Errata

Title & Document Type: E1426A Users Manual

Manual Part Number: e1426-97002

Revision Date: January 1999

HP References in this Manual

This manual may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. We have made no changes to this manual copy. The HP XXXX referred to in this document is now the Agilent XXXX. For example, model number HP8648A is now model number Agilent 8648A.

About this Manual

We've added this manual to the Agilent website in an effort to help you support your product. This manual provides the best information we could find. It may be incomplete or contain dated information, and the scan quality may not be ideal. If we find a better copy in the future, we will add it to the Agilent website.

Support for Your Product

Agilent no longer sells or supports this product. You will find any other available product information on the Agilent Test & Measurement website:

www.tm.agilent.com

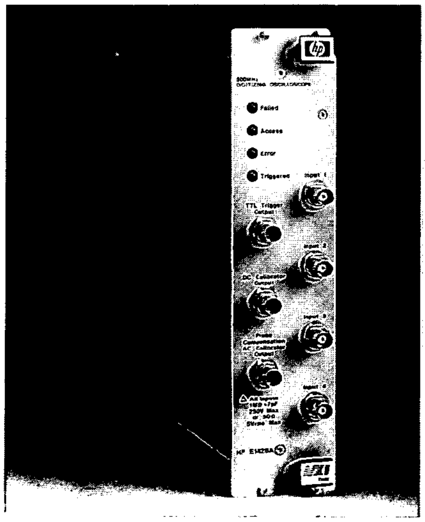
Search for the model number of this product, and the resulting product page will guide you to any available information. Our service centers may be able to perform calibration if no repair parts are needed, but no other support from Agilent is available.

HEWLETT-PACKARD

HP 75000 SERIES C

500 MHz Digitizing Oscilloscope HP E1426A

User's Manual



HP E1426A 500 MHz Digitizing Oscilloscope User's Manual

HP 75000 SERIES C

500 MHz
Digitizing Oscilloscope
HP E1426A

User's Manual

HEWLETT
PACKARD

Manual Part Number E1426-97002

Printed: January 1999
Printed in U.S.A.

CERTIFICATION

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology (formerly National Bureau of Standards), to the extent allowed by that organization's calibration facility, and to the calibration facilities of other International Standards Organization members.

WARRANTY

This Hewlett-Packard product is warranted against defects in materials and workmanship for a period of three years from date of shipment. Duration and conditions of warranty for this product may be superseded when the product is integrated into (becomes a part of) other HP products. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products which prove to be defective.

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

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

LIMITATION OF WARRANTY

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

The design and implementation of any circuit on this product is the sole responsibility of the Buyer. HP does not warrant the Buyer's circuitry or malfunctions of HP products that result from the Buyer's circuitry. In addition, HP does not warrant any damage that occurs as a result of the Buyer's circuit or any defects that result from Buyer-supplied products.

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

EXCLUSIVE REMEDIES

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

NOTICE

The information contained in this document is subject to change without notice. HEWLETT-PACKARD (HP) MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HP shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material. This document contains proprietary information which 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 Hewlett-Packard Company. HP assumes no responsibility for the use or reliability of its software on equipment that is not furnished by HP.

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (b)(3)(ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013. Hewlett-Packard Company, 3000 Hanover Street; Palo Alto, California 94304

Herstellerbescheinigung

Hiermit wird bescheinigt, dass dieses Gerät/System HP E1426A in Übereinstimmung mit den Bestimmungen von Postverfügung 1046/84 funkentstört ist. Der Deutschen Bundespost wurde das Inverkehrbringen dieses Gerätes/Systems angezeigt und die Berechtigung zur Uberprüfung der Serie auf Einhaltung der Bestimmungen eingeräumt.

Zusatzinformation für Mess-und Testgeräte:

Werden Mess- und Testgeräte mit ungeschirmten Kabeln und/oder in offenen Messaufbauten verwendet, so ist vom Betreiber sicherzustellen, dass die Funk-Entstörbestimmungen unter Betriebsbedingungen an seiner Grundstücksgrenze eingehalten werden.

Manufacturer's Declaration

This is to certify that the equipment HP E1426A/E meets the radio frequency interference requirements of Directive FTZ 1046/84. The German Bundespost has been notified that this equipment was put into circulation and has been granted the right to check the product type for compliance with these requirements.

Additional Information for Test and Measurement Equipment:

If test and measurement equipment is operated with unscreened cables and/or used for measurements on open set-ups, the user has to assure that under operating conditions the Radio Interference Limits are still met at the border of the user's premises.

DECLARATION OF CONFORMITY

according to ISO/IEC Guide 22 and EN 45014

Manufacturer's Name: Hewlett-Packard Co.

Manufacturer's Address: Colorado Springs Division
1900 Garden of the Gods Rd.
Colorado Springs, CO 80907 USA

declares that the product

Product Name: Digitizing Oscilloscope Module

Model Number(s): HP E1426A

Product Option(s): All

conforms to the following Product Specifications:

Safety: IEC 1010-1:1990+A1 / EN 61010-1:1993
UL 3111
CSA-C22.2 No. 1010.1:1993

EMC:	CISPR 11:1990 / EN 55011:1991	Group 1, Class A
	IEC 801-2:1991 / EN 50082-1:1992	4 kV CD, 8 kV AD
	IEC 801-3:1984 / EN 50082-1:1992	3 V/m, {1kHz 80% AM, 27-1000 MHz}
	IEC 801-4:1988 / EN 50082-1:1992	0.5 kV Sig. Lines, 1 kV Power Lines

Supplementary Information:

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC, and carries the CE-marking accordingly.

This product was tested in a typical configuration with Hewlett-Packard test systems.

Colorado Springs, 11/17/98

Ken Wyatt

Ken Wyatt / Product Regulations Manager

European Contact: Your local Hewlett-Packard Sales and Service Office or Hewlett-Packard GmbH, Department ZQ / Standards Europe, Herrenberger Strasse 130, D-71034 Boeblingen, Germany (FAX +49-7031-14-3143)

Printing History

The Printing History shown below lists all Editions and Updates of this manual and the printing date(s). The first printing of the manual is Edition 1. The Edition number increments by 1 whenever the manual is revised. Updates, which are issued between Editions, contain replacement pages to correct the current Edition of the manual. Updates are numbered sequentially starting with Update 1. When a new Edition is created, it contains all the Update information for the previous Edition. Each new Edition or Update also includes a revised copy of this printing history page. Many product updates or revisions do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

Edition 1 (Part Number E1426-97000)	January 1993
Edition 1 (Part Number E1426-97002)	January 1999

Safety Symbols

⚠	Instruction manual symbol affixed to product. Indicates that the user must refer to the manual for specific Warning or Caution information to avoid personal injury or damage to the product.
⏚	Indicates the field wiring terminal that must be connected to earth ground before operating the equipment-protects against electrical shock in case of fault.
⊥ or ⊥	Frame or chassis ground terminal-typically connects to the equipment's metal frame.
∼	Alternating current (AC).
⎓	Direct current (DC).
ϟ	Indicates hazardous voltages.
WARNING	Calls attention to a procedure, practice, or condition that could cause bodily injury or death.
CAUTION	Calls attention to a procedure, practice, or condition that could possibly cause damage to equipment or permanent loss of data.

