOS"

**MMEMory:LOAD****Subsystem**

This subsystem is used to load data from the flexible disk drive or the RAM disk memory.

**:CAL**

MMEMory:LOAD:CAL

Load the calibration and compensation data file CALREC\_C, which was automatically saved. (RECALL CALREC under Save/Recall); No Query)

Example                                    OUTPUT @Hp4286;"MMEM:LOAD:CAL"

**:STATe** <string(file\_name)>[,<string(m)>]

MMEMory:LOAD:STATe &lt;string(file\_name)&gt;[,&lt;string(m)&gt;]

Loads the instrument states. (file name under Recall); No query)

Parameter	Description
<string(file_name)>	File name with extension
<string(m)>	"DISK" for the flexible disk drive. "MEMORY" for the RAM disk memory.

Example                                    OUTPUT @Hp4286;"MMEM:LOAD:STAT ""TEST\_S"""

**:TRACe SEL**,<string(file\_name)>[,<string(m)>]

MMEMory:LOAD:TRACe SEL,&lt;string(file\_name)&gt;[,&lt;string(m)&gt;]

Loads data. (file name under Recall); No query)

Parameter	Description
<string(file_name)>	File name with extension.
<string(m)>	"DISK" for the flexible disk drive. "MEMORY" for the RAM disk memory.

## MMEMory:LOAD

Example                                    OUTPUT @Hp4286;"MMEM:LOAD:TRAC SEL, ""TEST\_D""

## MMEMory:STORe

## Subsystem

This subsystem is used to store data to the flexible disk drive or the RAM disk memory.

### :CAL:AUTO {ON|OFF}

MMEMory:STORe:CAL:AUTO {ON|OFF}

Set the automatic save function ON or OFF. This function save the calibration and compensation data automatically after the calibration or compensation is executed. When this function is turned is ON, the file is automatically named CALC\_C and saved on the nonvolatile memory. This file is automatically loaded to the meter when the meter is turned on. If AUTOREC is already saved on the disk, the meter gets AUTOREC and ignores CALC\_C. (BKUP CALL on OFF under **Save/Recall**)

Parameter	Description
ON	Automatically save calibration and compensation data
OFF	Do not save calibration and compensation data automatically

Example                                    OUTPUT @Hp4286;MMEMory:STORe:CAL:AUTO ON

### :DINterchange:GRAPhics <string(file\_name)>[,<string(m)>]

MMEMory:STORe:DINterchange:GRAPhics <string(file\_name)>[,<string(m)>]

Saves the graphic image on the screen as an HP-GL file. (**GRAPHICS** under **Save**); No query)

Parameter	Description
<string(file_name)>	File name , up to 8 characters, without extension.
<string(m)>	"DISK" for the flexible disk drive. "MEMORY" for the RAM disk memory.

Example                                    OUTPUT @Hp4286;"MMEM:STOR:DINT:GRAP ""GRA1""

### :DINterchange:TRACe SEL,<string(file\_name)>[,<string(m)>]

MMEMory:STORe:DINterchange:TRACe SEL,<string(file\_name)>[,<string(m)>]

Saves data arrays as an ASCII file. (**DATA ONLY (ascii)** under **Save**); No query)

Parameter	Description
<string(file_name)>	File name, up to 8 characters.
<string(m)>	"DISK" for the flexible disk drive "MEMORY" for the RAM disk memory

Example                                    OUTPUT @Hp4286;"MMEM:STOR:DINT:TRAC SEL, ""DATA1""

**:ITEM:TRACe:CATalog?**

MMEMemory:STORe:ITEM:TRACe:CATalog?

Returns information on the current states of the data arrays that are saved. (Query only)

Query response is <string>, ..., <string>.

Where, <string>s are

- CCO : calibration/compensation coefficients
- DATA : data
- DTR : data trace
- RAW : raw data

**:ITEM:TRACe:DELeTe {CCO|DATA|DTR|RAW}**

MMEMemory:STORe:ITEM:TRACe:DELeTe {CCO|DATA|DTR|RAW}

Selects the item that is not saved. (DEFINE SAVE DATA under **Save**); No query)

This command deletes the item which was set to be saved by the MMEMemory:STORe:ITEM:TRACe:SELeCt command. The deleted items will not be saved.

Parameter	Description
CCO	calibration/compensation coefficient (CAL on OFF)
DATA	data (DATA on OFF)
DTR	data trace (DATA TRACE on OFF)
RAW	raw data (RAW on OFF)

Related Command MMEMemory:STORe:ITEM:TRACe:SELeCt  
MMEMemory:STORe:TRACe  
MMEMemory:STORe:DINTErchange:TRACe

Example OUTPUT @Hp4286;"MME:STOR:ITEM:TRAC:DEL CCO"

**:ITEM:TRACe:SELeCt {CCO|DATA|DTR|RAW}**

MMEMemory:STORe:ITEM:TRACe:SELeCt {CCO|DATA|DTR|RAW}

Selects the items to be saved. (DEFINE SAVE DATA under **Save**); No query)

Parameter	Description
CCO	calibration/compensation coefficient (CAL on OFF)
DATA	data (DATA ON off)
DTR	data trace (DATA TRACE ON off)
RAW	raw data (RAW ON off)

Related Command MMEMemory:STORe:ITEM:TRACe:DELeTe  
MMEMemory:STORe:TRACe  
MMEMemory:STORe:DINTErchange:TRACe

Example OUTPUT @Hp4286;"MME:STOR:ITEM:TRAC:SEL CCO"



## PROGram Subsystem

The PROGram subsystem controls the HP Instrument BASIC furnished in the LCR meter from an external controller. The PROGram subsystem is used from an external controller.

The PROGram subsystem is organized by,

- PROGram:CATalog? command
- PROGram[:SElected] subsystem
- PROGram:EXPLicit subsystem

The PROGram[:SElected] subsystem and PROGram:EXPLicit subsystem are structured the same and have same function. This manual provides complete information on the PROGram[:SElected]. For PROGram:EXPLicit subsystem, it refers to the corresponding description in the PROGram[:SElected] subsystem.

### PROGram:CATalog?

PROGram:CATalog?

Returns the defined program name (defined by PROGram[:SElected]:NAME command). This command performs no practical function for the LCR meter. (Query only)

Query response is <string>.

## PROGram[:SElected] Subsystem

This subsystem is used to control the HP Instrument BASIC from an external controller.

### :DEFine <block>

PROGram[:SElected]:DEFine <block>

Creates and downloads programs from an external controller to HP Instrument BASIC. The PROGram[:SElected]:DEFine? query uploads HP Instrument BASIC programs to an external controller. This command can be used from an external controller only.

Parameter	Description
<block>	program

The <block> must be arbitrary block program data containing the lines of program code. The first line of <block> must be a header that shows the program size. There are two formats for the header as follows:

- #0 Allows the OUTPUT statement to send program lines 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 bytes (N digits).

Each line of the program must be separated by <New Line>. When the size of the <block> exceeds the amount of available memory in the LCR meter, the program lines are saved up to the point of memory overflow.

In response to the DEFine query, the <block> is uploaded as the definite length arbitrary block response data. The program size is returned in the first line as the header, then the program lines are returned. The program must be either in the paused or stopped state for the program to be uploaded.

## PROGram[:SElected]

Example

To create a program in the HP Instrument BASIC editor,

```
OUTPUT @Hp4286;"PROG:DEF #0"  
OUTPUT @Hp4286;"10 PRINT ""HELLO!""  
OUTPUT @Hp4286;"20 END"  
OUTPUT @Hp4286;" ",END
```

To upload the program to the external controller,

```
DIM A$ [100000]  
OUTPUT @Hp4286;"PROG:DEF?"  
ENTER @Hp4286 USING "%,2A";HEAD$  
B=VAL(HEAD$[2])  
FOR I=1 TO B  
    ENTER @Hp4286 USING "%,A";HEAD$  
NEXT I  
ENTER @Hp4286 USING "-K";A$
```

## :DELeTe[:SElected]

PROGram[:SElected]:DELeTe[:SElected]

Deletes the HP Instrument BASIC program in the LCR meter. This command can be used from an external controller only. (No query)

## :DELeTe:ALL

PROGram[:SElected]:DELeTe:ALL

Deletes the HP Instrument BASIC program in the LCR meter. This command can be used from an external controller only. (No query)

This command works the same as the PROGram[:SElected]:DELeTe[:SElected] command.

## :EXECute <string>

PROGram[:SElected]:EXECute <string>

Executes the program command. The program must be either paused or stopped before the EXECute command is executed. This command can be used from an external controller only. (No query)

Parameter	Description
<string>	Legal program command



Example<sup>1</sup>

To execute the program command, STEP

```
OUTPUT @Hp4286;"PROG:EXEC 'STEP'"
OUTPUT @Hp4286;"PROG:EXEC ""STEP"""
```

To execute the program command, MSI ":MEMORY"

```
OUTPUT @Hp4286;"PROG:EXEC 'MSI ':MEMORY'''"
OUTPUT @Hp4286;"PROG:EXEC ""MSI """:MEMORY"""""""
```

To execute the program command, GET "";File\$;""

```
OUTPUT @Hp4286;"PROG:EXEC 'GET ''";File$;'''"
OUTPUT @Hp4286;"PROG:EXEC ""GET """";";File$;"""""""""
```

1 Each example contains two cases. One uses the single quote, the other uses the double quote.

**:MAlLocate** {<numeric>|DEfAult}

PROG:SElected]:MAlLocate {<numeric>|DEfAult}

Performs no practical function for the LCR meter.

**:NAME** <string>

PROG:SElected]:NAME <string>

Defines the program name. This command performs no practical function for the LCR meter.

Parameter	Description
<string>	Program name (Default is "PROG".)

Query response is <string>.

**:NuMBer** <var>,<numeric1>,<numeric2>,...,<numeric n>

PROG:SElected]:NuMBer <var>,<numeric1>,<numeric2>,...,<numeric n>

Sets or queries the contents of numeric program variables and arrays in the HP Instrument BASIC of the LCR meter. This command performs a practical function from the external controller only.

Parameter	Description
<var>	Name of an existing variable in the program (either character data or string data)
<numeric1>,...,<numeric n>	Variable value (n is the number of the array.)

Query response is <numeric1>,<numeric2>,...,<numeric n>.

Example

```
OUTPUT @Hp4286;"PROG:NUMB A,1"
OUTPUT @Hp4286;"PROG:NUMB? A"
ENTER @Hp4286;B
```

## **:STATe** {**RUN**|**PAUSE**|**STOP**|**CONTInue**}

PROG<sub>ram</sub>[SE<sub>lected</sub>].STATe {**RUN**|**PAUSE**|**STOP**|**CONTInue**}

Sets or queries the state of the program in the LCR meter. The table below defines the affect of setting the state to the specified state from each of the possible current states. This command performs a practical function from the external controller only.

Parameter	Current State		
	RUN	PAUSE	STOP
RUN	error (-221)	RUN	RUN
CONT	error (-221)	RUN	error (-221)
PAUSE	PAUSE	PAUSE	STOP
STOP	STOP	STOP	STOP

Query response is {**RUN**|**PAUS**|**STOP**|**CONT**}.

Example                    OUTPUT @Hp4286;"PROG:STAT STOP"  
                              OUTPUT @Hp4286;"PROG:STAT?"  
                              ENTER @Hp4286;A\$

## **:STRing** <var>,<string1>,<string2>,...,<string n>

PROG<sub>ram</sub>[SE<sub>lected</sub>].STRing <var>,<string1>,<string2>,...,<string n>

Sets or queries the contents of the string program variables and arrays in the HP Instrument BASIC of the LCR meter. If a string value is too long, it is truncated when stored in the program's variable. This command performs a practical function from the external controller only.

Parameter	Description
<var>	Name of an existing variable in the program (either character data or string data).
<string1>,...,<string n>	Variable value (n is number of the element in the array).

Query response is <string1>,<string2>,...,<string n>.

Example                    OUTPUT @Hp4286;"PROG:STR A,""HELLO"""  
                              OUTPUT @Hp4286;"PROG:STR? A"  
                              ENTER @Hp4286;B\$

## **:WAIT**

PROG<sub>ram</sub>[SE<sub>lected</sub>].WAIT

Causes no further commands or queries to be executed until the program in the LCR meter exits from the RUN state (either stopped or paused). This command performs a practical function from the external controller only.

Query Response is 1.  
(1 is returned when the program is either stopped or paused.)

Example To wait until the program in the LCR meter completes,

```
OUTPUT @Hp4286;"PROG:WAIT"
PRINT "The program complete"

OUTPUT @Hp4286;"PROG:WAIT?"
ENTER @Hp4286;A
```

**PROG:EXPLICIT**

**Subsystem**

The following commands under the :EXPLICIT node perform the specified functions in the same manner as the corresponding commands under the :SELECTED node. The :EXPLICIT commands are included in the LCR meter’s GPIB commands to maintain compatibility with other SCPI instruments. Therefore, you can use either the :EXPLICIT or the :SELECTED commands for the LCR meter. However, you should select one set and use it consistently to avoid confusion.

**:DEFine “PROG”, <block>**  
 PROG:EXPLICIT:DEFine "PROG", <block>

See “PROG[:SELECTED]:DEFine <block>”

**:DELeTe “PROG”**  
 PROG:EXPLICIT:DELeTe "PROG"

See “PROG[:SELECTED]:DELeTe[:SELECTED]”

**:EXECute “PROG”, <string>**  
 PROG:EXPLICIT:EXECute "PROG", <string>

See “PROG[:SELECTED]:EXECute <string>”

**:MALLocate “PROG”, {<numeric>|DEFault}**  
 PROG:EXPLICIT:MALLocate "PROG", {<numeric>|DEFault}

See “PROG[:SELECTED]:MALLocate {<numeric>|DEFault}”

**:NAME “PROG”, <string>**  
 PROG:EXPLICIT:NAME "PROG", <string>

See “PROG[:SELECTED]:NAME <string>”

**:NUMBER “PROG”, <var>, <numeric1>, <numeric2>, ... , <numeric n>**  
PROGRAM:EXPLICIT:NUMBER "PROG", <var>, <numeric1>, <numeric2>, ..., <numeric n>

See “PROGRAM[:SELECTED]:NUMBER <var>, <numeric1>, <numeric2>, ... , <numeric n>”

**:STATE “PROG”, {RUN|PAUSE|STOP|CONTINUE}**  
PROGRAM:EXPLICIT:STATE "PROG", {RUN|PAUSE|STOP|CONTINUE}

See “PROGRAM[:SELECTED]:STATE {RUN|PAUSE|STOP|CONTINUE}”

**:STRING “PROG”, <var>, <string1>, <string2>, ... , <string n>**  
PROGRAM:EXPLICIT:STRING "PROG", <var>, <string1>, <string 2>, ..., <string n>

See “PROGRAM[:SELECTED]:STRING <var>, <string1>, <string2>, ... , <string n>”

**:WAIT “PROG”**  
PROGRAM:EXPLICIT:WAIT "PROG"

See “PROGRAM[:SELECTED]:WAIT”

## SENSe Subsystem

### SENSe:CORRection1

### Subsystem

This subsystem is used to control the calibration function.

**:CKIT** {APC35|APC7|UDEFined}

SENSe:CORRection1:CKIT {APC7|UDEFined}

Selects the calibration kit. (CAL KIT:7mm, or USER KIT under (Cal))

Parameter	Description
APC7	Select furnished calibration kit.
UDEFined	Select user-modified calibration kit.

Query response is {APC7|UDEF}.

Example                                    OUTPUT @Hp4286;"SENS:CORR:CKIT APC7"  
     OUTPUT @Hp4286;"SENS:CORR:CKIT?"  
     ENTER @Hp4286;A\$

**:CKIT:LABel** <string>

SENSe:CORRection1:CKIT:LABel <string>

Defines a label for the user-modified calibration kit. (LABEL KIT under (Cal))

Parameter	Description
<string>	Up to eight characters.

Example                                    OUTPUT @Hp4286;"SENS:CORR:CKIT:LAB ""MYKIT"""

**:CKIT:SAVE**

SENSe:CORRection1:CKIT:SAVE

Stores the user-modified calibration kit into the memory. (SAVE USER KIT under (Cal);  
 No query)

**:CKIT:STANdard1:C** <numeric>

SENSe:CORRection1:CKIT:STANdard1:C <numeric>

Enters the capacitance value of the OPEN standard that is used in the user-modified calibration kit. (CAP.(C) under (Cal) CAL KIT [ ])

## SENSe:CORRection1

Parameter	Range	Unit
<numeric>	-1×10 <sup>-9</sup> to 1×10 <sup>-9</sup>	F

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN1:C 53"

### :CKIT:STANdard1:G <numeric>

SENSe:CORRection1:CKIT:STANdard1:G <numeric>

Enters the conductance value of the OPEN standard that is used in the user-modified calibration kit. (OPEN:CONDUCT(G) under (Cal) CAL KIT [ ])

Parameter	Range	Unit
<numeric>	-1×10 <sup>6</sup> to 1×10 <sup>6</sup>	S

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN1:G 0"

### :CKIT:STANdard2:L <numeric>

SENSe:CORRection1:CKIT:STANdard2:L <numeric>

Enters the inductance value of the SHORT standard that is used in the user-modified calibration kit. (INDUCT (L) under (Cal) CAL KIT [ ])

Parameter	Range	Unit
<numeric>	-1×10 <sup>6</sup> to 1×10 <sup>6</sup>	H

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN2:L 0"

### :CKIT:STANdard2:R <numeric>

SENSe:CORRection1:CKIT:STANdard2:R <numeric>

Enters the resistance value of the SHORT standard that is used in the user-modified calibration kit. (SHORT:RESIST.(R) under (Cal) CAL KIT [ ])

Parameter	Range	Unit
<numeric>	-1×10 <sup>6</sup> to 1×10 <sup>6</sup>	Ω

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN2:R 0"

**:CKIT:STANdard3:F <numeric>**

SENSe:CORRection1:CKIT:STANdard3:F &lt;numeric&gt;

Enters the frequency value that is used to define Q value of the LOAD standard that is used in the user-modified calibration kit. (FREQ.(F) under (Cal))

Parameter	Range	Unit
<numeric>	$1 \times 10^6$ to $1 \times 10^{12}$	Hz

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"SENS:CORR1:LOAD:TYPE LQF"  
                               OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:F 1E6"

**:CKIT:STANdard3:L <numeric>**

SENSe:CORRection1:CKIT:STANdard3:L &lt;numeric&gt;

Enters the inductance value of the LOAD standard that is used in the user-modified calibration kit. (LOAD:INDUCT.(L) under (Cal))

Parameter	Range	Unit
<numeric>	$-1 \times 10^6$ to $1 \times 10^6$	H

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"SENS:CORR1:LOAD:TYPE RL"  
                               OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:L 1E-4"

**:CKIT:STANdard3:Q <numeric>**

SENSe:CORRection1:CKIT:STANdard3:Q &lt;numeric&gt;

Enters the Q value of the LOAD standard that is used in the user-modified calibration kit. (QFACT.(Q) under (Cal))

Parameter	Range	Unit
<numeric>	$1 \times 10^{-12}$ to $1 \times 10^6$	

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"SENS:CORR1:LOAD:TYPE LQF"  
                               OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:Q 10"

**:CKIT:STANdard3:R <numeric>**

SENSe:CORRection1:CKIT:STANdard3:R &lt;numeric&gt;

Enters the resistance value of the LOAD standard that is used in the user-modified calibration kit. (LOAD:RESIST.(R) under (Cal) CAL KIT [ ] )

Parameter	Range	Unit
<numeric>	$-1 \times 10^6$ to $1 \times 10^6$	$\Omega$

Query response is <numeric>.