WARNINGS

The following general safety precautions must be observed during all phases of operation, service, and repair of this product. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the product. Hewlett-Packard Company assumes no liability for the customer's failure to comply with these requirements.

Ground the equipment: For Safety Class 1 equipment (equipment having a protective earth terminal), an uninterruptible safety earth ground must be provided from the mains power source to the product input wiring terminals or supplied power cable.

DO NOT operate the product in an explosive atmosphere or in the presence of flammable gases or fumes.

For continued protection against fire, replace the line fuse(s) only with fuse(s) of the same voltage and current rating and type. DO NOT use repaired fuses or short-circuited fuseholders.

Keep away from live circuits: Operating personnel must not remove equipment covers or shields. Procedures involving the removal of covers or shields are for use by service-trained personnel only. Under certain conditions, dangerous voltages may exist even with the equipment switched off. To avoid dangerous electrical shock, DO NOT perform procedures involving cover or shield removal unless you are qualified to do so.

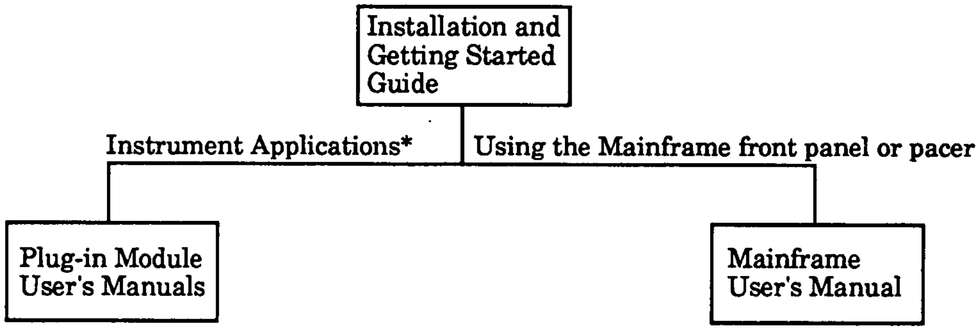
DO NOT operate damaged equipment: Whenever it is possible that the safety protection features built into this product have been impaired, either through physical damage, excessive moisture, or any other reason, REMOVE POWER and do not use the product until safe operation can be verified by service-trained personnel. If necessary, return the product to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

DO NOT service or adjust alone: Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

DO NOT substitute parts or modify equipment: Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the product. Return the product to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

HP 75000 Series C Documentation

Suggested Sequence for Using the Manuals


Installation and Getting Started Guide
Instrument Applications*
Using the Mainframe front panel or pacer
Plug-in Module User's Manuals
Mainframe User's Manual

*For Scanning Voltmeter Applications, refer to the HP E1326A/E1411 5 1/2 digit Multimeter User's Manual

Manual Descriptions

Installation and Getting Started Guide. Contains step-by-step instructions for all aspects of plug-in module and mainframe installation. This guide also contains introductory programming information and examples.

HP E1405 Command Module User's Guide. Contains programming information for the Command Module, operation information (for the HP E1400B mainframe), and general programming information for instruments installed in the mainframe.

Plug-In Module User's Manuals. Contains plug-in module programming and configuration information. These manuals contain examples for the most-used module functions, and a complete SCPI command reference for the plug-in module.

Related Documents

Beginner's Guide to SCPI. Explains the fundamentals of programming instruments with the Standard Commands for Programmable Instruments (SCPI). We recommend this guide to anyone who is programming with SCPI for the first time. Hewlett-Packard part number H2325-90001.

Tutorial Description of the Hewlett-Packard Interface Bus. Describes the technical fundamentals of the Hewlett-Packard Interface Bus (HP-IB). This book also includes general information on IEEE 488.2 Common Commands. We recommend this book to anyone who is programming with IEEE 488.2 for the first time. Hewlett-Packard part number 5952-0156.

IEEE Standard 488.2-1987, IEEE Standard Codes, Formats, Protocols, and Common Commands. Describes the underlying message formats and data types used in SCPI and defines Common Commands. You may find this document useful if you need to know the precise definition of certain message formats, data types, or Common Commands. Available from: The Institute of Electrical and Electronic Engineers, Inc.; 345 East 47th Street; New York, NY 10017; USA

How to Use this Manual

Manual Overview

This manual shows how to operate, configure, and program the HP E1426A 500 MHz Digitizing Oscilloscope Module. Except where noted, the term "Oscilloscope" refers to the E1426A 500 MHz Digitizing Oscilloscope Module.

The Oscilloscope is a VXIbus C-Size message-based slave device, and can operate in a C-size VXIbus mainframe using an HP E1405 Command Module. For other manufacturers' mainframes, refer to the applicable manual supplied by the manufacturer.

Most information in this manual applies to Oscilloscope operations in an HP 75000 Series C System using an HP E1405 Command Module. The Oscilloscope can be programmed using one of the following programming languages.

- Standard Commands for Programmable Instruments (SCPI)
- Hewlett-Packard 54503A Compatible Language (COMP).

Manual Content

This manual has six chapters and three appendices. For operations using COMP, use chapters 1 through 4. For operations using SCPI, use chapters 1, 2, 5, and 6.

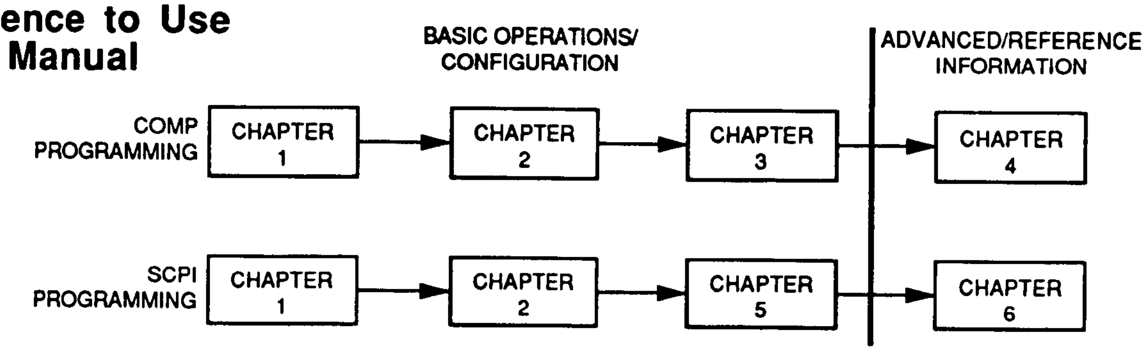
- Chapters 1 and 2 provide Oscilloscope descriptions and configurations.

  Chapter 3 shows several ways to use the Oscilloscope using COMP, and Chapter 4 describes COMP commands.

  Chapter 5 shows several ways to use the Oscilloscope using SCPI, and Chapter 6 describes SCPI commands.

- Appendix A lists the Oscilloscope specifications, Appendix B lists error codes and messages, and Appendix C provides information on optimizing Measurements.

Suggested Sequence to Use This Manual



Contents

5. Using the Oscilloscope with SCPI

6. Oscilloscope SCPI Command Reference

Appendices

A. Oscilloscope Specifications and Characteristics

B. Oscilloscope Error Messages

C. Optimizing Measurements

Programming Example

Index

Contents

Chapter 1 — Getting Started

1

Getting Started

Using This Chapter

This chapter describes the Oscilloscope module, and contains information on how to program it using COMP (Hewlett-Packard 54503A Compatible Language) and SCPI (Standard Commands for Programmable Instruments) commands. This chapter contains the following sections:

Oscilloscope Description

Refer to Figure 1-1 and the following explanation for a description of the HP E1426A 500 MHz Digital Oscilloscope module.

General Description

The Oscilloscope module is a general purpose, four channel, 500 MHz (repetitive bandwidth) oscilloscope, which provides all the versatility and capability of digitizing oscilloscopes. The Oscilloscope module is a VXIbus C-Size message-based product, and can operate in a C-Size VXIbus mainframe using an HP E1405 Command Module.

Four channels accept a wide variety of signals requiring 50Ω or 1MΩ input impedance, ac/dc coupling, bandwidth limiting, and vertical sensitivity from 8mV to 40V full scale.

High precision and advanced triggering enables repeatable and accurate measurements on simple or complex waveforms. Capture of complicated and elusive events on all four inputs is possible using edge, pattern, time qualified pattern, glitch, and state triggering. Available triggering is dependent on the programming language used.

- COMP: Edge, pattern, state, delay, and TV.
- SCPI: Edge triggering only.

Additional features include:

- Single shot bandwidth measurements to 2 MHz
- Four nonvolatile set-up memories
- Four nonvolatile waveform memories
- Pre- and post-trigger viewing capability
- Measurement limit test capability

The oscilloscope also performs an internal self test and calibration for greater confidence in measurement results. Probe compensation, AC calibrator, DC calibrator, and TTL trigger outputs are provided. Four front panel LED's indicate fail, access, error, and trigger status.

Basic Operation

The Oscilloscope module is comprised of a CPU Printed Circuit Assembly (PCA) (HP P/N E1426-69501) and an Acquisition PCA (HP P/N E1426-69502).

The Acquisition PCA attenuates/amplifies each of the four inputs. The conditioned input signal is then routed to a track and hold circuit. The signal is then multiplexed to an A/D Converter where it is changed into a digital word. This digital information is stored for use by the CPU PCA. A replica of the conditioned input signal is also used for triggering. Additional functions include:

- Time base circuit provides the timing signals necessary for data acquisition.
- AC calibrator circuit provides signals for probe compensation, trigger event, and calibration.
- DC calibrator circuit provides a calibration signal.

The CPU PCA contains the control and interface circuits necessary to direct oscilloscope operations. Control information (COMP or SCPI) is received from the mainframe controller, and the necessary instructions are sent to the Acquisition PCA to perform the specific task. When the digital waveform information is received from the Acquisition PCA, all the user requested parameters are measured and routed to the mainframe. Additional functions include:

- TTL and ECL trigger signals from the mainframe are routed to the Acquisition PCA to perform "external trigger" functions.
- TTL and ECL trigger signals from the oscilloscope are routed to the mainframe over the backplane trigger bus lines.
- Trigger circuit provides a TTL Trigger output signal on the front panel.

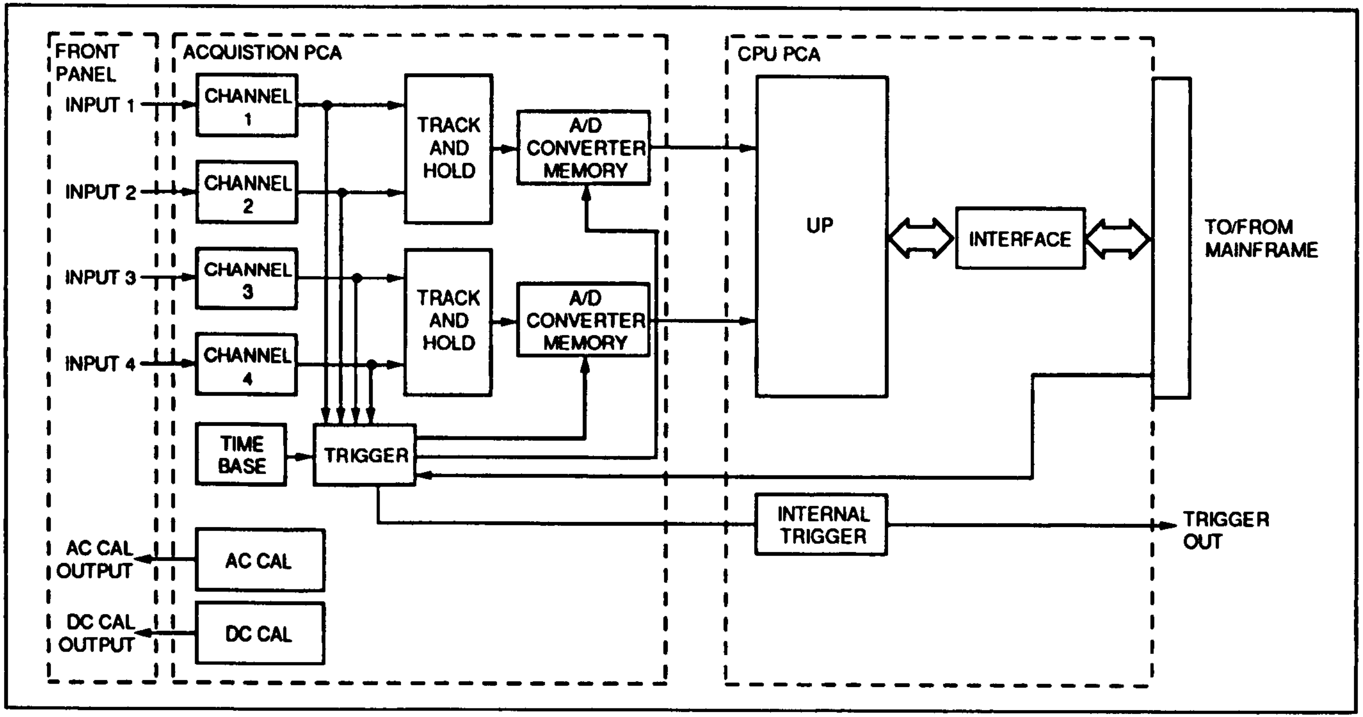


Figure 1-1. Oscilloscope Module Block Diagram

Instrument Definition

HP plug-in modules installed in an HP mainframe or used with an HP command module are treated as independent instruments each having a unique secondary HP-IB address. Each instrument is also assigned a dedicated error queue, input and output buffers, status registers and, if applicable, dedicated mainframe/command module memory space for readings or data. An instrument may be composed of a single plug-in module (such as a counter) or multiple plug-in modules (for a switchbox or scanning voltmeter instrument).

Programming the Oscilloscope Module

To program the Oscilloscope module, you must select the controller language, interface address, and the commands (SCPI or COMP) to be used. See the HP E1405 Command Module User's Guide for interface addressing and controller language information. Guidelines to select COMP commands for the Oscilloscope are provided in Chapter 3. Guidelines to select SCPI commands for the Oscilloscope are provided in Chapter 5.

```
                         XXXXXX  XXXX;"XXXXXX:XXXXXX  XX"
                              |      |               |
Controller Language  _________|      |               |
Interface Address    ________________|               |
SCPI or COMP Command ________________________________|
```

Note

All of the examples in this manual are written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Selecting the Language

The HP E1426A is unique in that it can be operated using two different programming languages.