## SENSe:CORRection1

Example                                    OUTPUT @Hp4286;"SENS:CORR1:LOAD:TYPE RL"  
    OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:R 50"

**:COLlect[:ACQuire] {STANdard1|STANdard2|STANdard3|STANdard4}**  
SENSe:CORRection1:COLlect[:ACQuire] {STANdard1|STANdard2|STANdard3|STANdard4}

Selects and acquires the calibration. (**OPEN**, **SHORT**, **LOAD**, and **LOW-LOSS CAPACITOR** under **Cal**); No query)

The order in which you acquire the STANdard1, STANdard2, STANdard3, and STANdard4 is changeable. You can suspend a calibration sequence and do a different operation, and then resume the calibration sequence.

You must program waiting time for the calibration standard measurement to complete. An example of acquiring the OPEN standard measurement is shown below.

Parameter	Description
STANdard1	OPEN
STANdard2	SHORT
STANdard3	LOAD
STANdard4	Low-loss capacitor

Example                                    ON INTR 8 GOTO Meas\_Complete  
    OUTPUT @Hp4286;"STAT:INST:ENAB 256;\*SRE 4"  
    OUTPUT @Hp4286;"\*CLS;\*OPC?"  
    ENTER @Hp4286;0pc  
    INPUT "Connect OPEN, then press [Return]",Dum\$  
    ENABLE INTR 8;2  
           OUTPUT @Hp4286;"SENS:CORR1:COLL STAN1"  
           Waiting: GOTO Waiting  
    Meas\_complete: !

**:COLlect:FPOints {FIXed|USER}**  
SENSe:CORRection1:COLlect:FPOints {FIXed|USER}

Selects the frequency points where the calibration data is collected. (**CAL POINTS [ ]** under **Cal**)

Parameter	Description
FIXed	172 points (LCR meter specific). <sup>1</sup>
USER	All measurement points of current setting.

Query response is {FIX|USER}.

<sup>1</sup> For the frequency values of the points, see the *Function Reference*.



**:COLlect:SAVE**

SENSe:CORRection1:COLlect:SAVE

Calculates the error-correction coefficients from the calibration data and stores the coefficients. (DONE: CAL under Cal); No query)

**:EDELay:STATe {OFF|ON|0|1}**

SENSe:CORRection1:EDELay:STATe {OFF|ON|0|1}

Sets the port extension ON or OFF. (EXTENSION ON off under Cal)

Parameter	Description
OFF or 0	Port extension OFF.
ON or 1	Port extension ON.

Query response is {0|1}.

Example                    OUTPUT @Hp4286;"SENS:CORR:EDEL:STAT ON"  
                               OUTPUT @Hp4286;"SENS:CORR:EDEL:STAT?"  
                               ENTER @Hp4286;A

**:EDELay[:TIME] <numeric>**

SENSe:CORRection1:EDELay[:TIME] &lt;numeric&gt;

Sets the port extension value. (EXTENSION VALUE under Cal)

Parameter	Range	Unit
<numeric>	-10 to 10	s

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"SENS:CORR1:EDEL 0"  
                               OUTPUT @Hp4286;"SENS:CORR1:EDEL?"  
                               ENTER @Hp4286;A

**[:STATe]?**

SENSe:CORRection1[:STATe]?

Queries to determine if the correction state is ON or OFF (Query only)

Query response is {0|1}.

Where,

- 0           : Correction OFF (Calibration was not performed).
- 1           : Correction ON (Calibration was performed).

**:LOAD:TYPE {RL|LQF}**

SENSe.CORRection1:LOAD:TYPE {RL|LQF}

Selects the combination of the parameters to define the LOAD standard.

<b>Parameter</b>	<b>Description</b>
RL	User Resistance value(R), Inductance value (L)
LQF	Use Inductance value (L), Q value (Q), and Frequency value that the L and Q values are defined.

**SENSE:CORRection2****Subsystem**

This subsystem controls test fixture compensation.

**:CKIT[1]:LABel <string>**

SENSE:CORRection2:CKIT[1]:LABel <string>

Defines the label for the user-modified compensation kit. (LABEL KIT under COMPEN KIT, (Cal))

Parameter	Description
<string>	Up to eight characters.

Example                                    OUTPUT @Hp4286;"SENS:CORR:CKIT:LAB ""MYKIT""

**:CKIT[1]:SAVE**

SENSE:CORRection2:CKIT[1]:SAVE

Stores the user-modified compensation kit into memory. (SAVE COMPEN KIT under (Cal); No query)

**:CKIT[1]:STANdard1:C <numeric>**

SENSE:CORRection2:CKIT[1]:STANdard1:C <numeric>

Enters the capacitance value of the OPEN standard that is used in OPEN compensation. (CAP.(C) under (Cal))

Parameter	Range	Unit
<numeric>	$-1 \times 10^{-9}$ to $1 \times 10^{-9}$	F

Query response is <numeric>.

Example                                    OUTPUT @Hp4286;"SENS:CORR2:CKIT:STAN1:C 53"

**:CKIT[1]:STANdard1:G <numeric>**

SENSE:CORRection2:CKIT[1]:STANdard1:G <numeric>

Enters the conductance value of the OPEN standard that is used in OPEN compensation. (OPEN: CONDUCT(G) under (Cal))

Parameter	Range	Unit
<numeric>	$-1 \times 10^6$ to $1 \times 10^6$	S

Query response is <numeric>.

Example                                    OUTPUT @Hp4286;"SENS:CORR2:CKIT:STAN1:G 0"



**:CKIT:STANdard3:F <numeric>**

SENSe:CORRection2:CKIT:STANdard3:F &lt;numeric&gt;

Enters the frequency value that is used to define the Q value of the LOAD standard that is used in LOAD compensation. (FREQ.(F) under (Cal))

Parameter	Range	Unit
<numeric>	$1 \times 10^6$ to $1 \times 10^{12}$	Hz

Query response is <numeric>.

Example                                    OUTPUT @Hp4286;"SENS:CORR2:LOAD:TYPE LQF"  
     OUTPUT @Hp4286;"SENS:CORR2:CKIT:STAN3:F 1E5"

**:CKIT:STANdard3:L <numeric>**

SENSe:CORRection2:CKIT:STANdard3:L &lt;numeric&gt;

Enters the L value of the LOAD standard that is used in LOAD compensation. (LOAD INDUCT.(L) under (Cal))

Parameter	Range	Unit
<numeric>	$-1 \times 10^6$ to $1 \times 10^6$	H

Query response is <numeric>.

Example                                    OUTPUT @Hp4286;"SENS:CORR2:CKIT:STAN3:L 1E-4"

**:CKIT:STANdard3:Q <numeric>**

SENSe:CORRection2:CKIT:STANdard3:Q &lt;numeric&gt;

Enters the Q value of the LOAD standard that is used in LOAD compensation. (QFACT.(Q) under (Cal))

Parameter	Range	Unit
<numeric>	$1 \times 10^{-12}$ to $1 \times 10^6$	

Query response is <numeric>.

Example                                    OUTPUT @Hp4286;"SENS:CORR2:LOAD:TYPE LQF"  
     OUTPUT @Hp4286;"SENS:CORR2:CKIT:STAN3:Q 10"

**:CKIT[1]:STANdard3:R <numeric>**

SENSe:CORRection2:CKIT[1]:STANdard3:R &lt;numeric&gt;

Enters the resistance value of the LOAD standard that is used in LOAD compensation. (LOAD: RESIST.(R) under (Cal))

Parameter	Range	Unit
<numeric>	$-1 \times 10^6$ to $1 \times 10^6$	$\Omega$

Query response is <numeric>.

## SENSe:CORRection2

Example                    OUTPUT @Hp4286;"SENS:CORR2:LOAD:TYPE RL"  
                              OUTPUT @Hp4286;"SENS:CORR2:CKIT:STAN3:R 50"

### :CKIT[1]:STANdard3[:SElect] {LIST|LPARAmeter}

SENSe:CORRection2:CKIT[1]:STANdard3[:SElect] {LIST|LPARAmeter}

Selects the fixture compensation load standard.

Parameter	Description
LIST	Fixture compensation load standard array
LPARAmeter	The value set by SENSe:CORRection2:CKIT[1]:STANdard3:L, and SENSe:CORRection2:CKIT[1]:STANdard3:R

Query response is {LIST|LPAR}.

### :COLLect[:ACQuire] STANdard{1-3}

SENSe:CORRection2:COLLect[:ACQuire] STANdard{1-3}

Measure the standard for the fixture compensation. (COMPEN : OPEN, SHORT, LOAD under Cal); No query)

Parameter	Description
STANdard1	OPEN compensation for impedance measurement. <sup>1</sup>
STANdard2	SHORT compensation for impedance. <sup>1</sup> measurement
STANdard3	LOAD compensation for impedance. <sup>1</sup> measurement

<sup>1</sup> Also admittance and reflection coefficient measurement

Example                    OUTPUT @Hp4286;"SENS:CORR2:COLL STAN1"

### :COLLect:FPOints {FIXed|USER}

SENSe:CORRection2:COLLect:FPOints {FIXed|USER}

Select the frequency points where the correction data is collected. (COMP POINT [ ] under Cal)

Parameter	Description
FIXed	172 points (LCR meter specific). <sup>1</sup>
USER	All measurement points of current setting.

Query response is {FIX|USER}.

<sup>1</sup> For the frequency values of the points, see the *Function Reference*.

Example                    OUTPUT @Hp4286;"SENS:CORR2:COLL:FPO FIX"

**:COLlect:SAVE**

SENSe:CORRection2:COLlect:SAVE

Calculates the fixture compensation coefficients and stores the coefficients.

(DONE: COMPEN under **Cal**); No query)

**:OPEN[:STATe] {OFF|ON|0|1}**

SENSe:CORRection2:OPEN[:STATe] {OFF|ON|0|1}

Sets the OPEN fixture compensation ON or OFF. (OPEN ON off under **Cal**)

Parameter	Description
OFF or 0	OPEN fixture compensation OFF.
ON or 1	OPEN fixture compensation ON.

Query response is {0|1}.

Example                    OUTPUT @Hp4286;"SENS:CORR2:OPEN ON"  
                               OUTPUT @Hp4286;"SENS:CORR2:OPEN?"  
                               ENTER @Hp4286;A

**:SHORT[:STATe] {OFF|ON|0|1}**

SENSe:CORRection2:SHORT[:STATe] {OFF|ON|0|1}

Sets the SHORT fixture compensation ON or OFF. (SHORT ON off under **Cal**)

Parameter	Description
OFF or 0	SHORT fixture compensation OFF.
ON or 1	SHORT fixture compensation ON.

Query response is {0|1}.

Example                    OUTPUT @Hp4286;"SENS:CORR2:SHOR ON"  
                               OUTPUT @Hp4286;"SENS:CORR2:SHOR?"  
                               ENTER @Hp4286;A

**:LOAD[:STATe] {OFF|ON|0|1}**

SENSe:CORRection2:LOAD[:STATe] {OFF|ON|0|1}

Sets the LOAD fixture compensation ON or OFF. (LOAD ON off under **Cal**)

Parameter	Description
OFF or 0	LOAD fixture compensation OFF.
ON or 1	LOAD fixture compensation ON.

Query response is {0|1}.

Example                    OUTPUT @Hp4286;"SENS:CORR2:LOAD ON"  
                               OUTPUT @Hp4286;"SENS:CORR2:LOAD?"  
                               ENTER @Hp4286;A

SENSe:LIST

Subsystem

This subsystem is used to control the list sweep function.

**:CLEAr**

SENSe:LIST:CLEAr

Clears the entire list. (CLEAR LIST under Sweep Setup); No query)

**:MODIfy:SEGMENT{1-10}:FREQUency <numeric>**

SENSe:LIST:MODIfy:SEGMENT{1-3}:FREQUency <numeric>

Sets frequency values of each segment. (SEGMENT 1 FREQ , SEGMENT 2 FREQ , SEGMENT 3 FREQ under Sweep Setup); No Query)

**:MODIfy:SEGMENT{1-10}:AVERAge:COUNt <numeric>**

SENSe:LIST:MODIfy:SEGMENT{1-3}:AVERAge:COUNt <numeric>

Sets averaging number of each segments. (SEGMENT 1 AVG , SEGMENT 2 AVG , SEGMENT 3 AVG under Sweep Setup); No Query)

**:SAVE**

SENSe:LIST:SAVE

Completes editing the frequency sweep list. (LIST DONE under Sweep Setup); No query)

**:SEGMENT <numeric>**

SENSe:LIST:SEGMENT <numeric>

Selects the segment to edit.

Parameter	Description
<numeric>	Segment number, 1 to 15

Query response is <numeric>.



**:SEGment:ADD**

SENSe:LIST:SEGment:ADD

Adds a new segment to a list sweep table. (**ADD** under **Sweep Setup**); No query)

**:SEGment:AVERage:COUNT <numeric>**

SENSe:LIST:SEGment:AVERage:COUNT &lt;numeric&gt;

Sets the point averaging factor for the list sweep table. (**AVERAGING ON POINT** under **Sweep Setup**)

Parameter	Range	Unit
<numeric>	0 to 999	—

Query response is <numeric>.

Related Command      SENSe:LIST:SEGment:ADD

Example

```

OUTPUT @Hp4286;"SENS:LIST:SEGM:ADD"
OUTPUT @Hp4286;"SENS:LIST:SEGM:AVER:COUN 4"

OUTPUT @Hp4286;"SENS:LIST:SEGM:AVER:COUN?"
ENTER @Hp4286;A

```

**:SEGment:DELeTe**

SENSe:LIST:SEGment:DELeTe

Deletes a segment from a list sweep table. (**DELETE** under **Sweep Setup**); No query)

**:SEGment:EDIT**

SENSe:LIST:SEGment:EDIT

Edits the segment. (**EDIT** under **Sweep Setup**); No query)

**:SEGment:FREQuency <numeric>**

SENSe:LIST:SEGment:FREQuency &lt;numeric&gt;

Sets the frequency value of the segment in the list sweep table. (**CENTER** under **Sweep Setup**)

Parameter	Range	Unit
<numeric>	1000000 (= 1 M) to $1.0 \times 10^9$ (= 1 G)	Hz

Query response is <numeric>.

## **:SEGment:QUIT**

SENSe:LIST:SEGment:QUIT

Quits editing a segment of the list sweep table. (**SEGMENT QUIT** under **Sweep Setup**); No query)

## **:SEGment:SAVE**

SENSe:LIST:SEGment:SAVE

Completes modifying a segment in a list sweep table. (**SEGMENT DONE** under **Sweep Setup**); No query)

## **SENSe:RDC [:STATe] {OFF|ON|0|1}**

SENSe:RDC [:STATe] {OFF|ON|0|1}

Sets the Rdc measurement for the contact check function ON or OFF

Parameter	Description
OFF or 0	Rdc measurement OFF.
ON or 1	Rdc measurement ON.

Query response is {0|1}.

Example

```
OUTPUT @Hp4286;"SENS:RDC ON"  
OUTPUT @Hp4286;"SENS:RDC?"  
ENTER @Hp4286;A
```

## **SENSe:SWEep:DWELl{1|2} <numeric>**

SENSe:SWEep:DWELl{1|2} <numeric>

Defines the delay times set by **SENSe:SWEep:DWELl{1|2}:AUTO**. (**POINT DELAY TIME**, **SWEEP DELAY TIME** under **Sweep Setup**)

Node	Description
:DWEL11	Point delay time ( <b>POINT DELAY TIME</b> )
:DWEL12	Sweep delay time ( <b>SWEEP DELAY TIME</b> )

Parameter	Range	Unit
<numeric>	0 to 3600 (resolution : 12.5 $\mu$ )	s

Query response is <numeric>.

Related Command **SENSe:SWEep:DWELl{1|2}:AUTO**

**SENSe:SWEep:DWELI{1|2}**

Example

```
OUTPUT @Hp4286;"SENS:SWE:DWEL 10"  
OUTPUT @Hp4286;"SENS:SWE:DWEL:AUTO ON"  
OUTPUT @Hp4286;"SENS:SWE:DWEL?"  
ENTER @Hp4286;A
```

---

## SOURce Subsystem

The SOURce subsystem controls the signal source of the LCR meter.

**SOURce:{CURRent|POWer}[:LEVel][:IMMediate][:AMPLitude] <numeric>**

SOURce:{CURRent|POWer}[:LEVel][:IMMediate][:AMPLitude] <numeric>

Sets the OSC level for a frequency sweep. (**SEGMENT:FREQ** under **Sweep Setup**)

Node	Description
:CURRent	Current
:POWer	Power

Parameter	Range	Unit
<numeric>	$200 \times 10^{-6}$ to $20 \times 10^{-3}$	A (:CURRent)
	-33 to 7	dBm (:POWer)

Query response is <numeric>.

Related Command      SOURce:VOLTage[:LEVel] [:IMMediate] [:AMPLitude]

---

## SOURce:VOLTage

## Subsystem

This subsystem is used to control the ac signal source.

**[:LEVel][:IMMediate][:AMPLitude] <numeric>**

SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude] <numeric>

Sets the OSC level for frequency/DC-I/DC-V sweep. (**OSC LEVEL** under **Source**)

Parameter	Range	Unit
<numeric>	$10 \times 10^{-3}$ to 1	V

Query response is <numeric>.

## STATus Subsystem

The LCR meter has a status-reporting structure that is defined by IEEE 488.2 and SCPI. The STATus subsystem controls the SCPI-defined status-reporting structures, which are STATus:INSTRument, STATus:OPERation, and STATus:QUESTionable registers. The common commands (listed under “Related Command”) control the IEEE 488.2-defined status reporting structure.

Related Command            \*CLS  
                                  \*ESE  
                                  \*ESR  
                                  \*OPC  
                                  \*SRE  
                                  \*STB?

## STATus:INSTRument Subsystem

This subsystem is used to control the Instrument Status Registers (Instrument Event Status Register and Instrument Event Status Enable Register).

### **:ENABLE** <numeric>

STATus:INSTRument:ENABle <numeric>

Sets the contents of the Instrument Event Status Enable Register.