- The Hewlett-Packard 54503A Compatible Language (COMP) is compatible with the Hewlett-Packard 54503A Oscilloscope Language.
- The Standard Commands for Programmable Instruments (SCPI) is the programming language used in all HP 75000 series VXI products.

When selecting a programming language, it is highly recommended that only one language be used for a given program. Context is not guaranteed if you switch to the other programming language in midstream. Each language has certain advantages as follows:

The advantages of using COMP are:

- Allows use of existing HP 54503A programs.
- All the features of the instrument can be used (e.g. all five triggering modes).
- Allows the HP 54503A to be used as a test program development tool. (If a problem is encountered with a program for the HP E1426A, the HP 54503A can simplify program debugging by adding local operation capabilities and a display.)

The advantages of using SCPI are:

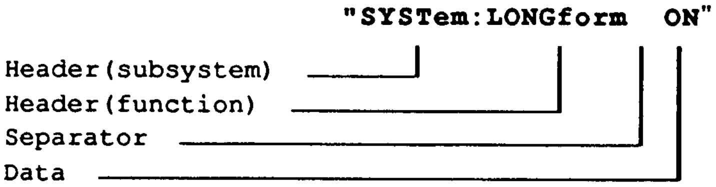
- Shorter learning curve for the user who will be programming numerous SCPI compatible instruments.

Language cross-reference tables are provided in Chapters 4 and 6.

- COMP to HP 54503A Oscilloscope Language – located at the end of Chapter 4.
- COMP to SCPI – located at the end of Chapter 4.
- SCPI to COMP – located at the end of Chapter 6.

Selecting COMP Commands

A COMP command (sometimes called a program message unit) contains the instructions necessary to operate the oscilloscope when the SYSTem COMPatible Language is selected. COMP commands are organized into subsystems, and each subsystem contains the lower level commands necessary to perform a specific task. COMP commands are comprised of headers, separators, and data as shown below.



- Headers specify the subsystem and any lower level function to be performed.
- Separator is one blank space that separates the header from the data.
- Data is used to set a function to a specific value, or a specific state (ON/OFF).

COMP Command Format Used In This Manual

You can send COMP commands in either short or long form. A long form example is:

```
SYSTem:LONGform ON
```

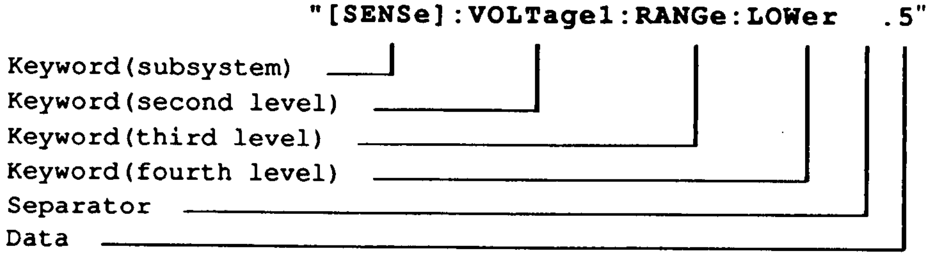
The same command shown without the lower case letters is the short form. The command then becomes:

```
SYST:LONG ON
```

See Chapter 3 for more explanation about COMP commands and how to send them.

Selecting SCPI Commands

A SCPI command contains the instructions necessary to operate the oscilloscope when the SYSTem SCPI Language is selected. SCPI commands are organized into subsystems, and each subsystem contains the lower level commands necessary to perform a specific task. SCPI commands are comprised of keywords, separators, and data as shown below.



- Keywords specify the subsystem and any lower level function to be performed.
- Separator is one blank space that separates the keyword/data.
- Data is used to set a keyword to a specific value, or a specific state (ON/OFF).

SCPI Command Format Used In This Manual

You can send SCPI commands in either short or long form. A long form example is:

```
[SENSe]:VOLTage1:RANGe:LOWer .5
```

The same command shown without the lower case letters is the short form. The command then becomes:

```
[SENS]:VOLT1:RANG:LOW .5
```

Some commands in this manual are shown with brackets ([ ]). These are implied or optional commands that you do not have to execute. For example, the SENSe command is an implied command and is shown in this manual as:

```
[SENS]:VOLT1:RANG:LOW .5
```

Thus, to execute these commands, simply enter:

```
:VOLT1:RANG:LOW .5
```

See Chapter 5 for more explanation about SCPI commands and how to send them.

Contents

Chapter 2 — Configuring the Oscilloscope Module

2

Configuring the Oscilloscope Module

Using This Chapter

This chapter shows how to connect external signals to the Oscilloscope, and how to configure the module for operation. This chapter contains the following sections:

Warnings and Cautions

Warning

SHOCK HAZARD. Only trained service personnel who are aware of the hazards involved should install, remove, or configure the Oscilloscope. Before you remove any installed module, disconnect AC power from the mainframe and from other modules that may be connected to the Oscilloscope.

SAFETY GROUND. When the Oscilloscope is installed in a mainframe, the four front panel module retaining screws must be screwed in completely.

Caution

MAXIMUM INPUT VOLTAGE. The maximum voltage that can be applied to the four input connectors is 5 Vrms at 50Ω or ±250 V (dc+peak ac<10 kHz) at 1MΩ.

STATIC ELECTRICITY. Static electricity is a major cause of component failure. To prevent damage to the electrical components in the Oscilloscope, observe anti-static techniques whenever removing a module from the mainframe or whenever working on a module.

Setting the Logical Address

The Oscilloscope module is selected by specifying its unique logical address. This value cannot be the same as any other module installed in the mainframe, or an error will occur. The logical address factory setting is decimal 40. You may have changed the setting during module installation. Valid address values are from 1 to 255. If the Oscilloscope is used with a HP E1405 Command Module in a C Size Mainframe, refer to the "E1405 Command Module User's Guide" for addressing information. Otherwise, use Figure 2-1 to change the setting.

Note

The address switch selected value must be a multiple of 8 if the Oscilloscope module is used with a VXIbus Command Module.

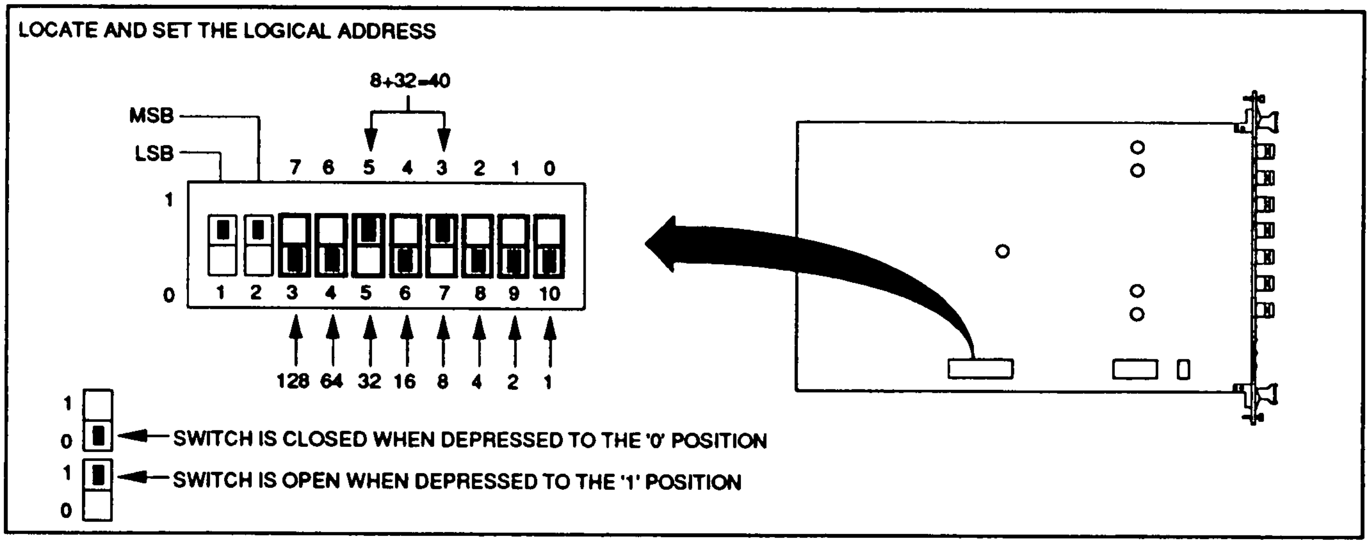

LOCATE AND SET THE LOGICAL ADDRESS
8+32=40
MSB
LSB
7 6 5 4 3 2 1 0
1
0
1 2 3 4 5 6 7 8 9 10
128 64 32 16 8 4 2 1
1
0
SWITCH IS CLOSED WHEN DEPRESSED TO THE '0' POSITION
1
0
SWITCH IS OPEN WHEN DEPRESSED TO THE '1' POSITION

Figure 2-1. Address Selection

Selecting the Bus Request/Grant Level

The Oscilloscope module sends data and addressing information to the command module using a Data Transfer Bus. However, before any information can be sent over the Data Transfer Bus, the Oscilloscope module must request its use. This request is sent over one of four bus request lines (0-3) as selected by the Bus Request/Grant Level switch.

For most applications where the Oscilloscope module is installed in an HP 75000 Series C mainframe, the bus request/grant level setting does not have to be changed from the factory setting of 0. Refer to the HP E1405 Command Module user's guide for more information.

Refer to Figure 2-2 if the bus request/grant level setting must be changed. Each of the four lines has a different priority level. Bus request line 0 has the lowest priority, and line 3 has the highest priority.

LOCATE AND SET THE BUS REQUEST/GRANT LEVEL

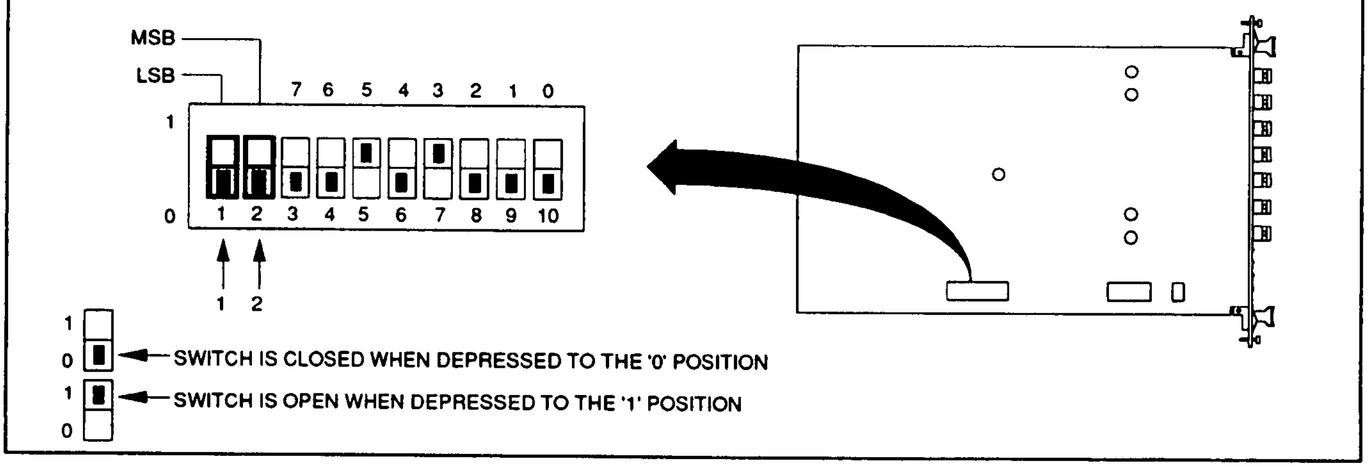

MSB
LSB
7 6 5 4 3 2 1 0
1
0
1 2 3 4 5 6 7 8 9 10
1 2
1
0
1
0
SWITCH IS CLOSED WHEN DEPRESSED TO THE '0' POSITION
SWITCH IS OPEN WHEN DEPRESSED TO THE '1' POSITION

Figure 2-2. Bus Request/Grant Level Selection

Setting the Servant Area Switch

This feature is not currently used in the Oscilloscope. Set all switches to the "0" position as shown in Figure 2-3.

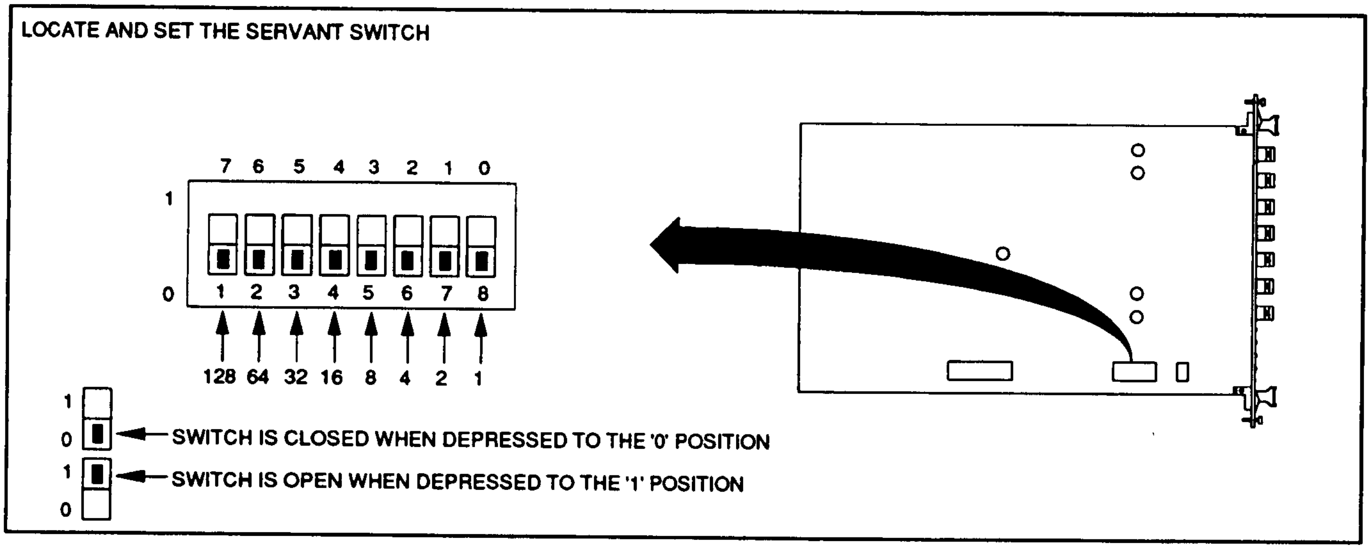

LOCATE AND SET THE SERVANT SWITCH
7 6 5 4 3 2 1 0
1
0
1 2 3 4 5 6 7 8
128 64 32 16 8 4 2 1
1
0
SWITCH IS CLOSED WHEN DEPRESSED TO THE '0' POSITION
1
0
SWITCH IS OPEN WHEN DEPRESSED TO THE '1' POSITION

Figure 2-3. Servant Area Selection

Connecting User Inputs

Cabling Guidelines

Figure 2-4 shows the Oscilloscope module connectors. Use the following guidelines for user connections.