Parameter	Range
<numeric>	Decimal expression of the contents of the register; 0 to 32767 ( $=2^{15}-1$ )

Query response is <numeric>.

Example                            OUTPUT @Hp4286;"STAT:INST:ENAB 8"  
    OUTPUT @Hp4286;"STAT:INST:ENAB?"  
    ENTER @Hp4286;A

### **[:EVENT]?**

STATus:INSTRument[:EVENT]?

Returns the contents of the Instrument Event Status Register. (Query Only)

Reading the event register clears it.

Query response is <numeric>.

Example                            OUTPUT @Hp4286;"STAT:INST?"  
    ENTER @Hp4286;A

## STATus:INSTrument

### STATus:OPERation

### Subsystem

This subsystem is used to control the Operation Status Registers (Operation Event Status Register, Operation Event Status Enable Register, and Operation Status Condition Register) and the transition filters.

#### :CONDition?

STATus:OPERation:CONDition?

Queries the contents of Operation Status Condition Register. (Query only)

Query response is *<numeric>*.

Example                    OUTPUT @Hp4286;"STAT:OPER:COND?"  
                             ENTER @Hp4286;A

#### :ENABle <numeric>

STATus:OPERation:ENABle <numeric>

Sets the contents of the Operation Status Enable Register.

Parameter	Description
<i>&lt;numeric&gt;</i>	Decimal expression of the contents of the register; 0 to 65535 ( $=2^{16}-1$ )

Query response is *<numeric>*.

Example                    OUTPUT @Hp4286;"STAT:OPER:ENAB 1"  
                             OUTPUT @Hp4286;"STAT:OPER:ENAB?"

#### [:EVENT]?

STATus:OPERation[:EVENT]?

Queries the contents of the Operation Status Event Register. (Query only)

Reading the event register clears it.

Query response is *<numeric>*.

Example                    OUTPUT @Hp4286;"STAT:OPER?"  
                             ENTER @Hp4286;A

**:NTRansition <numeric>**

STATus:OPERation:NTRansition <numeric>

Sets the negative transition filter of the Operation Status Register.

Parameter	Description
<numeric>	Decimal expression of the contents of the register; 0 to 65535 ( $=2^{16}-1$ )

Query response is <numeric>.

Example

```
OUTPUT @Hp4286;"STAT:OPER:NTR 1"
OUTPUT @Hp4286;"STAT:OPER:NTR?"
ENTER @Hp4286;A
```

**:PTRansition <numeric>**

STATus:OPERation:PTRansition <numeric>

Sets the positive transition filter of the Operational Status Register.

Parameter	Description
<numeric>	Decimal expression of the contents of the register; 0 to 65535 ( $=2^{16}-1$ )

Query response is <numeric>.

Example

```
OUTPUT @Hp4286;"STAT:OPER:PTR 1"
OUTPUT @Hp4286;"STAT:OPER:PTR?"
ENTER @Hp4286;A
```

**STATus:PRESet**

STATus:PRESet

Presets the Operation and Questionable Status Enable Registers and transition filters. This command does not affect the Event Registers. (No query)

This command does not affect the Instrument Event Status Register and the Enable Register.

Register/Filter	Preset value
Enable register	0
Positive transition filter	65535 ( $2^{16}-1$ : all bits are 1)
Negative transition filter	0

## STATUS:PRESet

### STATUS:QUESTIONable

### Subsystem

---

This subsystem is used to control the Questionable Status Registers (Questionable Event Status Register, Questionable Event Status Enable Register, and Questionable Status Condition Register). The LCR meter has no operation that reports an event to the Questionable Status Register.

#### **:CONDition?**

STATUS:QUESTIONable:CONDition?

Queries the contents of the Questionable Status Condition Register. (Query only)

The LCR meter has no operation that reports an event to the questionable register.

#### **:ENABle** <numeric>

STATUS:QUESTIONable:ENABle <numeric>

Sets the value of the Questionable Status Enable Register.

The LCR meter has no operation that reports an event to the questionable register.

#### **[:EVENT]?**

STATUS:QUESTIONable[:EVENT]?

Queries the contents of the Questionable Event Status Register. (Query only)

The LCR meter has no operation that reports an event to the questionable register.



## SYSTem Subsystem

### SYSTem:BEEPer{1|2}:STATe {OFF|ON|0|1}

SYSTem:BEEPer{1|2}:STATe {OFF|ON|0|1}

Sets an annunciator that sounds to indicate completion of certain operations or as a warning.

(BEEP DONE ON off, BEEP WARN ON off under (System))

Node	Description
BEEPer1	An annunciator that sounds to indicate completion of certain operations such as calibration or instrument state save. (BEEP DONE ON off under (System))
BEEPer2	The warning annunciator. When the annunciator is ON, it sounds a warning when a cautionary message is displayed. (BEEP WARN ON off under (System))

Parameter	Description
OFF or 0	Beeper OFF
ON or 1	Beeper ON

Query response is {0|1}.

```
Example          OUTPUT @Hp4286;"SYST:BEEP1:STAT ON"
                  OUTPUT @Hp4286;"SYST:BEEP1:STAT?"
                  ENTER @Hp4286;A
```

## SYSTem:COMMunicate:GPIB

## Subsystem

This subsystem sets the GPIB address that the LCR meter will use to communicate with other devices on the GPIB.

### :CONTroller:ADDRess <numeric>

SYSTem:COMMunicate:GPIB:CONTroller:ADDRess <numeric>

Sets the GPIB address that the LCR meter will use to communicate with the external controller. (ADDRESS: CONTROLLER under (Local))

Parameter	Range	Unit
<numeric>	0 to 30	—

Query response is <numeric>.

```
Example          OUTPUT @Hp4286;"SYST:COMM:GPIB:CONT:ADDR 8"
                  OUTPUT @Hp4286;"SYST:COMM:GPIB:CONT:ADDR?"
                  ENTER @Hp4286;A
```

### **:RDEvice{1|2}:ADDRESS <numeric>**

SYSTem.COMMunicate:GPIB:RDEvice{1|2}:ADDRess <numeric>

Sets the GPIB address that the LCR meter will use to communicate with the plotter or printer. (ADDRESS: PLOTTER under (Local))

Node	Description
:RDEvice1	plotter
:RDEvice2	printer

Parameter	Range	Unit
<numeric>	0 to 30	—

Query response is <numeric>.

Example                                    OUTPUT @Hp4286;"SYST:COMM:GPIB:RDEV1:ADDR 5"  
    OUTPUT @Hp4286;"SYST:COMM:GPIB:RDEV1:ADDR?"  
    ENTER @Hp4286;A

## **SYSTem:COMMunicate:PARallel[1]**

## **Subsystem**

This subsystem controls the 8-bit parallel I/O port.

### **[:RECeive]:DATA?**

SYSTem.COMMunicate:PAPallel[:RECeive]:DATA?

Reads data from the 4-bit parallel input of the LCR meter. (Query only)

Query response is <numeric>.

Example                                    OUTPUT @Hp4286;"SYST:COMM:PAR:DATA?"  
    ENTER @Hp4286;A

### **:TRANsmit:DATA <numeric>**

SYSTem.COMMunicate:PAPallel:TRANsmit:DATA <numeric>

Outputs the data to the 8-bit parallel output port. (No query)

Parameter	Range	Unit
<numeric>	0 to 255	—

Example                                    OUTPUT @Hp4286;"SYST:COMM:PAR:TRAN:DATA 0"

**SYSTem:COMMunicate:PARallel2**

**Subsystem**

This subsystem controls the handler interface I/O port.

**:MODE {0|1|2}**

SYSTem:COMMunicate:PARallel2:MODE {0|1|2}

Sets the output mode for the handler interface I/O port. (No query)

Parameter	Description
0	The port OFF
1	Outputs the data in comparator mode
2	Outputs the data in list sweep mode

Query response is {0|1}.

Example                                 OUTPUT @Hp4286;"SYST:COMM:PAR2:MODE 2"

**SYSTem:DATE <numeric(year)>,<numeric(month)>,<numeric(day)>**

SYSTem:DATE <numeric(year)>,<numeric(month)>,<numeric(day)>

Sets the date of the internal clock. (DATE MM/DD/YY under **System**)

Parameter	Range	Unit
<numeric(year)>	1900 to 2099	—
<numeric(month)>	1 to 12	—
<numeric(day)>	1 to 31	—

Query response is <numeric(year)><numeric(month)><numeric(day)>.

Example                                 OUTPUT @Hp4286;"SYST:DATE 1993,1,1"  
   OUTPUT @Hp4286;"SYST:DATE?"  
   ENTER @Hp4286;A,B,C

**SYSTem:DATE:MODE {MDY|DMY}**

SYSTem:DATE:MODE {MDY|DMY}

Changes the displayed date to the “month:day:year” format or the “day:month:year” format.

Parameter	Description
MDY	“month:day:year” format. (MonDayYear under <b>System</b> )
DMY	“day:month:year” format. (DayMonYear under <b>System</b> )

Query response is {MDY|DMY}.

## SYSTem:DATE:MODE

Example                    OUTPUT @Hp4286;"SYST:DATE:MODE DMY"  
                          OUTPUT @Hp4286;"SYST:DATE:MODE?"  
                          ENTER @Hp4286;A\$

## SYSTem:ERRor?

SYSTem:ERRor?

Outputs the error message in the error queue.

Query response is <numeric (Error number)> <string (Error message)>.

Example                    DIM A\$ [50]  
                          OUTPUT @Hp4286;"SYST:ERR?"  
                          ENTER @Hp4286;A,A\$

## SYSTem:FIXTure {NONE|16191|16192|16193|16194|UDEFI ned}

SYSTem:FIXTure {NONE|16191|16192|16193|16194| UDEFI ned}

Specifies the fixture in use in order to select which electrical length (recorded in the LCR meter) is to be used. ( FIXTURE [ ] under (Cal) )

Parameter	Description
NONE	No test fixture (FIXTURE: NONE)
16191	16191A (16191)
16192	16192A (16192)
16193	16193A (16193)
16194	16194A (16194)
UDEFI ned	User-defined test fixture (USER)

Query response is {NONE|16191|16192|16193|16194|UDEFI}.

## SYSTem:FIXTure:DIStance <numeric>

SYSTem:FIXTure:DIStance <numeric>

Sets the electrical length of the user-modified fixture. (DEFINE EXTENSION under (Meas) FIXTURE [ ] )

Parameter	Range	Unit
<numeric>	-10 to 10 (m)	-

Query response is <numeric>.

## SYSTem:FIXTure:LABel <string>

SYSTem:FIXTure:LABel <string>

Modifies the label of the user-modified fixture. (LABEL FIXTURE under (Meas) FIXTURE [ ] )

**SYSTem:FIXTure:SAVE**

SYSTem:FIXTure:SAVE

Saves the settings of the user-modified fixture. (SAVE USER FXTR KIT under **Meas** FIXTURE [ ] ;No query)

**SYSTem:KEY <numeric>**

SYSTem:KEY <numeric>

Sends the key code for a key or a softkey on the front panel. This is equivalent to actually pressing a key.

Do not use multiple commands in a program line (for example, OUTPUT @Hp4286;"SYST:KEY 0;KEY 1"). If you do, the LCR meter's operation cannot keep up with the program.

Parameter	Description
<numeric>	Key code, 0 to 52

Query response is <numeric>. (The last key operation)

Example LOCAL @Hp4286  
OUTPUT @Hp4286;"SYST:KEY 0"

**SYSTem:KLOCK {OFF|ON|0|1}**

SYSTem:KLOCK {OFF|ON|0|1}

Locks the front panel keys and the rotary knob.

Parameter	Description
OFF or 0	Unlock the front panel keys.
ON or 1	Lock the front panel keys.

Query response is {0|1}.

Example OUTPUT @Hp4286;"SYST:KLOC ON"  
OUTPUT @Hp4286;"SYST:KLOC?"  
ENTER @Hp4286;A

**SYSTem:PRESet**

SYSTem:PRESet

Presets the LCR meter to the preset default values. See Appendix C of the *Function Reference* for the default values. This command does *not* preset the HP Instrument BASIC. (**PRESET**); No query)

## **SYSTem:TIME** <numeric(hour)>,<numeric(min)>,<numeric(sec)>

SYSTem:TIME <numeric(hour)>,<numeric(min)>,<numeric(sec)>

Sets the time of the internal clock. (TIME HH:MM:SS under (System) SET CLOCK)

Parameter	Description
<numeric(hour)>	Hour, 0 to 23
<numeric(min)>	Minute, 0 to 59
<numeric(sec)>	Second, 0 to 59

Query response is <numeric(hour)>,<numeric(min)>,<numeric(sec)>.

Example                                   OUTPUT @Hp4286;"SYST:TIME 10,30,0"  
  OUTPUT @Hp4286;"SYST:TIME?"  
  ENTER @Hp4286;A,B,C

## **SYSTem:VERSion?**

SYSTem:VERSion?

Queries the SCPI version to which the LCR meter conforms. (Query only)

Query response is <string>. (For example, 1993.0)

Example                                   DIM A\$[10]  
  OUTPUT @Hp4286;"SYST:VERS?"  
  ENTER @Hp4286;A\$

## TRIGger Subsystem

The TRIGger subsystem is used to control the trigger event of the trigger system.

Related Command            INITiate:CONTinuous  
                               INITiate[:IMMediate]  
                               INITiate[:IMMediate]:AGain:ALL

### TRIGger:EVENT:TYPE {POINT|SWEep}

TRIGger:EVENT:TYPE {POINT|SWEep}

Selects the trigger event mode. (TRIG EVENT [ ] under (Trigger))

Parameter	Description
POINT	Trigger event on point <sup>1</sup>
SWEep	Trigger event on sweep

Query Response is {POINT|SWE}.

<sup>1</sup> Available only when the trigger source is GPIB, Manual, or External trigger.

Example                    OUTPUT @Hp4286;"TRIG:EVENT:TYPE SWE"  
                               OUTPUT @Hp4286;"TRIG:EVENT:TYPE?"  
                               ENTER @Hp4286;A\$

### TRIGger:SLOPe {POSitive|NEGative}

TRIGger:SLOPe {POSitive|NEGative}

Sets the trigger signal polarity of an external signal connected to the rear panel EXT TRIGGER input. (TRIG PLRTY pos neg under (Trigger))

Parameter	Description
POSitive	Positive trigger (low-to-high transition)
NEGative	Negative trigger (high-to-low transition)

Query Response is {POS|NEG}.

Example                    OUTPUT @Hp4286;"TRIG:SLOP POS"  
                               OUTPUT @Hp4286;"TRIG:SLOP?"  
                               ENTER @Hp4286;A\$

### TRIGger:SOURce {BUS|EXTernal|INTernal|MANual}

TRIGger:SOURce {BUS|EXTernal|INTernal|MANual}

Selects the trigger source, which is common to both channels. (TRIGGER: [ ] under (Trigger))

Parameter	Description
BUS	GPIB trigger (*TRG)
EXTernal	External trigger input from EXT TRIGGER input on the rear panel
INTernal	Internal trigger
MANual	Manual trigger from front panel key

Query Response is {BUS|EXT|INT|MAN}.

---

## Common Commands

### \*CLS

\*CLS

Clears the Status Byte Register, and the Operation Event Status Register, the Standard Event Status Register, and the Instrument Event Status Register. (No query)

Example                    OUTPUT @Hp4286;"\*CLS"

### \*ESE <numeric>

\*ESE <numeric>

Sets the enable bits of the Standard Event Status Register.

Parameter	Description
<numeric>	0 to 255 (decimal expression of enable bits of the operation status register)

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"\*ESE 1"  
                          OUTPUT @Hp4286;"\*ESE?"  
                          ENTER @Hp4286;A

### \*ESR?

\*ESR?

Returns the contents of the Standard Event Status Register. (Query only)

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"\*ESR?"  
                          ENTER @Hp4286;A

### \*IDN?

\*IDN?

Returns a string that represents the analyzer's ID.

Query response is <manufacturer>,<model>,<serial no.>,<firmware rev.>.

Where,

<manufacturer>	Agilent Technologies.
<model>	4291A.
<serial no.>	Serial Number, like JP1KA00101.
<firmware rev.>	Firmware revision, like REV1.00.

Example                    OUTPUT @Hp4286;"\*IDN?"  
                          ENTER @Hp4286;A\$



**\*OPC**

\*OPC

Sets the Operation Complete bit (bit 0 of the Standard Event Status Register) to 1 when it completes all pending operations.

\*OPC? query returns an ASCII character 1 when all pending operations have been completed.

Query response is 1.

```
Example          OUTPUT @Hp4286;"*OPC"
                  OUTPUT @Hp4286;"*OPC?"
                  ENTER @Hp4286;A
```

**\*OPT?**

\*OPT?

Queries the options installed. (Query only)

Query response is <option>s.

Where <option>s are,  
           102           Add HP IBASIC, HP-HIL keyboard and cable.

```
Example          OUTPUT @Hp4286;"*OPT?"
                  ENTER @Hp4286;A$
```

**\*PCB <numeric>**

\*PCB <numeric>

Specifies the address of a controller that is temporarily passing GPIB control to the analyzer. (No query)

Parameter	Description
<numeric>	0 to 30

```
Example          OUTPUT @Hp4286;"*PCB 0"
```

**\*RST**

\*RST

Resets the analyzer to its default values, (see the *Function Reference* for information on the default values), stops sweeping and taking data, and resets the HP Instrument BASIC . (No query)

**\*SRE <numeric>**

\*SRE <numeric>

Sets the contents of the Status Byte Enable Register.

Parameter	Description
<numeric>	0 to 255 (decimal expression of the contents of the Service Request Enable Register)

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"\*SRE 1"  
                              OUTPUT @Hp4286;"\*SRE?"  
                              ENTER @Hp4286;A

**\*STB?**

\*STB?

Returns the contents of the Status Byte Register. The content of the register's bit 6 is the master summary status bit. (Query only)

Query response is <numeric>.

Example                    OUTPUT @Hp4286;"\*STB?"  
                              ENTER @Hp4286;A

**\*TRG**

\*TRG

Triggers the analyzer when the trigger mode is set to BUS trigger. (No query)

Example                    To trigger a measurement,  
                              OUTPUT @Hp4286;"INIT:CONT ON"  
                              OUTPUT @Hp4286;"TRIG:SOUR BUS"  
                              OUTPUT @Hp4286;"\*TRG"

**\*TST?**

\*TST?

Executes an internal self-test and returns the test result. (Query only)

Query response is {0|1}.

Where,

0                    : Pass  
1                    : Fail

Example                    OUTPUT @Hp4286;"\*TST?"  
                              ENTER @Hp4286;A

**\*WAI**

\*WAI

Makes the analyzer wait until all previously sent commands are completed. (No query)



## Using Status Reporting System

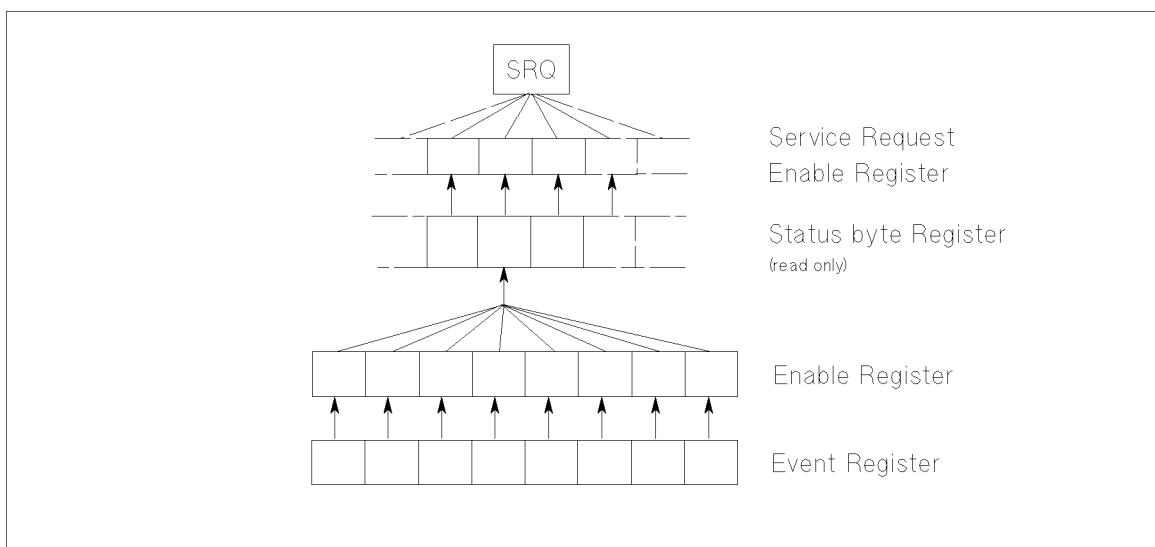
The analyzer has status registers that report system conditions. The register contents are changed depending on the particular condition of the analyzer. By reading this register, you can determine the specific analyzer status.

This chapter provides the following information:

- General status register model.
- Status register structure.
- 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 4-1. General Status Register Model**

The status reporting system has a hierarchical structure as shown in Figure 4-1. When the analyzer condition satisfies the particular condition, the corresponding bit of the event register is set TRUE. Therefore, you can check the analyzer condition by reading the event register.

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

If the corresponding bit of the service request enable register is TRUE, 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. You can program for an interruption request by the SRQ. See “SRQ and Interrupt” for more details about the program requirements.

## **Event Register**

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

You cannot change bit status by GPIB command.

The analyzer has the following event registers:

- Instrument Event Status Register (see Table 4-2 for details).
- Standard Event Status Register (see Table 4-3).
- Operation Status Event Register (see Table 4-4).

## **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 \*STB? query by the controller. SPOLL reads the status byte register value directly without being set the analyzer 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 for the SPOLL clears RQS bit. Table 4-1 lists the contents of the status byte register.

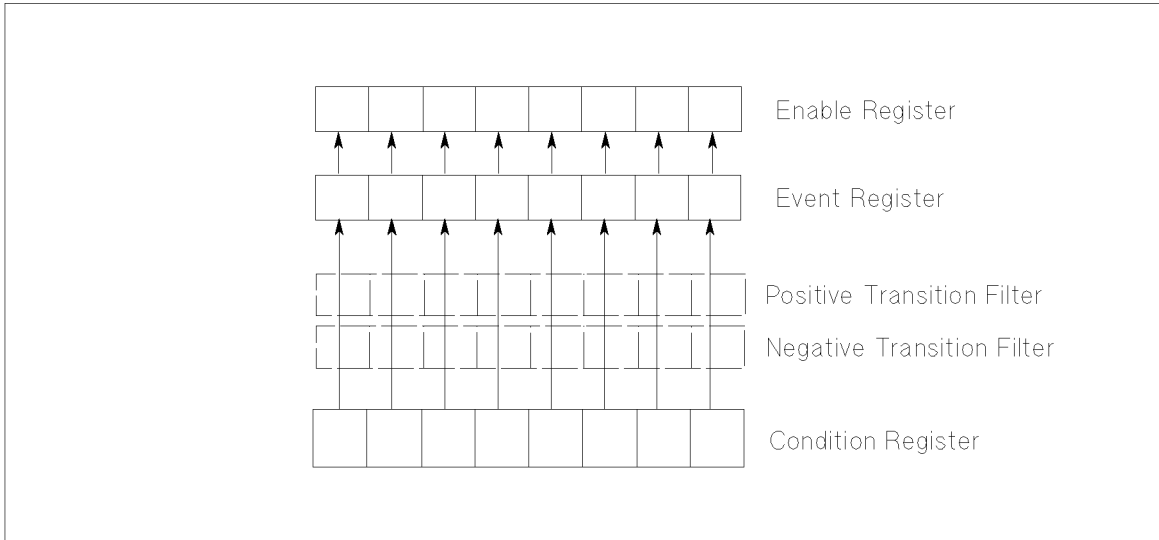
A serial poll initiated by using the SPOLL command reads bit 6 of the status byte register as the RQS bit. The \*STB? command reads bit 6 as the MSS bit.

SRQ (Service Request) can be generated 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 to select which transitions of the analyzer condition to 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 in between the event register and 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, 1 is set in the event register by changing 1 to 0 in the event register.

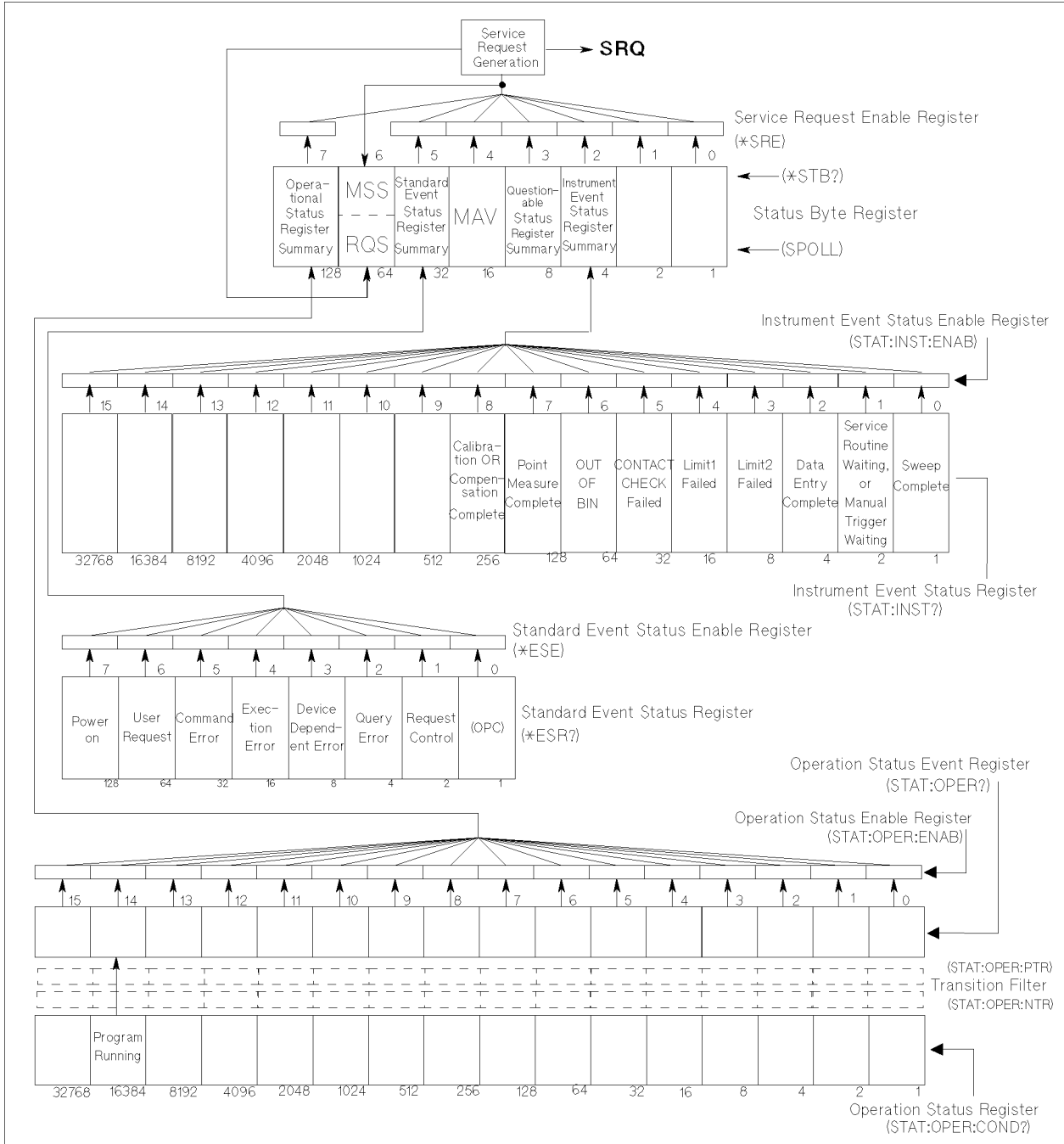


**Figure 4-2. Transition Filter and Condition Register**

For the 4286A, only “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

The status register has a hierarchical structure. The status byte register summarizes the low level registers.



L3404001

Figure 4-3. Status Register Structure

## 4-4 Using Status Reporting System

**Table 4-1. Status Bit Definitions of the Status Byte Register**

<b>Bit</b>	<b>Name</b>	<b>Description</b>
2	Instrument Event Status Register Summary Bit	One of the enabled bits in Instrument Event Status Register has been set.
3	Questionable Status Register Summary Bit	The analyzer has no operation to report the event to the Questionable Status Register group. This register is available to maintain consistency with other SCPI compatible devices.
4	MAV (Message Available)	An information has been prepared to be output, but it has not been read yet.
5	Standard Event Status Register Summary Bit	One of the enabled bits in the Standard Event Status Register has been set.
6	RQS	The analyzer generated an SRQ. When reading the status byte register by SPOLL, bit 6 is RQS bit.
	MSS	The analyzer has at least one TRUE bit in the status byte register that is enabled by the service request enable register. When reading the status byte register by *STB?, bit 6 is MSS bit.
7	Operation Status Register Summary Bit	One of the enabled bits in the Operation Status Register has been set.

A status bit is cleared when \*CLS command is executed.



**Table 4-2.  
Status Bit Definitions of the Instrument Event Status Register**

<b>Bit</b>	<b>Name</b>	<b>Description</b>
0	Sweep Complete	A sweep has been completed since the last read of the register.
1	Service Routine Waiting or Manual Trigger Waiting	1. An internal service routine has completed an operation or is waiting for an operator response.  2. The analyzer has set the manual trigger mode and is waiting for a manual trigger.
2	Data Entry Complete	A terminator key has been pressed.
3	Limit 2 Failed	Limit test 2 was failed.
4	Limit 1 Failed	Limit test 1 was failed.
5	Contact Check Failed	Contact check was failed
6	Out of BIN	Measurement result was out of limits of all bins.
7	Point Measurement Complete <sup>1</sup>	One measurement point of a sweep has been completed.
8	Calibration or Compensation Complete	The analyzer completed a calibration or compensation process including a measurement and a calculation for the coefficients, and ready to perform next process.

<sup>1</sup> This bit is set only when both related bits of the service request enable register and the instrument event status enable register are enabled.

**Table 4-3. Status Bit Definitions of the Standard Event Status Register**

<b>Bit</b>	<b>Name</b>	<b>Description</b>
0	Operation Complete	The analyzer has completed all pending operations and is ready to accept new commands. This bit is generated only in response to the *0PC command.
1	Request Control	The analyzer requires control of GPIB to proceed the current operation.
2	Query Error	<ol style="list-style-type: none"> <li>1. The analyzer has been addressed to talk, but there is nothing in the output queue to transmit.</li> <li>2. Data in the Output Queue has been lost.</li> </ol>
3	Device Dependent Error	An error, other than a command error, a query error, or an execution error has occurred.
4	Execution Error	<ol style="list-style-type: none"> <li>1. An GPIB command parameter exceeded its input range, or is inconsistent with the analyzer's capabilities.</li> <li>2. An GPIB command could not be properly executed due to some analyzer condition.</li> </ol>
5	Command Error	<ol style="list-style-type: none"> <li>1. An IEEE 488.2 syntax error has occurred. The analyzer receives a command that did not follow the syntax defined by the IEEE 488.2 standard. Possible violations include, a command parameter violated the analyzer listening formats or a command parameter type is unacceptable to the analyzer.</li> <li>2. A semantic error occurred. For example, the analyzer received an incorrectly spelled command. Another example would be that the analyzer received an optional 488.2 command that is not implemented to the analyzer.</li> <li>3. The analyzer received a Group Execute Trigger (GET) inside a GPIB command.</li> </ol>
6	User Request	The operator pressed a front panel key or an optional keyboard key or turned the rotary knob.
7	Power ON	A power-on sequence has occurred since the register was last read.

**Table 4-4. Status Bit Definitions of the Operation Status Register**

<b>Bit</b>	<b>Name</b>	<b>Description</b>
14	Program running	An HP Instrument BASIC program is running.

---

## 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.

<pre>OUTPUT @Hp4286;"STAT:INST?" ENTER @Hp4286;Esb IF BIT(Esb,4) THEN   DISP "LIMIT TEST FAILED AT Ch 1." END IF</pre>	<p><i>Queries instrument event status register contents.</i></p> <p><i>Retrieve return value.</i></p> <p><i>If bit 4 of the instrument event status register is set to 1, the limit test failed on channel 1.</i></p>
--	---

#### Module 4-. Reading an Event Register

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

*STB?	Returns Status Byte Register contents.
*ESR?	Returns Event Status Register contents.
STAT:INST?	Returns Instrument Event Status Register contents.
STAT:OPER:COND?	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 to 1 the enable register for the correspondent event register bit.
3. Set to 1 the service request enable register bit for the correspondent status byte register bit.
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.

<pre>ON INTR Scode GOTO Sweep_end OUTPUT @Hp4286; "STAT:INST:ENAB 1" OUTPUT @Hp4286; "*SRE 4" OUTPUT @Hp4286; "*CLS" OUTPUT @Hp4286; "*OPC?" ENTER @Hp4286; 0pc ! OUTPUT @Hp4286; "TRIG:SOUR INT" OUTPUT @Hp4286; "INIT:CONT OFF" OUTPUT @Hp4286; "ABOR" ENABLE INTR Scode;2 OUTPUT @Hp4286; "INIT" Waiting: GOTO Waiting Sweep_end: ! IF NOT BIT(SPOLL(@Hp4286),6) THEN     ENABLE INTR Scode;2     GOTO Waiting END IF</pre>	<p><i>When the SRQ has occurred, jumps to label, "Sweep_end."</i></p> <p><i>Set bit 0 of the instrument event status enable register to 1. (<math>2^0 = 1</math>)</i></p> <p><i>Set bit 2 of the service request enable register to 1. (<math>2^2 = 4</math>)</i></p> <p><i>Clears the event register.</i></p> <p><i>Confirms the clear operation is completed.</i></p> <p><i>Selects internal trigger source.</i></p> <p><i>Sets the continuous mode OFF.</i></p> <p><i>Moves trigger sequence to the idle state.</i></p> <p><i>Enables the SRQ interruption just before triggering.</i></p> <p><i>Trigger a measurement.</i></p> <p><i>Waits until SRQ is generated.</i></p> <p><i>When on SRQ, program jumps to this label.</i></p> <p><i>Check the SRQ is generated from the target device by checking status byte register bit 6.</i></p> <p><i>If not, enable the SRQ again, then returns to the endless looping.</i></p>
--	---

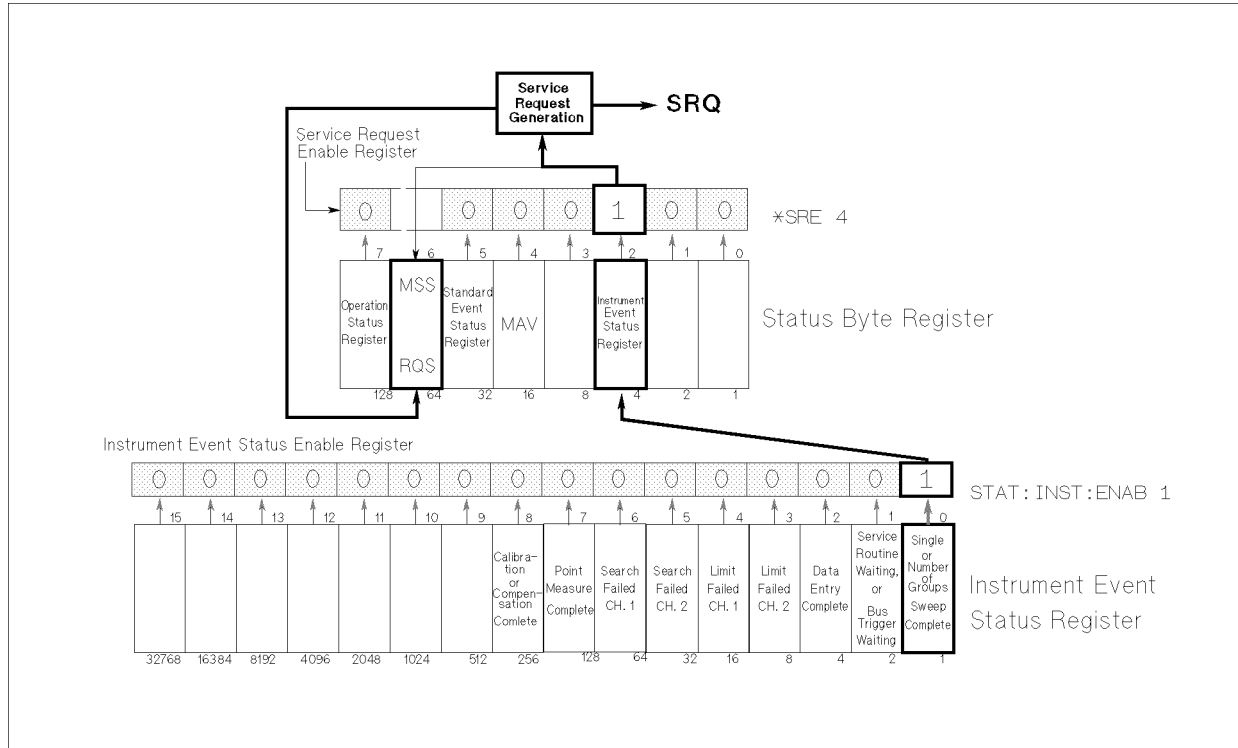
### Module 4-. Detecting Sweep End Using SRQ and Interrupt

#### 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 SYST:PRES command.

Figure 4-4 shows the SRQ generation sequence of the example above.