- All connections to the Oscilloscope are made using BNC male connectors.
- Always use shielded coaxial cables with a characteristic impedance of 50Ω. Keep cables as short as possible, especially in high-frequency pulse circuits where a rise/fall time is critical. Long cables can add delay time which can cause timing problems.
- When using probes, make sure that they match the input specifications of the Oscilloscope module. See Appendix A for more information.

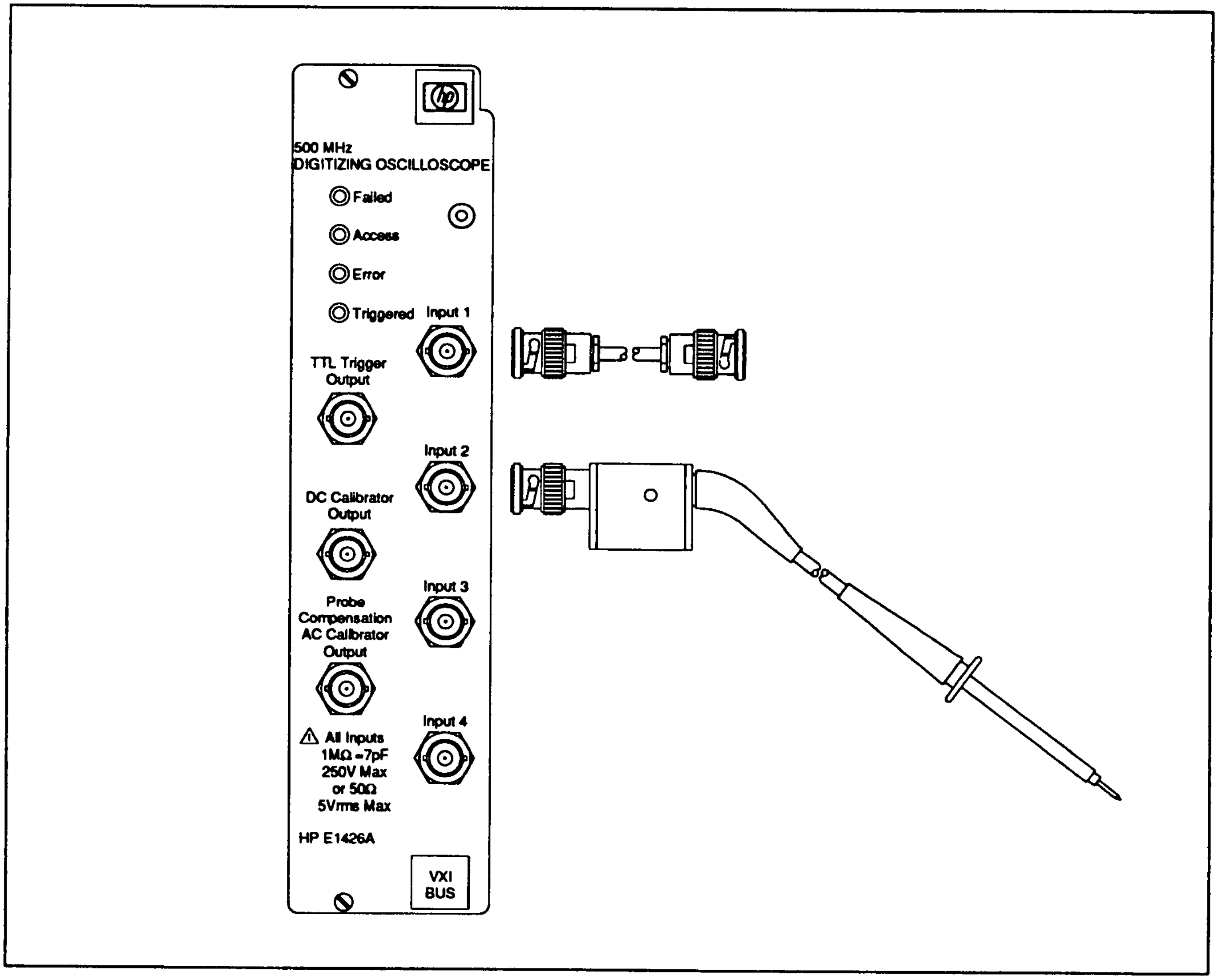


Figure 2-4. Oscilloscope Module Front Panel

Contents

Chapter 3 — Using the Oscilloscope with HP 54503A Compatible Commands

3

Using the Oscilloscope with HP 54503A Compatible Commands

Using This Chapter

This chapter uses typical examples to show how to use the Oscilloscope module using COMP (Hewlett-Packard 54503A Compatible Language) commands. See Chapter 5 for instructions on using SCPI (Standard Commands for Programmable Instruments) commands. This chapter contains the following sections:

Oscilloscope COMP Commands

Table 3-1. Oscilloscope COMP Commands Used in Chapter 3

Command	Description
*CLS	Clear status.
*RST	Reset the Oscilloscope to default state.
ACQuire	
:COMPlete	Specify the data acquisition completion criteria.
:COUNt	Select the number of averages for average mode.
:POINts	Specify the number of data points for data acquisition.
:TYPe	Select the acquisition type.
AUToscale	Perform an autoscale.
CALibrate	
:SCALibration	
:BCALibration	Begin a configured calibration or load default data.
:DCALibration	Configure for a default calibration routine.
:DELay	Configure for a delay calibration routine.
:LTCalibration	Configure for a logic trigger calibration routine.
:TNULl	Configure for a time null calibration routine.
:VERTical	Configure for a vertical calibration routine.
CHANnel<n>	<n> is the channel number 1-4.
:PROBe	Select the input probe attenuation factor.
:RANGe	Set the full scale vertical range.