**Figure 4-4. 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.
- `STAT:INST:ENAB decimal` Sets the enable register for instrument event status register.
- `STAT:OPER:ENAB decimal` Sets the enable register for operation status register.
- `STAT:OPER:PTR decimal` Sets the transition filter to positive for operation status register.
- `STAT:OPER:NTR decimal` Sets the transition filter to negative for operation status register.

## 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 4286A RF Impedance/Material Analyzer serial number prefix listed on the title page of this manual.

### Manual Changes

To adapt this manual to your 4286A, 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 with 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**

Version	Make Manual Changes

**Table A-2. Manual Changes by Firmware Version**

Version	Make Manual Changes
REV.1.xx	CHANGE 1

---

## Change 1

The firmware revision 1.xx does not support the following commands. Please delete the descriptions of the following commands.

- CALCulate:COMParator:COUNT:CLEar
- CALCulate:COMParator:COUNT[:STATE] {ON|OFF}
- DATA[:DATA] BCOU?
- DISP:DATA:PARAmeter1:MSD
- DISP:DATA:PARAmeter1:MSD:ALL
- DISP:DATA:PARAmeter1:MSD:AUTO
- DISP:DATA:PARAmeter1:MSD:FIX {ON|OFF}
- DISPLay[:WINDow]:BCOUnT:STATe {ON|OFF}
- MMEMory:DELeTe:CAL
- MMEMory:LOAD:CAL
- MMEMory:STOre:CAL:AUTO {ON|OFF}
- SENSE:CORR{1|2}:CKIT:STANdard3 {Q|F}
- SENSE:CORR{1|2}:LOAD:TYPE {RL|LQF}
- SENSE:LIST:MODIfy:SEGMENT{1-10}:AVERAge:COUNT
- SENSE:LIST:MODIfy:SEGMENT{1-10}:FREQUency

---

## 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 Program

---

### Example 1: BIN sort at a single frequency

#### Measurement Conditions

- **Measurement parameter** :  $L_s$ -Q
- **Measurement frequency** : 100 MHz
- **OSC level** : 10 mA
- **Contact check** : ON
  - Upper limit : 10  $\Omega$
  - Lower limit : 1  $\Omega$
- **Comparater (BIN)**
  - Parameter : L
    - BIN 1 : 10 nH $\pm$ 1%
    - BIN 2 : 10 nH $\pm$ 2%
    - BIN 3 : 10 nH $\pm$ 3%
    - BIN 4 : 10 nH $\pm$ 4%
    - BIN 5 : 10 nH $\pm$ 5%
    - BIN 6 : 10 nH $\pm$ 6%
    - BIN 7 : 10 nH $\pm$ 7%
    - BIN 8 : 10 nH $\pm$ 8%
    - BIN 9 : 10 nH $\pm$ 9%
  - Secondary parameter : Q
    - Upper limit : 20
    - Lower limit : 5
- **Display** : OFF
- **Calibration** : OPEN/SHORT/LOAD cal using working standard set
- **Output** : Measurement value, BIN sort result, contact check result

## EXAMPLE 1

```
10 !**** INITIALIZE ****
20 DIM A(1:2,1:2)
30 !
40 Osc_level=.01      ! [A]
50 Meas_frequency=1.E+8 ! [Hz]
60 !
70 ASSIGN @Hp4286 TO 800
80 Scode=8
90 CLEAR @Hp4286
100 ABORT Scode
110 OUTPUT @Hp4286;"DISP:ALL HIHB"
120 OUTPUT @Hp4286;"CALC:COMP:SAVE"
130 !
140 ! **** Set measurment parameter ****
150 OUTPUT @Hp4286;"CALC:FORM LSQ"
160 !
170 ! **** Sweep Setup ****
180 OUTPUT @Hp4286;"SENS:LIST:CLE"
190 OUTPUT @Hp4286;"SENS:LIST:SEGM:EDIT"
200 OUTPUT @Hp4286;"SENS:LIST:SEGM:FREQ";Meas_frequency
210 OUTPUT @Hp4286;"SENS:LIST:SEGM:SAVE"
220 OUTPUT @Hp4286;"SENS:LIST:SAVE"
230 !
240 ! ***** Set OSC Level *****
250 OUTPUT @Hp4286;"SOUR:CURR";Osc_level
260 !
270 ! **** Setup Contact Check ****
280 OUTPUT @Hp4286;"CALC:RDC:LIM:UPP 10"! [OHM] Upper Limit
290 OUTPUT @Hp4286;"CALC:RDC:LIM:LOW 1"! [OHM] Lower Limit
300 OUTPUT @Hp4286;"CALC:RDC:LIM:STAT ON"
310 OUTPUT @Hp4286;"SENS:RDC ON"
320 !
330 !**** CALIBRATION STATE CHEKING ****
340 OUTPUT @Hp4286;"SENS:CORR1?"
350 ENTER @Hp4286;Stat
360 IF Stat THEN
370   INPUT "ARE YOU SURE TO RE-CALIBRATE NOW? [Y/N]",Ans$
380   IF Ans$"Y" THEN Skip_cal
390 END IF
400 !
410 !**** SELECT USER CAL *****
420 OUTPUT @Hp4286;"SENS:CORR1:COLL:FPO USER"
430 !
440 !**** USER CAL KIT SETTING *****
450 OUTPUT @Hp4286;"SENS:CORR:CKIT UDEF"
460 INPUT "INPUT Rs OF WORKING STANDARD",Rs
470 OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:R ";Rs
480 INPUT "INPUT Ls OF WORKING STANDARD",Ls
490 OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:L ";Ls
500 !
510 !**** SETTING ENABLE REGISTER FOR SRQ ****
520 OUTPUT @Hp4286;"STAT:INST:ENAB 256"
530 OUTPUT @Hp4286;"*SRE 4"
```

```

540 OUTPUT @Hp4286;"TRIG:SOUR INT"
550 !
560 !**** OPEN STD. MEASUREMENT ****
570 ON INTR Scode GOTO Open_end
580 OUTPUT @Hp4286;"*CLS;*OPC?"
590 ENTER @Hp4286;0pc
600 INPUT "MAKE OPEN, THEN PRESS [RETURN]",Dummy
610 ENABLE INTR Scode;2
620 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN1"
630 Wait_open:GOTO Wait_open
640 Open_end:!
650 !
660 !**** SHORT STD. MEASUREMENT ****
670 ON INTR Scode GOTO Short_end
680 OUTPUT @Hp4286;"*CLS;*OPC?"
690 ENTER @Hp4286;0pc
700 INPUT "MAKE SHORT, THEN PRESS [RETURN]",Dummy
710 ENABLE INTR Scode;2
720 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN2"
730 Wait_short:GOTO Wait_short
740 Short_end:!
750 !
760 !**** LOAD STD. MEASUREMENT ****
770 ON INTR Scode GOTO Load_end
780 OUTPUT @Hp4286;"*CLS;*OPC?"
790 ENTER @Hp4286;0pc
800 INPUT "CONNECT LOAD, THEN PRESS [RETURN]",Dummy
810 ENABLE INTR Scode;2
820 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN3"
830 Wait_load:GOTO Wait_load
840 Load_end:!
850 !
860 OUTPUT @Hp4286;"SENS:CORR1:COLL:SAVE"
870 !
880 Skip_cal:!
890 !
900 ! **** Initialize BIN table *****
910 OUTPUT @Hp4286;"CALC:COMP:CLE"
920 OUTPUT @Hp4286;"CALC:COMP:CONT:SEGM 1"
930 OUTPUT @Hp4286;"CALC:COMP:SEGM:EDIT"
940 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
950 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.1E-9"
960 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.9E-9"
970 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
980 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
990 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.2E-9"
1000 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.8E-9"
1010 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1020 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1030 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.3E-9"
1040 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.7E-9"
1050 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1060 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1070 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.4E-9"
1080 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.6E-9"

```

```

1090 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1100 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1110 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.5E-9"
1120 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.5E-9"
1130 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1140 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1150 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.6E-9"
1160 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.4E-9"
1170 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1180 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1190 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.7E-9"
1200 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.3E-9"
1210 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1220 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1230 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.8E-9"
1240 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.2E-9"
1250 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1260 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1270 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.9E-9"
1280 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.1E-9"
1290 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1300 OUTPUT @Hp4286;"CALC:COMP:SAVE"
1310 OUTPUT @Hp4286;"CALC:COMP:STAT ON"
1320 !
1330 !***** Set secondary limit parameter
1340 OUTPUT @Hp4286;"CALC:COMP:SREJ:LOW 5"! Lower Limit
1350 OUTPUT @Hp4286;"CALC:COMP:SREJ:UPP 20"! Upper Limit
1360 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1370 OUTPUT @Hp4286;"CALC:COMP:SREJ:STAT ON"
1380 OUTPUT @Hp4286;"CALC:COMP:SAVE"
1390 !
1400 !**** MEAS DATA DISPLAY OFF *****
1410 OUTPUT @Hp4286;"DISP:TRAC:STAT OFF"
1420 !
1430 !**** BINARY FORMAT TRANSFER SETTING ****
1440 ASSIGN @Dt TO 800;FORMAT OFF
1450 OUTPUT @Hp4286;"FORM:DATA REAL,64"
1460 INTEGER Head1,Head2,Head3,Head4,Term
1470 !
1480 ! **** Measurement *****
1490 OUTPUT @Hp4286;"INIT:CONT OFF"
1500 OUTPUT @Hp4286;"ABOR"
1510 ON INTR Scode GOTO Meas_end
1520 OUTPUT @Hp4286;"STAT:INST:ENAB 1;*SRE 4;"
1530 OUTPUT @Hp4286;"*CLS;*OPC?"
1540 ENTER @Hp4286;Opc
1550 INPUT "CONNECT DUT, THEN PRESS [RETURN]",Dummy
1560 ENABLE INTR Scode;2
1570 OUTPUT @Hp4286;"INIT"
1580 Wait_meas:GOTO Wait_meas
1590 Meas_end:!
1600 !
1610 ! **** Output Measurement and BIN sort results *****
1620 OUTPUT @Hp4286;"DATA? BIN"
1630 ENTER @Dt;Head1,Head2,Head3,Head4,Bin,Term

```

#### B-4 Sample Program

```
1640 OUTPUT @Hp4286;"DATA? RDCDT"  
1650 ENTER @Dt;Head1,Head2,Head3,Head4,A(*),Term  
1660 PRINT "          Ls =";A(2,1);"[H]"  
1670 PRINT "          Q =";A(2,2)  
1680 PRINT  
1690 PRINT "          BIN =";Bin;" (0=OUT OF BIN, 10=AUX BIN, -1=NO TEST)"  
1700 PRINT "CONTACT CHECK =";A(1,1);" (0=FAIL, 1=PASS, -1=NO TEST)"  
1710 PRINT  
1720 END
```

---

## Example 2: Two Measurement frequency points

### Measurement Conditions

- **Measurement parameter** :  $L_s$ -Q
- **Measurement frequency** : 100 MHz, 300 MHz
- **OSC level** : 10 mA
- **Contact check** : ON
  - Upper limit : 10  $\Omega$
  - Lower limit : 1  $\Omega$
- **Comparater (BIN)**
  - Frequency : 100 MHz
  - Parameter : L
    - BIN 1 : 10 nH $\pm$ 1%
    - BIN 2 : 10 nH $\pm$ 2%
    - BIN 3 : 10 nH $\pm$ 5%
    - BIN 4 : 10 nH $\pm$ 10%
    - BIN 5 : 10 nH $\pm$ 20%
  - Secondary parameter : Q
    - Secondary upper limit : 20
    - Secondary lower limit : 5
- **Limit test**
  - Frequency : 300 MHz
  - Parameter : L
    - Upper limit : 10 nH + 20%
    - Lower limit : 10 nH - 20%
  - Secondary parameter : Q
    - Upper limit : 50
    - Lower limit : 5
- **Display** : OFF
- **Calibration** : OPEN/SHORT/LOAD cal using working standard set
- **Output** : Measurement value, BIN sort result, contact check result

## Sample Program List 2

```
10 !**** INITIALIZE ****
20 DIM A(1:2,1:2)
30 !
40 Osc_level=.01      ![A]
50 Meas_frequency=1.E+8 ![Hz]
60 !
70 ASSIGN @Hp4286 TO 800
80 Scode=8
90 CLEAR @Hp4286
100 ABORT Scode
110 OUTPUT @Hp4286;"DISP:ALL HIHB"
120 OUTPUT @Hp4286;"CALC:COMP:SAVE"
130 !
140 ! **** Set measurment parameter ****
150 OUTPUT @Hp4286;"CALC:FORM LSQ"
160 !
170 ! **** Sweep Setup ****
180 OUTPUT @Hp4286;"SENS:LIST:CLE"
190 OUTPUT @Hp4286;"SENS:LIST:SEGM:EDIT"
200 OUTPUT @Hp4286;"SENS:LIST:SEGM:FREQ";Meas_frequency
210 OUTPUT @Hp4286;"SENS:LIST:SEGM:SAVE"
220 OUTPUT @Hp4286;"SENS:LIST:SAVE"
230 !
240 ! ***** Set OSC Level *****
250 OUTPUT @Hp4286;"SOUR:CURR";Osc_level
260 !
270 ! **** Setup Contact Check ****
280 OUTPUT @Hp4286;"CALC:RDC:LIM:UPP 10"! [OHM] Upper Limit
290 OUTPUT @Hp4286;"CALC:RDC:LIM:LOW 1"! [OHM] Lower Limit
300 OUTPUT @Hp4286;"CALC:RDC:LIM:STAT ON"
310 OUTPUT @Hp4286;"SENS:RDC ON"
320 !
330 !**** CALIBRATION STATE CHEKING ****
340 OUTPUT @Hp4286;"SENS:CORR1?"
350 ENTER @Hp4286;Stat
360 IF Stat THEN
370   INPUT "ARE YOU SURE TO RE-CALIBRATE NOW? [Y/N]",Ans$
380   IF Ans$"Y" THEN Skip_cal
390 END IF
400 !
410 !**** SELECT USER CAL *****
420 OUTPUT @Hp4286;"SENS:CORR1:COLL:FPO USER"
430 !
440 !**** USER CAL KIT SETTING *****
450 OUTPUT @Hp4286;"SENS:CORR:CKIT UDEF"
460 INPUT "INPUT Rs OF WORKING STANDARD",Rs
470 OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:R ";Rs
480 INPUT "INPUT Ls OF WORKING STANDARD",Ls
490 OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:L ";Ls
500 !
510 !**** SETTING ENABLE REGISTER FOR SRQ ****
520 OUTPUT @Hp4286;"STAT:INST:ENAB 256"
530 OUTPUT @Hp4286;"*SRE 4"
```



```

540 OUTPUT @Hp4286;"TRIG:SOUR INT"
550 !
560 !**** OPEN STD. MEASUREMENT ****
570 ON INTR Scode GOTO Open_end
580 OUTPUT @Hp4286;"*CLS;*OPC?"
590 ENTER @Hp4286;0pc
600 INPUT "MAKE OPEN, THEN PRESS [RETURN]",Dummy
610 ENABLE INTR Scode;2
620 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN1"
630 Wait_open:GOTO Wait_open
640 Open_end:!
650 !
660 !**** SHORT STD. MEASUREMENT ****
670 ON INTR Scode GOTO Short_end
680 OUTPUT @Hp4286;"*CLS;*OPC?"
690 ENTER @Hp4286;0pc
700 INPUT "MAKE SHORT, THEN PRESS [RETURN]",Dummy
710 ENABLE INTR Scode;2
720 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN2"
730 Wait_short:GOTO Wait_short
740 Short_end:!
750 !
760 !**** LOAD STD. MEASUREMENT ****
770 ON INTR Scode GOTO Load_end
780 OUTPUT @Hp4286;"*CLS;*OPC?"
790 ENTER @Hp4286;0pc
800 INPUT "CONNECT LOAD, THEN PRESS [RETURN]",Dummy
810 ENABLE INTR Scode;2
820 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN3"
830 Wait_load:GOTO Wait_load
840 Load_end:!
850 !
860 OUTPUT @Hp4286;"SENS:CORR1:COLL:SAVE"
870 !
880 Skip_cal:!
890 !
900 ! **** Initialize BIN table ****
910 OUTPUT @Hp4286;"CALC:COMP:CLE"
920 OUTPUT @Hp4286;"CALC:COMP:CONT:SEGM 1"
930 OUTPUT @Hp4286;"CALC:COMP:SEGM:EDIT"
940 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
950 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.1E-9"
960 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.9E-9"
970 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
980 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
990 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.2E-9"
1000 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.8E-9"
1010 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1020 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1030 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.3E-9"
1040 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.7E-9"
1050 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1060 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1070 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.4E-9"
1080 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.6E-9"

```

```

1090 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1100 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1110 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.5E-9"
1120 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.5E-9"
1130 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1140 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1150 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.6E-9"
1160 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.4E-9"
1170 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1180 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1190 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.7E-9"
1200 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.3E-9"
1210 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1220 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1230 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.8E-9"
1240 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.2E-9"
1250 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1260 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1270 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.9E-9"
1280 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.1E-9"
1290 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1300 OUTPUT @Hp4286;"CALC:COMP:SAVE"
1310 OUTPUT @Hp4286;"CALC:COMP:STAT ON"
1320 !
1330 !***** Set secondary limit parameter
1340 OUTPUT @Hp4286;"CALC:COMP:SREJ:LOW 5"! Lower Limit
1350 OUTPUT @Hp4286;"CALC:COMP:SREJ:UPP 20"! Upper Limit
1360 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1370 OUTPUT @Hp4286;"CALC:COMP:SREJ:STAT ON"
1380 OUTPUT @Hp4286;"CALC:COMP:SAVE"
1390 !
1400 !**** MEAS DATA DISPLAY OFF *****
1410 OUTPUT @Hp4286;"DISP:TRAC:STAT OFF"
1420 !
1430 ! **** Measurement *****
1435 OUTPUT @Hp4286;"FORM ASC"
1440 OUTPUT @Hp4286;"INIT:CONT OFF"
1450 OUTPUT @Hp4286;"ABOR"
1460 ON INTR Scode GOTO Meas_end
1470 OUTPUT @Hp4286;"STAT:INST:ENAB 1;*SRE 4;"
1480 OUTPUT @Hp4286;"*CLS;*OPC?"
1490 ENTER @Hp4286;0pc
1500 INPUT "CONNECT DUT, THEN PRESS [RETURN]",Dummy
1510 ENABLE INTR Scode;2
1520 OUTPUT @Hp4286;"INIT"
1530 Wait_meas:GOTO Wait_meas
1540 Meas_end:!
1550 !
1560 ! **** Output Measurement and BIN sort results *****
1570 OUTPUT @Hp4286;"DATA? BIN"
1580 ENTER @Hp4286;Bin
1590 OUTPUT @Hp4286;"DATA? RDCDT"
1600 ENTER @Hp4286;A(*)
1610 PRINT "          Ls =";A(2,1);" [H]"
1620 PRINT "          Q =";A(2,2)

```

```
1630 PRINT
1640 PRINT "          BIN =";Bin;" (0=OUT OF BIN, 10=AUX BIN, -1=NO TEST)"
1650 PRINT "CONTACT CHECK =";A(1,1);" (0=FAIL, 1=PASS, -1=NO TEST)"
1660 PRINT
1670 END
```

---

## Example 3: Binary data transfer format

### Measurement Conditions

- **Measurement parameter** :  $L_s$ -Q
- **Measurement frequency** : 100 MHz, 300 MHz
- **OSC level** : 10 mA
- **Contact check** : ON
  - Upper limit : 10  $\Omega$
  - Lower limit : 1  $\Omega$
- **Comparater (BIN)**
  - Frequency : 100 MHz
  - Parameter : L
    - BIN 1: 10 nH $\pm$ 1%
    - BIN 2 : 10 nH $\pm$ 2%
    - BIN 3 : 10 nH $\pm$ 5%
    - BIN 4 : 10 nH $\pm$ 10%
    - BIN 5 : 10 nH $\pm$ 20%
  - Secondary parameter : Q
    - Upper limit : 20
    - Lower limit : 5
- **Display** : OFF
- **Calibration** : OPEN/SHORT/LOAD cal using working standard set
- **Output** : Measurement value, BIN sort result, contact check result using binary transfer format

### Sample Program List 3

```
10 !**** INITIALIZE ****
20 DIM A(1:3,1:2),Result_ls(1:2),Result_q(1:2)
30 !
40 Osc_level=.01      ![A] Osc Level
50 Meas_freq_1=1.E+8  ![Hz] Test Frequency 1
60 Meas_freq_2=3.E+8  ![Hz] Test Frequency 2
70 !
80 ASSIGN @Hp4286 TO 800
90 Scode=8
100 CLEAR @Hp4286
110 ABORT Scode
120 OUTPUT @Hp4286;"DISP:ALL HIHB"
130 !
140 ! **** Set measurment parameter ****
150 OUTPUT @Hp4286;"CALC:FORM LSQ"
160 ! **** Sweep Setup ****
170 OUTPUT @Hp4286;"SENS:LIST:CLE"
180 OUTPUT @Hp4286;"SENS:LIST:SEGM:EDIT"
190 OUTPUT @Hp4286;"SENS:LIST:SEGM:FREQ";Meas_freq_1
200 OUTPUT @Hp4286;"SENS:LIST:SEGM:SAVE"
210 OUTPUT @Hp4286;"SENS:LIST:SEGM:ADD"
220 OUTPUT @Hp4286;"SENS:LIST:SEGM:FREQ";Meas_freq_2
230 OUTPUT @Hp4286;"SENS:LIST:SEGM:SAVE"
240 OUTPUT @Hp4286;"SENS:LIST:SAVE"
250 !
260 ! ***** Set OSC Level *****
270 OUTPUT @Hp4286;"SOUR:CURR";Osc_level
280 !
290 ! **** Setup Contact Check ****
300 OUTPUT @Hp4286;"CALC:RDC:LIM:UPP 10"!Upper Limit
310 OUTPUT @Hp4286;"CALC:RDC:LIM:LOW 1"!Lower Limit
320 OUTPUT @Hp4286;"CALC:RDC:LIM:STAT ON"
330 OUTPUT @Hp4286;"SENS:RDC ON"
340 !
350 !**** CALIBRATION STATE CHEKING ****
360 OUTPUT @Hp4286;"SENS:CORR1?"
370 ENTER @Hp4286;Stat
380 IF Stat THEN
390   INPUT "ARE YOU SURE TO RE-CALIBRATE NOW? [Y/N]",Ans$
400   IF Ans$"Y" THEN Skip_cal
410 END IF
420 !
430 !**** SELECT USER CAL *****
440 OUTPUT @Hp4286;"SENS:CORR1:COLL:FPO USER"
450 !**** USER CAL KIT SETTING *****
460 OUTPUT @Hp4286;"SENS:CORR:CKIT UDEF"
470 INPUT "INPUT Rs OF WORKING STANDARD",Rs
480 OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:R ";Rs
490 INPUT "INPUT Ls OF WORKING STANDARD",Ls
500 OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:L ";Ls
510 !
520 !**** SETTING ENABLE REGISTER FOR SRQ ****
530 OUTPUT @Hp4286;"STAT:INST:ENAB 256"
```

```

540 OUTPUT @Hp4286;"*SRE 4"
550 OUTPUT @Hp4286;"TRIG:SOUR INT"
560 !
570 !**** OPEN STD. MEASUREMENT ****
580 ON INTR Scode GOTO Open_end
590 OUTPUT @Hp4286;"*CLS;*OPC?"
600 ENTER @Hp4286;Opc
610 INPUT "CONNECT OPEN, THEN PRESS [RETURN]",Dummy
620 ENABLE INTR Scode;2
630 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN1"
640 Wait_open:GOTO Wait_open
650 Open_end:!
660 !
670 !**** SHORT STD. MEASUREMENT ****
680 ON INTR Scode GOTO Short_end
690 OUTPUT @Hp4286;"*CLS;*OPC?"
700 ENTER @Hp4286;Opc
710 INPUT "CONNECT SHORT, THEN PRESS [RETURN]",Dummy
720 ENABLE INTR Scode;2
730 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN2"
740 Wait_short:GOTO Wait_short
750 Short_end:!
760 !
770 !**** LOAD STD. MEASUREMENT ****
780 ON INTR Scode GOTO Load_end
790 OUTPUT @Hp4286;"*CLS;*OPC?"
800 ENTER @Hp4286;Opc
810 INPUT "CONNECT LOAD, THEN PRESS [RETURN]",Dummy
820 ENABLE INTR Scode;2
830 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN3"
840 Wait_load:GOTO Wait_load
850 Load_end:!
860 !
870 OUTPUT @Hp4286;"SENS:CORR1:COLL:SAVE"
880 !
890 Skip_cal:!
900 !
910 ! **** Initialize BIN table ****
920 OUTPUT @Hp4286;"CALC:COMP:CLE"
930 OUTPUT @Hp4286;"CALC:COMP:CONT:SEGM 1"
940 OUTPUT @Hp4286;"CALC:COMP:SEGM:EDIT"
950 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
960 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.1E-9"
970 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.9E-9"
980 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
990 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1000 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.2E-9"
1010 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.8E-9"
1020 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1030 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1040 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.5E-9"
1050 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.5E-9"
1060 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1070 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1080 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 11.E-9"

```

```

1090 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.E-9"
1100 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1110 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1120 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 12.E-9"
1130 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 8.E-9"
1140 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1150 OUTPUT @Hp4286;"CALC:COMP:SAVE"
1160 OUTPUT @Hp4286;"CALC:COMP:STAT ON"
1170 OUTPUT @Hp4286;"CALC:COMP:CONT:SEGM 1"
1180 !
1190 !***** Set secondary BIN limit parameter
1200 OUTPUT @Hp4286;"CALC:COMP:SREJ:LOW 5"!Lower Limit
1210 OUTPUT @Hp4286;"CALC:COMP:SREJ:UPP 20"!Upper Limit
1220 OUTPUT @Hp4286;"CALC:COMP:SREJ:STAT ON"
1230 !
1240 !***** Set limt parameter
1250 OUTPUT @Hp4286;"CALC:LIM1:CLE"
1260 OUTPUT @Hp4286;"CALC:LIM1:SEGM:EDIT"
1270 OUTPUT @Hp4286;"CALC:LIM1:SEGM:CONT ";Meas_freq_2
1280 OUTPUT @Hp4286;"CALC:LIM1:SEGM:UPP 10.2E-9"
1290 OUTPUT @Hp4286;"CALC:LIM1:SEGM:LOW 9.8E-9"
1300 OUTPUT @Hp4286;"CALC:LIM1:SEGM:SAVE"
1310 OUTPUT @Hp4286;"CALC:LIM1:SAVE"
1320 OUTPUT @Hp4286;"CALC:LIM1:STAT ON"
1330 !
1340 !***** Set secondary limt parameter
1350 OUTPUT @Hp4286;"CALC:LIM2:CLE"
1360 OUTPUT @Hp4286;"CALC:LIM2:SEGM:EDIT"
1370 OUTPUT @Hp4286;"CALC:LIM2:SEGM:CONT ";Meas_freq_2
1380 OUTPUT @Hp4286;"CALC:LIM2:SEGM:LOW 5"
1390 OUTPUT @Hp4286;"CALC:LIM2:SEGM:UPP 50"
1400 OUTPUT @Hp4286;"CALC:LIM2:SEGM:SAVE"
1410 OUTPUT @Hp4286;"CALC:LIM2:SAVE"
1420 OUTPUT @Hp4286;"CALC:LIM2:STAT ON"
1430 !
1440 ! **** Meas Result Display Off
1450 OUTPUT @Hp4286;"DISP:TRAC:STAT OFF"
1460 !
1470 ! **** Binary Format Transfer Setting
1480 ASSIGN @Dt TO 800;FORMAT OFF
1490 OUTPUT @Hp4286;"FORMAT:DATA REAL,64"
1500 INTEGER Head1,Head2,Head3,Head4,Term
1510 !
1520 ! **** Measurement ****
1530 OUTPUT @Hp4286;"INIT:CONT OFF"
1540 OUTPUT @Hp4286;"ABOR"
1550 ON INTR Scode GOTO Meas_end
1560 OUTPUT @Hp4286;"STAT:INST:ENAB 1;*SRE 4;"
1570 OUTPUT @Hp4286;"*CLS;*OPC?"
1580 ENTER @Hp4286;0pc
1590 INPUT "CONNECT DUT, THEN PRESS [RETURN]",Dummy
1600 ENABLE INTR Scode;2
1610 OUTPUT @Hp4286;"INIT"
1620 Wait_meas:GOTO Wait_meas
1630 Meas_end:!
```

```

1640 !
1650 ! **** Output Measurement and BIN sort results ****
1660 OUTPUT @Hp4286;"DATA? BIN"
1670 ENTER @Dt;Head1,Head2,Head3,Head4,Bin,Term
1680 OUTPUT @Hp4286;"DATA? RDCDT"
1690 ENTER @Dt;Head1,Head2,Head3,Head4,A(*),Term
1700 OUTPUT @Hp4286;"DATA? LRES1"
1710 ENTER @Dt;Head1,Head2,Head3,Head4,Result_ls(*),Term
1720 OUTPUT @Hp4286;"DATA? LRES2"
1730 ENTER @Dt;Head1,Head2,Head3,Head4,Result_q(*),Term
1740 PRINT "CONTACT CHECK =" ;A(1,1);" (0=FAIL, 1=PASS, -1=NO TEST)"
1750 PRINT "100MHz Ls:" ;A(2,1);"[H]","Q:" ;A(2,2)
1760 PRINT "      BIN:" ;Bin;" (0=OUT OF BIN, 10=AUX BIN, -1=NO TEST)"
1770 PRINT "300MHz Ls:" ;A(3,1);"H","Q:" ;A(3,2)
1780 PRINT "      LIMIT TEST:" ;"Ls:" ;Result_ls(2),"Q:" ;Result_q(2)
1790 PRINT "      (0=FAIL, 1=PASS, -1=NO TEST)"
1800 END

```



---

## Example 4: BIN sort and Limit test at two frequency points

### Measurement Conditions

- **Measurement parameter** : L<sub>s</sub>-Q
- **Measurement frequency** : 100 MHz, 300 MHz
- **OSC level** : 10 mA
- **Contact check** : ON
  - Upper limit : 1 Ω
  - Lower limit : 10 Ω
- **Comparater (BIN)**
  - Frequency : 100 MHz, 300 MHz
  - Parameter : L
    - Frequency = 100 MHz
    - BIN 1: 10 nH±1%
    - BIN 2 : 10 nH±2%
    - BIN 3 : 10 nH±5%
    - BIN 4 : 10 nH±10%
    - BIN 5 : 10 nH±20%
  - Frequency = 300 MHz
  - BIN 1: 10 nH±2%
  - Secondary parameter : Q
    - Frequency = 100 MHz
    - Upper limit : 20
    - Lower limit : 5
  - Frequency = 300 MHz
  - Upper limit : 50
  - Lower limit : 5
- **Display** : OFF
- **Calibration** : OPEN/SHORT/LOAD cal using working standard set
- **Output** : Measurement value, BIN sort result, contact check result

## Sample Program List 4

```
10 !**** INITIALIZE ****
20 DIM A(1:3,1:2),Result_ls(1:2),Result_q(1:2)
30 !
40 Osc_level=.01      ![A] Osc Level
50 Meas_freq_1=1.E+8  ![Hz] Test Frequency 1
60 Meas_freq_2=3.E+8  ![Hz] Test Frequency 2
70 !
80 ASSIGN @Hp4286 TO 800
90 Scode=8
100 CLEAR @Hp4286
110 ABORT Scode
120 OUTPUT @Hp4286;"DISP:ALL HIHB"
130 !
140 ! **** Set measurment parameter ****
150 OUTPUT @Hp4286;"CALC:FORM LSQ"
160 ! **** Sweep Setup ****
170 OUTPUT @Hp4286;"SENS:LIST:CLE"
180 OUTPUT @Hp4286;"SENS:LIST:SEGM:EDIT"
190 OUTPUT @Hp4286;"SENS:LIST:SEGM:FREQ";Meas_freq_1
200 OUTPUT @Hp4286;"SENS:LIST:SEGM:SAVE"
210 OUTPUT @Hp4286;"SENS:LIST:SEGM:ADD"
220 OUTPUT @Hp4286;"SENS:LIST:SEGM:FREQ";Meas_freq_2
230 OUTPUT @Hp4286;"SENS:LIST:SEGM:SAVE"
240 OUTPUT @Hp4286;"SENS:LIST:SAVE"
250 !
260 ! ***** Set OSC Level *****
270 OUTPUT @Hp4286;"SOUR:CURR";Osc_level
280 !
290 ! **** Setup Contact Check ****
300 OUTPUT @Hp4286;"CALC:RDC:LIM:UPP 10"!Upper Limit
310 OUTPUT @Hp4286;"CALC:RDC:LIM:LOW 1"!Lower Limit
320 OUTPUT @Hp4286;"CALC:RDC:LIM:STAT ON"
330 OUTPUT @Hp4286;"SENS:RDC ON"
340 !
350 !**** CALIBRATION STATE CHEKING ****
360 OUTPUT @Hp4286;"SENS:CORR1?"
370 ENTER @Hp4286;Stat
380 IF Stat THEN
390   INPUT "ARE YOU SURE TO RE-CALIBRATE NOW? [Y/N]",Ans$
400   IF Ans$"Y" THEN Skip_cal
410 END IF
420 !
430 !**** SELECT USER CAL ****
440 OUTPUT @Hp4286;"SENS:CORR1:COLL:FPO USER"
450 !**** USER CAL KIT SETTING ****
460 OUTPUT @Hp4286;"SENS:CORR:CKIT UDEF"
470 INPUT "INPUT Rs OF WORKING STANDARD",Rs
480 OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:R ";Rs
490 INPUT "INPUT Ls OF WORKING STANDARD",Ls
500 OUTPUT @Hp4286;"SENS:CORR1:CKIT:STAN3:L ";Ls
510 !
520 !**** SETTING ENABLE REGISTER FOR SRQ ****
530 OUTPUT @Hp4286;"STAT:INST:ENAB 256"
```

```

540 OUTPUT @Hp4286;"*SRE 4"
550 OUTPUT @Hp4286;"TRIG:SOUR INT"
560 !
570 !**** OPEN STD. MEASUREMENT ****
580 ON INTR Scode GOTO Open_end
590 OUTPUT @Hp4286;"*CLS;*OPC?"
600 ENTER @Hp4286;0pc
610 INPUT "CONNECT OPEN, THEN PRESS [RETURN]",Dummy
620 ENABLE INTR Scode;2
630 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN1"
640 Wait_open:GOTO Wait_open
650 Open_end:!
660 !
670 !**** SHORT STD. MEASUREMENT ****
680 ON INTR Scode GOTO Short_end
690 OUTPUT @Hp4286;"*CLS;*OPC?"
700 ENTER @Hp4286;0pc
710 INPUT "CONNECT SHORT, THEN PRESS [RETURN]",Dummy
720 ENABLE INTR Scode;2
730 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN2"
740 Wait_short:GOTO Wait_short
750 Short_end:!
760 !
770 !**** LOAD STD. MEASUREMENT ****
780 ON INTR Scode GOTO Load_end
790 OUTPUT @Hp4286;"*CLS;*OPC?"
800 ENTER @Hp4286;0pc
810 INPUT "CONNECT LOAD, THEN PRESS [RETURN]",Dummy
820 ENABLE INTR Scode;2
830 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN3"
840 Wait_load:GOTO Wait_load
850 Load_end:!
860 !
870 OUTPUT @Hp4286;"SENS:CORR1:COLL:SAVE"
880 !
890 Skip_cal:!
900 !
910 ! **** Initialize BIN table ****
920 OUTPUT @Hp4286;"CALC:COMP:CLE"
930 OUTPUT @Hp4286;"CALC:COMP:CONT:SEGM 1"
940 OUTPUT @Hp4286;"CALC:COMP:SEGM:EDIT"
950 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
960 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.1E-9"
970 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.9E-9"
980 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
990 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1000 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.2E-9"
1010 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.8E-9"
1020 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1030 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1040 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 10.5E-9"
1050 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.5E-9"
1060 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1070 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1080 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 11.E-9"