Table 3-1. Oscilloscope COMP Commands Used in Chapter 3 — Continued

Command	Description
DIGitize	Digitize waveform data.
LTER?	Read the limit test event register.
MEASure	
:ALL?	Measure all the parameters on present signal and return the measurement results.
:COMPare	Configure for a limit test.
:DESTination	Select the destination for limit test violations.
:DUTycycle	Start a continuous duty cycle measurement, but results are NOT returned.
:FALLtime?	Perform a fall time measurement and return the results.
:FREQuency	Start a continuous frequency measurement, but results are NOT returned.
:FREQuency?	Perform a frequency measurement and return the results.
:LIMittest	Start or stop a limit test.
:POSTfailure	Used to stop or continue the limit test after a violation has occurred.
:RESults?	Return current measurement results.
:SCRatch	Clear the measurement queue.
:SOURce	Select the source for all MEASure commands.
:VPP	Start a continuous peak-to-peak voltage measurement, but the results are NOT returned.
SUMMary	
:PRESet	Preset the Oscilloscope questionable enable registers.
:QUEStionable	
:CALibration?	Read the Calibration event register.
SYSTem	
:HEADer	Select system headers to on or off.
:LONGform	Set system headers to long form or short form.
TIMebase	
:RANGe	Specify the full scale horizontal range for the main sweep.
:WINDow	Turn the expanded window on or off.
:DELay	Select the expanded window delay.
:RANGe	Select full scale horizontal range for the expanded window.
TRIGger	
:DELay	Delay the trigger circuit for a specified time or number of events.
:LEVel	Specify the trigger level.
:MODe	Specify the trigger mode (edge, pattern, state, delay, TV).
:OCCurrence	Set the number of trigger events that occur before a sweep is triggered.
:SOURce	Select the source for the OCCurrence command.
:SLOPe	Select a rising or falling slope for the OCCurrence command.
:QUALify	Select a mode (edge, pattern, state) to qualify the trigger before a delay is defined.
:SLOPe	Select the rising or falling edge for the trigger.
:SOURce	Select the source that will produce the trigger.
WAVeform	
:DATA?	Read waveform data from the Oscilloscope.
:FORMat	Select the waveform data format (byte, word, or compressed).
:PREamble?	Read preamble data from the Oscilloscope.
:SOURce	Select the source for all WAVeform commands.

Reset Conditions

When the Oscilloscope is sent a *RST (reset), certain command parameters are set to their default values. Unless these parameters are changed prior to performing the measurement, the reset values will be used.

Table 3-2 lists the reset values for the Oscilloscope module. All parameters not listed will remain in the state last selected.

Table 3-2. *RST (Reset) Conditions and Values

Parameter	Reset	Description
BNC	PROBe	Probe Compensation ON, Trigger Out OFF.
SYSTem:		
HEADer	OFF	Command Headers not returned.
LONGform	OFF	Command Headers abbreviated.
ACQuire:		
COMPlete	100	Acquisition complete when at 100%.
COUNt	8	8 hits per time bucket for completion (will return "1" in NORMal mode)
POINts	500	Acquisition record contains 500 pts.
TYPe	NORMal	Acquisition complete in 1 count.
CHANnel:	1	Channel 1 on, channels 2-4 off.
COUPling	DC	Coupling to DC on all channels.
HFReject	OFF	Internal low pass filter off on all channels.
LFReject	OFF	Internal high pass filter off on all channels.
OFFSet	0	Center screen is 0 V on all channels.
PROBe	1:1	Probe attenuation factor is 1:1 on all channels.
RANGe	4	Full scale vertical range is 4 V.
DISPlay:		
PERSistence	SINGle	Persistence set to minimum.
SOURce	PMEMory0	Acquired data set to pixel memory 0.
FUNCtion:	OFF	FUNCtion 1 and 2 off.
OFFSet	0	Center screen is 0 V for selected function.
MEASure:		
DESTination	OFF	Destination function off.
LIMittest	OFF	Limit test function off.
LOWer	10	Lower measurement threshold to 10%.
MODe	STANdard	Measurement performed using IEEE practice definitions and thresholds.

Table 3-2. *RST (Reset) Conditions and Values — Continued

Parameter	Reset	Description
POSTfailure	STOP	Limit test stopped after violation.
SOURce	CHAN1,CHAN1	Measurement source to channel 1.
STATistics	OFF	Current measurement is returned.
UNITs	PERCent	Threshold units to percent.
UPPer	90	Upper measurement threshold to 90%.
TIMebase:		
DELay	0	Time base delay to 0 seconds.
MODe	AUTo	Time base mode set to auto-trigger.
RANGe	1 ms	Full scale horizontal time to 1 ms.
REFerence	CENTer	Delay reference set to center of sweep.
WINDow:	OFF	Second time base to off.
DELay	0	Second time base delay to 0 seconds.
RANGe	1 ms	Second time base full scale horizontal time to 1 ms.
TRIGger:		
HOLDoff	TIME,40 ns	Holdoff set to 40 ns.
LEVel	0	Trigger level at 0 V.
MODe	EDGe	Edge trigger mode active.
SENSitivity	NORMal	Noise reject off.
SLOPe	POSitive	Positive edge trigger.
SOURce	CHAN1	Channel 1 produces trigger.
WAVeform:		
FORMat	BYTE	Waveform data output to BYTE.
SOURce	CHAN1	Channel 1 source for waveform commands.
OUTPut:	OFF	Output trigger set to off.
TTLTrg	OFF	TTL trigger lines 0-7 set to off.
ECLTrg	OFF	ECL trigger lines 0-1 set to off.
EXTernal	OFF	External trigger set to off.
MEMory:VME:		
ADDRess	200000H	External memory address space (hex).
SIZE	8000H	External memory size in bytes (hex).
STATe	OFF	External memory is disabled.
MEASure:		
ADDRess	200000H	External measurement address space.
STATe	OFF	External measurement address is disabled.

Measurement Sequence

Oscilloscope measurements are most successful if the following measurement sequence is followed.

1. **Determine if a firmware calibration is required** (due to time, operating temperature differences, or measurement accuracy requirements). See Firmware Calibration in this chapter for more information.
2. **Know the signal and type of measurement**. Remember, in most cases you will not have a displayed waveform to view. Have an understanding of the type of signal you want to measure; its amplitude and frequency; is it repetitive? An understanding of the signal you want to measure, and the type of measurement being performed, will help you select the correct oscilloscope and measurement setups. See Measurement Considerations in this chapter for more information.
3. **Set the Oscilloscope controls** (vertical, time base, and trigger). See Oscilloscope Setup in this chapter for more information.
4. **Set the measurement controls** (mode and source). See Measurement Setup in this chapter for more information.
5. **Digitize the Waveform**. Digitizing the waveform ensures that all measurements are performed using the same data, and that the data obtained is valid. The user defines the acquisition and completion criteria for the waveform data. Although this step is not required for some "one time measurements", it's use is highly recommended for all measurements. See Digitizing Waveforms in this chapter for more information.
6. **Perform the measurement.** See Performing a Measurement in this chapter for more information.
7. **Read the results.** See Performing a Measurement in this chapter for more information.

Notes

It is critical that the oscilloscope controls are setup for the specific measurement being performed. Just because measurement results are obtained does not guarantee that they are correct. It is highly recommended that all measurement results be analyzed to verify authenticity. **A returned +9.99999E+37 indicates an invalid measurement.**

See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Firmware Calibration

There are two levels of calibration for the Oscilloscope module. The first level, called vertical, delay, time null, and logic trigger calibrations, can be performed by the operator. Procedures are provided in this section. The second level of calibration should be performed only by qualified service personnel using the service manual.

First level calibration should **ONLY** be performed under the following conditions:

- at six month intervals or every 1000 hours of use,
- if the ambient temperature changes more than 10°C from the temperature at full calibration,
- or to optimize measurement accuracy.

Caution

Do not remove the module with power applied to the mainframe.

The firmware calibration procedures should only be performed after the instrument has run for one hour at ambient temperature installed in the mainframe.

Notes

It is NOT necessary to perform first level calibration procedures prior to every operation.

When performing a first level calibration, **all procedures should be done in the order given.**

After calibrating, you MUST perform an AUToscale or *RST (reset) to return to normal operation.