```

```

1090 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 9.E-9"
1100 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1110 OUTPUT @Hp4286;"CALC:COMP:SEGM:ADD"
1120 OUTPUT @Hp4286;"CALC:COMP:SEGM:UPP 12.E-9"
1130 OUTPUT @Hp4286;"CALC:COMP:SEGM:LOW 8.E-9"
1140 OUTPUT @Hp4286;"CALC:COMP:SEGM:SAVE"
1150 OUTPUT @Hp4286;"CALC:COMP:SAVE"
1160 OUTPUT @Hp4286;"CALC:COMP:STAT ON"
1170 OUTPUT @Hp4286;"CALC:COMP:CONT:SEGM 1"
1180 !
1190 !***** Set secondary BIN limit parameter
1200 OUTPUT @Hp4286;"CALC:COMP:SREJ:LOW 5"!Lower Limit
1210 OUTPUT @Hp4286;"CALC:COMP:SREJ:UPP 20"!Upper Limit
1220 OUTPUT @Hp4286;"CALC:COMP:SREJ:STAT ON"
1230 !
1240 !***** Set limt parameter
1250 OUTPUT @Hp4286;"CALC:LIM1:CLE"
1260 OUTPUT @Hp4286;"CALC:LIM1:SEGM:EDIT"
1270 OUTPUT @Hp4286;"CALC:LIM1:SEGM:CONT ";Meas_freq_2
1280 OUTPUT @Hp4286;"CALC:LIM1:SEGM:UPP 10.2E-9"
1290 OUTPUT @Hp4286;"CALC:LIM1:SEGM:LOW 9.8E-9"
1300 OUTPUT @Hp4286;"CALC:LIM1:SEGM:SAVE"
1310 OUTPUT @Hp4286;"CALC:LIM1:SAVE"
1320 OUTPUT @Hp4286;"CALC:LIM1:STAT ON"
1330 !
1340 !***** Set secondary limt parameter
1350 OUTPUT @Hp4286;"CALC:LIM2:CLE"
1360 OUTPUT @Hp4286;"CALC:LIM2:SEGM:EDIT"
1370 OUTPUT @Hp4286;"CALC:LIM2:SEGM:CONT ";Meas_freq_2
1380 OUTPUT @Hp4286;"CALC:LIM2:SEGM:LOW 5"
1390 OUTPUT @Hp4286;"CALC:LIM2:SEGM:UPP 50"
1400 OUTPUT @Hp4286;"CALC:LIM2:SEGM:SAVE"
1410 OUTPUT @Hp4286;"CALC:LIM2:SAVE"
1420 OUTPUT @Hp4286;"CALC:LIM2:STAT ON"
1430 !
1440 ! **** Meas Result Display Off
1450 OUTPUT @Hp4286;"DISP:TRAC:STAT OFF"
1460 !
1520 ! **** Measurement ****
1525 OUTPUT @Hp4286;"FORM ASC"
1530 OUTPUT @Hp4286;"INIT:CONT OFF"
1540 OUTPUT @Hp4286;"ABOR"
1550 ON INTR Scode GOTO Meas_end
1560 OUTPUT @Hp4286;"STAT:INST:ENAB 1;*SRE 4;"
1570 OUTPUT @Hp4286;"*CLS;*OPC?"
1580 ENTER @Hp4286;0pc
1590 INPUT "CONNECT DUT, THEN PRESS [RETURN]",Dummy
1600 ENABLE INTR Scode;2
1610 OUTPUT @Hp4286;"INIT"
1620 Wait_meas:GOTO Wait_meas
1630 Meas_end:!
1640 !
1650 ! **** Output Measurement and BIN sort results *****
1660 OUTPUT @Hp4286;"DATA? BIN"
1670 ENTER @Hp4286;Bin

```

```
1680 OUTPUT @Hp4286;"DATA? RDCDT"  
1690 ENTER @Hp4286;A(*)  
1700 OUTPUT @Hp4286;"DATA? LRES1"  
1710 ENTER @Hp4286;Result_ls(*)  
1720 OUTPUT @Hp4286;"DATA? LRES2"  
1730 ENTER @Hp4286;Result_q(*)  
1740 PRINT "CONTACT CHECK =";A(1,1);" (0=FAIL, 1=PASS, -1=NO TEST)"  
1750 PRINT "100MHz Ls:";A(2,1);"[H]","Q:";A(2,2)  
1760 PRINT "      BIN:";Bin;" (0=OUT OF BIN, 10=AUX BIN, -1=NO TEST)"  
1770 PRINT "300MHz Ls:";A(3,1);"H","Q:";A(3,2)  
1780 PRINT "      LIMIT TEST:";"Ls:";Result_ls(2),"Q:";Result_q(2)  
1790 PRINT "      (0=FAIL, 1=PASS, -1=NO TEST)"  
1800 END
```

---

## Example 5: Capacitor measurement

### Measurement Conditions

- **Measurement parameter** :  $C_p$ -D
- **Measurement frequency** : 1 M, 3 M, 5 M, 7 M, 10M, 30 M, 50 M, 70 M, 100 M, 300 M 500 M, 700 M, 1000 M
- **OSC level** : 500 mV
- **Averaging factor**: 8
- **Fixture**: 16192A
- **Contact check** : OFF
- **Display** : ON
- **Calibration** : OPEN/SHORT/LOAD/LOW LOSS capacitor
- **Fixture compensation** OPEN/SHORT
- **Output** : Measurement value

## Sample Program List 5

```
10 !**** INITIALIZE ****
20 DIM Test_freq(1:13),A(1:2)
30 !
40 ! ** TEST FREQUENCY SETTINGS **
50 DATA 1,3,5,7,10,30,50,70,100,300,500,700,1000
60 FOR I=1 TO 13
70 READ Test_freq(I)
80 NEXT I
90 !
100 Osc_level=.5          ![V]
110 Avg=8                 !Averaging on point
120 Fixture$="HP16192"    !Fixture
130 !
140 ASSIGN @Hp4286 TO 800
150 Scode=8
160 CLEAR @Hp4286
170 ABORT Scode
180 OUTPUT @Hp4286;"DISP:ALL HIHB"
190 !
200 ! **** Set measurment parameter ****
210 OUTPUT @Hp4286;"CALC:FORM CPD"
220 !
230 ! ***** Set OSC Level *****
240 OUTPUT @Hp4286;"SOUR:VOLT";Osc_level
250 !
260 !**** CALIBRATION STATE CHEKING ****
270 OUTPUT @Hp4286;"SENS:CORR1?"
280 ENTER @Hp4286;Stat
290 IF Stat THEN
300   INPUT "ARE YOU SURE TO RE-CALIBRATE NOW? [Y/N]",Ans$
310   IF Ans$"Y" THEN Skip_cal
320 END IF
330 !
340 !**** CALIBRATION *****
350 OUTPUT @Hp4286;"SENS:CORR1:COLL:FPO FIX"
360 !
370 !**** SETTING ENABLE REGISTER FOR SRQ ****
380 OUTPUT @Hp4286;"STAT:INST:ENAB 256"
390 OUTPUT @Hp4286;"*SRE 4"
400 OUTPUT @Hp4286;"TRIG:SOUR INT"
410 !
420 !**** OPEN STD. MEASUREMENT ****
430 ON INTR Scode GOTO Open_end
440 OUTPUT @Hp4286;"*CLS;*OPC?"
450 ENTER @Hp4286;Opc
460 INPUT "Connect OPEN at APC-7, then press [RETURN]",Dummy
470 ENABLE INTR Scode;2
480 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN1"
490 Wait_open:GOTO Wait_open
500 Open_end:!
510 !
520 !**** SHORT STD. MEASUREMENT ****
```

```

530 ON INTR Scode GOTO Short_end
540 OUTPUT @Hp4286;"*CLS;*OPC?"
550 ENTER @Hp4286;0pc
560 INPUT "Connect SHORT at APC-7, then press [RETURN]",Dummy
570 ENABLE INTR Scode;2
580 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN2"
590 Wait_short:GOTO Wait_short
600 Short_end:!
610 !
620 !**** LOAD STD. MEASUREMENT ****
630 ON INTR Scode GOTO Load_end
640 OUTPUT @Hp4286;"*CLS;*OPC?"
650 ENTER @Hp4286;0pc
660 INPUT "Connect LOAD at APC-7, then press [RETURN]",Dummy
670 ENABLE INTR Scode;2
680 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN3"
690 Wait_load:GOTO Wait_load
700 Load_end:!
710 !
720 !**** LOW LOSS CAPACITOR MEASUREMENT ****
730 ON INTR Scode GOTO Llc_end
740 OUTPUT @Hp4286;"*CLS;*OPC?"
750 ENTER @Hp4286;0pc
760 INPUT "Connect LOW LOSS CAPACITOR at APC-7, then press [RETURN]",Dummy$
770 ENABLE INTR Scode;2
780 OUTPUT @Hp4286;"SENS:CORR1:COLL STAN4"
790 Wait_llc:GOTO Wait_llc
800 Llc_end:!
810 !
820 OUTPUT @Hp4286;"SENS:CORR1:COLL:SAVE"
830 !
840 Skip_cal:!
850 !
860 !**** FIXTURE ELECTRICAL LENGTH *****
870 OUTPUT @Hp4286;"SYST:FIXT ";Fixture$
880 !
890 !**** FIXTURE COMPENSATION *****
900 OUTPUT @Hp4286;"TRIG:SOUR INT"
910 OUTPUT @Hp4286;"STAT:INST:ENAB 256"
920 OUTPUT @Hp4286;"*SRE 4"
930 !
940 OUTPUT @Hp4286;"SENS:CORR2:COLL:FPO FIX"
950 ON INTR Scode GOTO Compen_open_end
960 OUTPUT @Hp4286;"*CLS;*OPC?"
970 ENTER @Hp4286;0pc
980 INPUT "Make OPEN condition at fixture, then [RETURN]",Dummy$
990 ENABLE INTR Scode;2
1000 OUTPUT @Hp4286;"SENS:CORR2:COLL STAN1"
1010 Comp_wait_open:GOTO Comp_wait_open
1020 Compen_open_end:!
1030 !
1040 ON INTR Scode GOTO Comp_short_end
1050 OUTPUT @Hp4286;"*CLS;*OPC?"
1060 ENTER @Hp4286;0pc
1070 INPUT "Make SHORT condition at fixture, then [RETURN]",Dummy$

```



```

1080 ENABLE INTR Scode;2
1090 OUTPUT @Hp4286;"SENS:CORR2:COLL STAN2"
1100 Comp_wait_short:GOTO Comp_wait_short
1110 Comp_short_end:!
1120 OUTPUT @Hp4286;"SENS:CORR2:COLL:SAVE"
1130 !
1140 ! **** FREQUENCY SETTING AND MEASUREMENT ****
1145 OUTPUT @Hp4286;"FORM ASC"
1150 OUTPUT @Hp4286;"INIT:CONT OFF"
1160 OUTPUT @Hp4286;"ABOR"
1170 !
1180 INPUT "Connect DUT at fixture, then press [RETURN]",Dummy
1190 FOR I=1 TO 13
1200 ON INTR Scode GOTO Meas_end
1210 OUTPUT @Hp4286;"SENS:LIST:CLE"
1220 OUTPUT @Hp4286;"STAT:INST:ENAB 1;*SRE 4;"
1230 OUTPUT @Hp4286;"*CLS;*OPC?"
1240 ENTER @Hp4286;0pc
1250 !OUTPUT @Hp4286;"SENS:LIST:SEGM 1"
1260 OUTPUT @Hp4286;"SENS:LIST:SEGM:EDIT"
1270 OUTPUT @Hp4286;"SENS:LIST:SEGM:FREQ ";Test_freq(I);"MAHZ"
1280 OUTPUT @Hp4286;"SENS:LIST:SEGM:AVER:COUN ";Avg
1290 OUTPUT @Hp4286;"SENS:LIST:SEGM:SAVE"
1300 OUTPUT @Hp4286;"SENS:LIST:SAVE"
1310 !
1320 ENABLE INTR Scode;2
1330 OUTPUT @Hp4286;"INIT"
1340 Wait_meas:GOTO Wait_meas
1350 Meas_end:!
1360 !
1370 ! **** MEASUREMENT RESULTS ****
1380 OUTPUT @Hp4286;"DATA? DTR"
1390 ENTER @Hp4286;A(*)
1400 PRINT "TEST FREQ.=";Test_freq(I);"MHZ","Cp=";A(1);"F","D=";A(2)
1410 NEXT I
1420 END

```

# Error Messages

---

This section lists the error messages that are displayed on the meter display or transmitted by the instrument over GPIB. Each error message is accompanied by an explanation, and suggestions are provided to help in solving the problem. Where applicable, references are provided to the related chapter of the appropriate manual. The messages are listed in alphabetical order.

In the explanation of many error commands, section numbers of the IEEE standard 488.2 are included. Refer to them for further information about an error with these IEEE section numbers.

## **POWER FAILED ON *nnn***

Sever error. Contact your nearest Agilent Technologies office. One or more power is failed. *nnn* is one of  $-5$  V,  $-15$  V,  $+5$  V,  $+15$  V,  $+65$  V, and PostRegHot. It shows that which power line is failed. When this error occurs, the system halts so a controller cannot read this error by GPIB.

### **+0 No error**

The error queue is empty. Every error in the queue has been read (SYSTEM:ERROR? query) or the queue was cleared by power-on or the \*CLS command.

## **6 ADDITIONAL STANDARDS NEEDED**

Error-correction coefficients cannot be computed until all the necessary standards have been measured. Execute all OPEN, SHORT, LOAD calibration

(SENSe:CORRection1:COLLect[:ACQuire] {STAN1|STAN2|STAN3}) before press DONE: CAL (SENSe:CORRection1:COLLect:SAVE).

## **7 CALIBRATION REQUIRED**

No valid calibration coefficients were found when you attempted to perform fixture compensation. See *Users Guide* for information on how to perform calibration.

## **8 CALIBRATION ON FIXED POINTS REQUIRED**

COMP POINT [FIXED] (SENSe:CORRection1:COLLect:FP0ints FIXEd—the fixture compensation on fixed points) cannot be selected when the calibration has been performed CAL POINTS [USER] (SENSe:CORRection1:COLLect:FP0ints USER—the fixture compensation on user-defined points). If you need to set COMP POINT [FIXED], perform calibration again with CAL POINTS [FIXED] setting.

## 9 NO CALIBRATION CURRENTLY IN PROGRESS

The **RESUME CAL SEQUENCE** softkey (No GPIB command) is not valid unless a calibration is in progress. Start a new calibration. See “**Cal**” key” in the *Function Reference* .

## 10 CALIBRATION ABORTED

The calibration in progress was terminated due to a change of the stimulus parameter or calibration measurement points. For example,

- Changing **CAL POINT [FIXED]** between **CAL POINT [USER]** (**SENSe:CORRection1:COLLect:FPOints {FIXed|USER}**).

## 11 COMPENSATION REQUIRED

No valid fixture compensation coefficients were found when you attempted to turn fixture compensation ON (**OPEN ON off** –**SENSe:CORRection2:OPEN ON**, **SHORT ON off** –**SENSe:CORRection2:SHORT ON**, **LOAD ON off** –**SENSe:CORRection2:LOAD ON**). See *Users Guide* for information on how to perform compensation.

## 12 NO COMPENSATION CURRENTLY IN PROGRESS

The **RESUME COMP SEQ** softkey (No GPIB command) is not valid unless a fixture compensation is in progress. Start a new calibration. See “**Cal**” key” in the *Function Reference* .

## 13 COMPENSATION ABORTED

The compensation in progress was terminated due to a change of the stimulus parameter or calibration measurement points. For example,

- Changing **COMP POINT [FIXED]** between **COMP POINT [USER]** (**SENSe:CORRection2:COLLect:FPOints {FIXed|USER}**) before pressing **DONE: COMPEN** (**SENSe:CORRection2:COLLect:SAVE**).

## 15 COMPENSATION STD LIST UNDEFINED

(*GPIB only*) You cannot execute **SENSe:CORRection2:CKIT[1]:STANdard{1-3}[:SElect] LIST** when the fixture compensation standard array is not defined.

## 22 PRINTER: not on, not connect, wrong address

The printer does not respond to control. Verify power to the plotter, and check the GPIB connection between the meter and the printer. Ensure that the printer address recognized by the meter matches the GPIB address set on the printer itself.

## 23 PLOTTER: not on, not connect, wrong address

The plotter does not respond to control. Verify power to the plotter, and check the GPIB connection between the meter and the plotter. Ensure that the plotter address recognized by the meter matches the GPIB address set on the plotter itself.

## 25 PLOTTER NOT READY - PINCH WHEELS UP

If you attempt to plot when the plotter’s pinch wheels are up, this message is displayed.

40 **PHASE LOCK LOOP UNLOCKED**

Sever error. Contact your nearest Agilent Technologies office.

46 **TOO MUCH DATA**

*(GPIB only)* Either there is too much binary data to send to the meter when the data transfer format is binary, or the amount of data is greater than the number of points.

47 **NOT ENOUGH DATA**

*(GPIB only)* The amount of data sent to the meter is less than that expected when the data transfer format is binary.

48 **OPTION NOT INSTALLED**

*(GPIB only)* This error occurs when an GPIB command which is optional command is sent and the meter is not installed the option. Please confirm options installed to the meter using \*OPT? command.

56 **TOO MANY SEGMENTS**

The maximum number of segments for the limit table is 10.

66 **CURRENT EDITING SEGMENT SCRATCHED**

The current editing the table of list sweep or the limit line is scratched. It is occur when the operation other than editing the table is executed before terminate editing the table (SENSe:LIST:SAVE, or CALCulate:LIMit:SAVE )

67 **COMMAND IGNORED - SEGMENT NOT DONE YET**

*(GPIB only)* The GPIB command the meter received is ignored, because the segment is editing . Send CALCulate:LIMit:SEGMemt:SAVE (limit segment done) or SENSe:LIST:SEGment:SAVE (segment done) to terminate editing segment.

69 **TOO MANY SEGMENTS OR POINTS**

Frequency sweep list is limited to 10 points.

74 **CAN'T CHANGE- ANOTHER CONTROLLER ON BUS**

The meter cannot assume the mode of system controller until the system controller is removed from the bus or relinquishes the bus.

75 **NO TEST HEAD CONNECTED**

Check the test head connection.

76 **UNKNOWN TEST HEAD CONNECTED**

The test head get wrong. Contact your nearest Agilent Technologies office.

80 **NOT AVAILABLE FOR THIS FIXTURE**

*(GPIB only)* You cannot execute CALCulate:MATH1[:EXPRession]:NAME {DC0|PER} when the SYSTem:FIXTure {NONE|16191|16192|16193|16194} is selected.

104 **SAVE ERROR**

A serious error, for example physically damaged disk surface, is detected on saving a file.

105 **RECALL ERROR: INSTR STATE PRESET**

A serious error, for example corrupted data, is detected on recalling a file, and this forced the meter to be PRESET.

106 **INVALID FILE NAME**

( *GPIB only* ) The parameter *<file\_name>* for MMEMory:DELEte command must have a “\_D” or “\_S” extension for LIF format, or “STA” or “.DTA” for DOS format.

107 **NO STATE/DATA FILES ON DISK**

( *Front-panel key only* ) The RE-SAVE FILE, COPY FILE, PURGE FILE, or (Recall) key pressed, but there are no files with extensions ( “\_D” or “\_S” for LIF format, or “.STA” or “.DTA” for DOS format) on the flexible disk.

108 **CAN'T SAVE GRAPHICS WHEN COPY IN PROGRESS**

If you attempt to save graphics when a print or plot is in progress, this error message is displayed. Wait until print or plot is complete, then save graphics again.

109 **LIF-DOS COPY NOT ALLOWED**

If you try to copy a file between the RAM disk and the flexible disk when the format of the RAM disk is different from the format of the flexible disk, this message is displayed.

110 **NO STATE/DATA FILES ON MEMORY**

( *Front-panel key only* ) The RE-SAVE FILE, COPY FILE, PURGE FILE, or (Recall) key pressed, but there are no files with extensions ( “\_D” or “\_S” for LIF format, or “.STA” or “.DTA” for DOS format) on the RAM disk memory.

111 **DUPLICATE FILE EXTENSION**

The extension name (GRAPHICs [ ] or ASCII DATA [ ] — MMEMory:FNAME:EXTension{1|2}) is already used for other file types. Use other extension name.

119 **FREQUENCY SWEEP ONLY**

Equivalent circuit function is executed in OSC level sweep, DC-I sweep, DC-V sweep. The equivalent circuit function is available in frequency sweep only.

127 **DC BIAS OVERLOAD**

Hardware failure. Do not input external DC BIAS. If this message keeps on being displayed, contact your nearest Agilent Technologies service office.

132 **BACKUP DATA LOST**

Data checksum error on the battery backup memory has occurred. The battery is recharged for approximately 10 minutes after power was turned ON.

133    **INSUFFICIENT MEMORY**

If a lot of tasks is executed at same time, memory might be insufficient for a while. (For example, running HP Instrument BASIC program, printing a screen, and sending or receiving data array by GPIB are required at same time.) Please wait until finishing some tasks then execute the next task.

140    **ON POINT NOT ALLOWED FOR THE CURRENT TRIG**

The trigger event mode cannot be changed to the ON POINT mode because the current trigger source setting does not allow the ON POINT mode. The trigger event ON POINT mode is available for only MANUAL, EXTERNAL, and BUS trigger sources.

148    **INVALID DATE**

The date entered to set the real time clock is invalid. Reenter correct date.

158    **UNIT STRING TOO LONG**

(*GPIB only*) DISPLAY[:WINDow]:TRACe{18-21}:X:UNIT <string> or DISPLAY[:WINDow]:TRACe{18-21}:Y:UNIT <string> commands can send <string> up to 4 characters.

189    **NOT ALLOWED IN SVC MODE**

Dual channel cannot be displayed in the service mode.

190    **CORR. CONST. DATA LOST; DEFAULT DATA IS USED**

This message is displayed when the correction constants EEPROM data is lost and turned on in the service mode. See the *Service Manual* for troubleshooting.

190    **CORR. CONST. DATA LOST; DEFAULT DATA IS USED**

This message is displayed when the correction constants EEPROM data is lost and turned on in the service mode. See the *Service Manual* for troubleshooting.

198    **POWER ON TEST FAILED**

An internal test fails in the power on sequence (the power on self-test fails). Contact your nearest Agilent Technologies office or see the *Service Manual* for troubleshooting.

199    **EEPROM WRITE ERROR**

Data cannot be stored properly into the EEPROM on the A1 CPU, when performing the display background adjustment or updating correction constants in the EEPROM using the adjustment program. See the *Service Manual* for troubleshooting.

203    **EEPROM CHECK SUM ERROR**

An “internal test 1: A1 CPU” fails. The data (Correction Constants and so on) stored in the A1 CPU’s EEPROM are invalid. See the *Service Manual* for troubleshooting.

204    **DSP CHIP TEST FAILED**

An “internal test 1: A1 CPU” fails. The A1 CPU’s DSP (Digital Signal Processor) does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

205 **F-BUS TIMER CHIP TEST FAILED**

An “internal test 1: A1 CPU” fails. The A1 CPU’s F-BUS (Frequency Bus) timer does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

206 **RTC CHIP TEST FAILED**

An “internal test 1: A1 CPU” fails. The A1 CPU’s RTC (Real Time Clock) does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

207 **KEY CHIP TEST FAILED**

An “internal test 1: A1 CPU” fails. The A1 CPU’s front keyboard control chip does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

208 **FDC CHIP TEST FAILED**

An “internal test 1: A1 CPU” fails. The A1 CPU’s FDC (Flexible Disk drive control) ship does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

209 **GPIB CHIP TEST FAILED**

An “internal test 1: A1 CPU” fails. The A1 CPU’s GPIB chip does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

210 **HP-HIL CHIP TEST FAILED**

An “internal test 1: A1 CPU” fails. The A1 CPU’s HP-HIL control chip does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

211 **CPU INTERNAL SRAM R/W ERROR**

An “internal test 2: A1 VOLATILE MEMORY” fails. The A1 CPU’s internal SRAM does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

212 **CPU BACKUP SRAM R/W ERROR**

An “internal test 2: A1 VOLATILE MEMORY” fails. The A1 CPU’s BACKUP SRAM does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

213 **DSP SRAM R/W ERROR**

An “internal test 2: A1 VOLATILE MEMORY” fails. The DSP’s SRAM on the A1 CPU does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

214 **DUAL PORT SRAM R/W ERROR**

An “internal test 2: A1 VOLATILE MEMORY” fails. The DSP’s dual port SRAM on the A1 CPU does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

215 **POST REGULATOR OUTPUT VOLTAGE OUT OF SPEC**

An “internal test 4: A2 POST REGULATOR” fails. A power supply voltage of the A2 post-regulator is out of its limits. See the *Service Manual* for troubleshooting.

216 **GND LEVEL OUT OF SPEC**

An “internal test 4: A2 POST REGULATOR” fails. The voltage of the GND (Ground) at the DC bus node 26 is out of its limits. See the *Service Manual* for troubleshooting.

217 **FAN POWER OUT OF SPEC**

An “internal test 4: A2 POST REGULATOR” fails. The voltage of the fan power supply at the DC bus node 11 is out of its limits. See the *Service Manual* for troubleshooting.

218 **FAILURE FOUND FROM A/D MUX TO A/D CONVERTER**

An “internal test 5: A6 A/D CONVERTER” fails. A trouble is found on the signal path from the A/D multiplexer to A/D converter on the A6 receiver IF. See the *Service Manual* for troubleshooting.

219 **REF OSC TEST FAILED**

An “internal test 6: A5 REFERENCE OSC” fails. The reference oscillator on the A5 synthesizer does not work properly. See the *Service Manual* for troubleshooting.

220 **FRACTIONAL N OSC TEST FAILED**

An “internal test 7: A5 FRACTIONAL N OSC” fails. The fractional N oscillator on the A5 synthesizer does not work properly. See the *Service Manual* for troubleshooting.

221 **STEP OSC TEST FAILED**

An “internal test 8: A5 STEP OSC” fails. The step oscillator on the A5 synthesizer does not work properly. See the *Service Manual* for troubleshooting.

222 **1st LO OSC TEST FAILED**

An “internal test 9: A4A1 1ST LO OSC” fails. The 1st LO OSC (first local oscillator) on the A4A1 1st LO does not work properly. See the *Service Manual* for troubleshooting.

223 **2nd LO OSC TEST FAILED**

An “internal test 10: A3A2 2ND LO” fails. The 2nd LO OSC (second local oscillator) on the A3A2 2nd LO does not work properly. See the *Service Manual* for troubleshooting.

224 **A3 DIVIDER OUTPUT FREQUENCY OUT OF SPEC**

An “internal test 11: A3A1 DIVIDER” fails. The output frequency of the divider circuit on the A3A1 ALC is out of its limits. See the *Service Manual* for troubleshooting.

225 **3rd LO OSC TEST FAILED**

An “internal test 12: A6 3RD LO OSC” fails. The 3rd LO OSC (third local oscillator) on the A6 receiver IF does not work properly. See the *Service Manual* for troubleshooting.



226 **SOURCE OSC TEST FAILED**

An “internal test 13: A3A1 SOURCE OSC” fails. The source oscillator on the A3A1 ALC does not work properly. See the *Service Manual* for troubleshooting.

227 **SAMPLE FREQUENCY OUT OF SPEC**

An “internal test 14: A6 SEQUENCER” fails. The sampling frequency of the sample/hold circuit on the A6 receiver IF is out of its limits.

228 **SOURCE LEVEL TEST FAILED**

An “internal test 15: SOURCE LEVEL” fails. See the *Service Manual* for troubleshooting.

230 **FLOPPY DISK DRIVE FAILURE FOUND**

An “external test 18: DSK DR FAULT ISOL’N” fails. The A53 built-in FDD (flexible disk drive) does not work properly. Replace the A53 FDD with a new one. See the *Service Manual* for troubleshooting.

232 **SOURCE LEVEL TEST FAILED**

An “external test 20: SOURCE LEVEL” fails. See the *Service Manual* for troubleshooting.

233 **OUTPUT ATTENUATOR TEST FAILED**

An “external test 21: OUTPUT ATTENUATOR” fails. See the *Service Manual* for troubleshooting.

234 **TRD LOSS TEST FAILED**

An “external test 22: TRD LOSS” fails. See the *Service Manual* for troubleshooting.

235 **TRD ISOL’N I TO V TEST FAILED**

An “external test 28: TRD ISOL’N I TO V” fails. See the *Service Manual* for troubleshooting.

236 **TRD ISOL’N V TO I TEST FAILED**

An “external test 29: TRD ISOL’N V TO I” fails. See the *Service Manual* for troubleshooting.

239 **FRONT ISOL’N TEST FAILED**

An “external test 25: FRONT ISOL’N” fails. See the *Service Manual* for troubleshooting.

240 **CABLE ISOL’N TEST FAILED**

An “external test 27: ” fails. See the *Service Manual* for troubleshooting.

241 **RECEIVER GAIN TEST FAILED**

An “external test 22: RECEIVER GAIN” fails. See the *Service Manual* for troubleshooting.

242 **RECEIVER GAIN OUT OF SPEC**

An “external test 25: FRONT ISOL’N” fails. A6 receiver IF gain is incorrect. See the *Service Manual* for troubleshooting.

243 **A6 GAIN TEST FAILED**

An “external test 23: A6 GAIN” fails. See the *Service Manual* for troubleshooting.

244 **A6 VI NORMALIZER TEST FAILED**

An “external test 24: A6 VI NORMALIZER” fails. See the *Service Manual* for troubleshooting.

245 **MAX VCXO LEVEL OUT OF SPEC**

Maximum VCXO level is incorrect, in performing an “adjustment test 36: 3RD VCXO LEVEL ADJ” or an “adjustment test 39: SOURCE VCXO LEVEL ADJ”. See the *Service Manual* for troubleshooting.

246 **VCXO TUNING VOLTAGE OUT OF LIMIT**

VCXO tuning voltage is incorrect, in performing an “adjustment test 36: 3RD VCXO LEVEL ADJ” or an “adjustment test 39: SOURCE VCXO LEVEL ADJ”. See the *Service Manual* for troubleshooting.

-100 **Command error**

This is a generic syntax error that the meter cannot detect more specific errors. This code indicates only that a command error, as defined in IEEE 488.2, 11.5.1.1.4, has occurred.

-101 **Invalid character**

A syntax element contains a character that is invalid for that type. For example, a header containing an ampersand (SENS&).

-102 **Syntax error**

An unrecognized command or data type was encountered. For example, a string was received when the meter was not expecting to receive a string.

-103 **Invalid separator**

The parser was expecting a separator and encountered an illegal character. For example, the semicolon was omitted after a program message unit, \*RST:INIT.

-104 **Data type error**

The parser recognized an unallowed data element. For example, numeric or string data was expected but block data was encountered.

-105 **GET not allowed**

A Group Execute Trigger (GET) was received within a program message (see IEEE 488.2, 7.7).

-108 **Parameter not allowed**

More parameters were received than expected for the header. For example, the \*SRE command only accepts one parameter, so receiving \*SRE 4,16 is not allowed.

-109 **Missing parameter**

Fewer parameters were received than required for the header. For example, the \*SRE command requires one parameter, so receiving only \*SRE is not allowed.

**-110 Command header error**

An error was detected in the header. This error message is used when the meter cannot detect the more specific errors described for errors –111 through –119.

**-111 Header separator error**

A character that is not a legal header separator was encountered while parsing the header. For example, no white space followed the header, thus \*SRE4 is an error.

**-112 Program mnemonic too long**

The header contains more than twelve characters (see IEEE 488.2, 7.6.1.4.1).

**-113 Undefined header**

The header is syntactically correct, but it is undefined for the meter. For example, \*XYZ is not defined for the meter.

**-114 Header Suffix out of range**

The value of a numeric suffix attached to a program mnemonic makes the header invalid.

**-120 Numeric data error**

This error, as well as errors –121 through –129, are generated when parsing a data element that appears to be numeric, including the nondecimal numeric types. This particular error message is used if the meter cannot detect a more specific error.

**-121 Invalid character in number**

An invalid character for the data type being parsed was encountered. For example, an alpha character in a decimal numeric or a “9” in octal data.

**-123 Exponent too large**

The magnitude of the exponent was larger than 32000 (see IEEE 488.2, 7.7.2.4.1).

**-124 Too many digits**

The mantissa of a decimal numeric data element contains more than 255 digits excluding leading zeros (see IEEE 488.2, 7.7.2.4.1).

**-128 Numeric data not allowed**

A legal numeric data element was received, but the meter does not accept it in this position for a header.

**-130 Suffix error**

This error, as well as errors –131 through –139, are generated when parsing a suffix. This particular error message is used if the meter cannot detect a more specific error.

**-131 Invalid suffix**

The suffix does not follow the syntax described in IEEE 488.2, 7.7.3.2, or the suffix is inappropriate for the meter.

-134 **Suffix too long**

The suffix contained more than 12 characters (see IEEE 488.2, 7.7.3.4).

-138 **Suffix not allowed**

A suffix was encountered after a numeric element that does not allow suffixes.

-140 **Character data error**

This error, as well as errors –141 through –148, are generated when analyzing the syntax of a character data element. This particular error message is used if the meter cannot detect a more specific error.

-141 **Invalid character data**

Either the character data element contains an invalid character or the particular element received is not valid for the header.

-144 **Character data too long**

The character data element contains more than twelve characters (see IEEE 488.2, 7.7.1.4).

-148 **Character data not allowed**

A legal character data element was encountered where prohibited by the meter.

-150 **String data error**

This error, as well as errors –151 and –158, are generated when analyzing the syntax of a string data element. This particular error message is used if the meter cannot detect a more specific error.

-151 **Invalid string data**

A string data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.5.2). For example, an END message was received before the terminal quote character.

-158 **String data not allowed**

A string data element was encountered but was not allowed by the meter at this point in parsing.

-160 **Block data error**

This error, as well as errors –161 and –168, are generated when analyzing the syntax of a block data element. This particular error message is used if the meter cannot detect a more specific error.

-161 **Invalid block data**

A block data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.6.2). For example, an END message was received before the length was satisfied.

-168 **Block data not allowed**

A legal block data element was encountered but was not allowed by the meter at this point in parsing.

**-200 Execution error**

This is the generic syntax error that the meter cannot detect more specific errors. This code indicates only that an execution error as defined in IEEE 488.2, 11.5.1.1.5 has occurred.

**-210 Trigger error**

A trigger related error occurred. This error message is used when the meter cannot detect the more specific errors described for errors –211 through –219.

**-211 Trigger ignored**

A GET, \*TRG, or triggering signal was received and recognized by the meter but was ignored because of meter timing considerations. For example, the meter was not ready to respond.

**-213 Init ignored**

A request for a measurement initiation was ignored as another measurement was already in progress.

**-220 Parameter error**

Indicates that a program data element related error occurred. This error message is used when the meter cannot detect the more specific errors described for errors –221 through –229.

**-221 Settings conflict**

A legal program data element was parsed but could not be executed due to the current device state (see IEEE 488.2, 6.4.5.3 and 11.5.1.1.5).

**-222 Data out of range**

A legal program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the meter (see IEEE 488.2, 11.5.1.1.5).

**-223 Too much data**

A legal program data element of block, expression, or string type was received that contained more data than the meter could handle due to memory or related device-specific requirements.

**-224 Illegal parameter value**

Used where exact value, from a list of possibilities, was expected.

**-225 Data out of memory**

The meter has insufficient memory to perform the requested operation.

**-230 Data corrupt or stale**

Possibly invalid data. New reading started but not completed since last access.

**-231 Data questionable**

Measurement accuracy is suspect.

**-240 Hardware error**

A legal program command or query could not be executed because of a hardware problem in the meter. Definition of what constitutes a hardware problem is completely device-specific. This error message is used when the meter cannot detect the more specific errors described for errors –241 through –249.

**-241 Hardware missing**

A legal program command or query could not be executed because of missing meter hardware. For example, an option was not installed.

**-250 Mass storage error**

A mass storage error occurred. This error message is used when the meter cannot detect the more specific errors described for errors –251 through –259.

**-256 File name not found**

A legal program command could not be executed because the file name on the device media was not found: for example, an attempt was made to read or copy a nonexistent file.

**-257 File name error**

A legal program command or query could not be executed because the file name on the device media was in error. For example, an attempt was made to copy to a duplicate file name. The definition of what constitutes a file name error is device-specific.

**-280 Program error**

A downloaded program-related execution error occurred. This error message is used when the meter cannot detect the more specific errors described for errors –281 through –289.

**-281 Cannot create program**

An attempt to create a program was unsuccessful. A reason for the failure might include not enough memory.

**-282 Illegal program name**

The name used to reference a program was invalid. For example, redefining an existing program, deleting a nonexistent program, or in general, referencing a nonexistent program.

**-283 Illegal variable name**

An attempt was made to reference a nonexistent variable in a program.

**-284 Program currently running**

Certain operations dealing with programs may be illegal while the program is running. For example, deleting a running program might not be possible.

**-285 Program syntax error**

A syntax error appears in a downloaded program. The syntax used when parsing the downloaded program is device-specific.

-286 **Program runtime error**

A program runtime error of the HP Instrument BASIC has occurred. To get a more specific error information, use the `ERRM$` or `ERRN` command of the HP Instrument BASIC.

-310 **System error**

Some error, termed “system error” by the meter, has occurred.

-311 **Memory error**

An error was detected in the meter’s memory.

-330 **Self-test failed**

A self-test failed. Contact your nearest Agilent Technologies office or see the *Service Manual* for troubleshooting.

-350 **Queue overflow**

A specific code entered into the queue in lieu of the code that caused the error. This code indicates that there is no room in the queue and an error occurred but was not recorded.

-400 **Query errors**

This is the generic query error that the meter cannot detect more specific errors. This code indicates only that a query error as defined in IEEE 488.2, 11.5.1.1.7 and 6.3 has occurred.

-410 **Query INTERRUPTED**

A condition causing an interrupted query error occurred (see IEEE 488.2, 6.3.2.3). For example, a query followed by DAB or GET before a response was completely sent.

-420 **Query UNTERMINATED**

A condition causing an unterminated query error occurred (see IEEE 488.2, 6.3.2.2). For example, the meter was addressed to talk and an incomplete program message was received by the controller.

-430 **Query DEADLOCKED**

A condition causing a deadlocked query error occurred (see IEEE 488.2, 6.3.1.7). For example, both input buffer and output buffer are full and the meter cannot continue.