Vertical Cal Procedure

Vertical calibration is performed on channels one through four (simultaneously) using the following procedure:

1. Set the CAL FACTOR PROTECT switch to **UNPROTECTED**.
2. Connect the Oscilloscope DC Calibrator Output connector to the Input 1-4 connectors.

Note

Verify that the BNC cables are not longer than 1 meter and as close in length as possible.

3. Load the "default" calibration data.
4. Perform clear status, reset, then preset the Oscilloscope.

5. Select and start the vertical calibration routine. Calibration will last for approximately 15 minutes. During calibration, the access and error LED's will be on.

Note

If the calibration time exceeds 15 minutes, the calibration is invalid. Recycle power then repeat procedure. If the condition repeats, the instrument is malfunctioning. Refer to the Service Manual for information on correcting the malfunction before proceeding.

6. After calibration is complete (LED's to off), read the calibration event register to verify that no errors were generated during the procedure.
   - If "0" is returned, the calibration was successful.
   - If "0" is not returned, the calibration attempt was unsuccessful. See Chapter 4, CALibrate Subsystem for more information.
7. Disconnect cables and perform Delay Calibration Procedure.

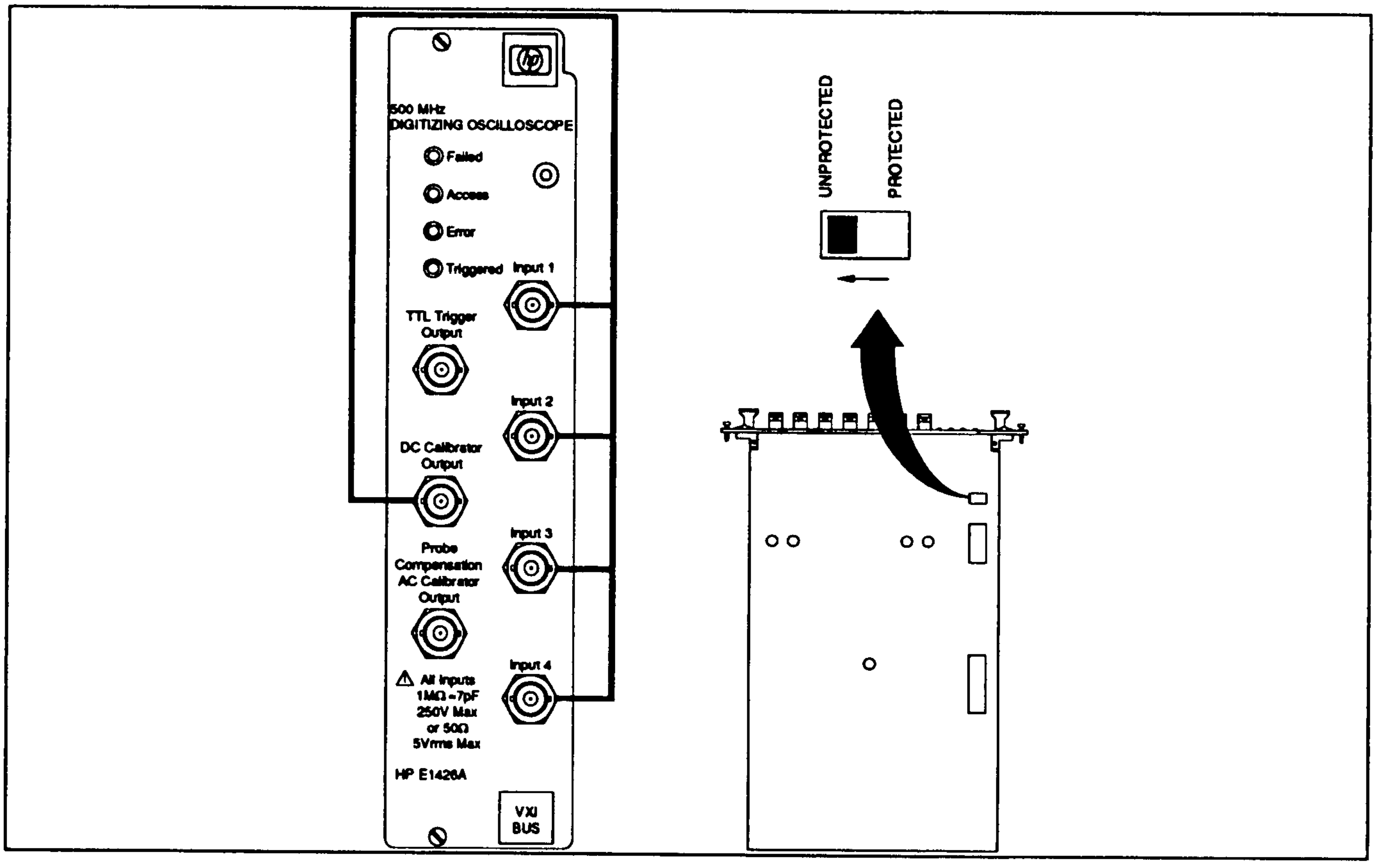


Figure 3-1. Example: Vertical Calibration Setup

Example

The following example shows how to perform a vertical calibration. The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10 OUTPUT 70905;"CAL:SCAL:DCAL"      Selects default calibration routine.
 20 OUTPUT 70905;"CAL:SCAL:BCAL"      Load default calibration data.
 30 OUTPUT 70905;"*CLS"               Clear status.
 40 OUTPUT 70905;"*RST"               Resets the Oscilloscope to its default
                                      state.
 50 OUTPUT 70905;"SUMM:PRES"          Presets the Oscilloscope.
 60 OUTPUT 70905;"CAL:SCAL:VERT"      Selects vertical calibration routine.
 70 OUTPUT 70905;"CAL:SCAL:BCAL"      Starts vertical calibration routine.
 80 OUTPUT 70905;"SUMM:QUES:CAL?"     Read calibration event register.
 90 ENTER 70905;A                     Enter calibration event register results.
100 PRINT A                           Print calibration event register results.
110 END                               Terminate program.
```

Delay Cal Procedure

Delay calibration is performed on channels one through four (one at a time) using the following procedure:

1. Verify the CAL FACTOR PROTECT switch is set to **UNPROTECTED.**
2. Connect the Oscilloscope AC Calibrator Output connector to the Input 1 connector using a BNC cable.

Note

Verify that the BNC cable is not longer than 1 meter.

3. Perform clear status, reset, then preset the Oscilloscope.
4. Select and start the delay calibration routine. During calibration, the access and error LED's will be on.

Note

If the calibration time exceeds 1 minute, the calibration is invalid. Recycle power then repeat procedure. If the condition repeats the instrument is malfunctioning. Refer to the Service Manual for information on correcting the malfunction before proceeding.

5. After calibration is complete (LED's to off), read the calibration event register to verify that no errors were generated during the procedure.

   - If "0" is returned, the calibration was successful.
   - If "0" is not returned, the calibration attempt was unsuccessful. See Chapter 4, CALibrate Subsystem for more information.

6. Disconnect cable from input 1 and reconnect to input 2. Repeat steps 4 and 5 for input 2.

7. Disconnect cable from input 2 and reconnect to input 3. Repeat steps 4 and 5 for input 3.

8. Disconnect cable from input 3 and reconnect to input 4. Repeat steps 4 and 5 for input 4.

9. When complete, disconnect BNC cable and perform Time Null Calibration Procedure.

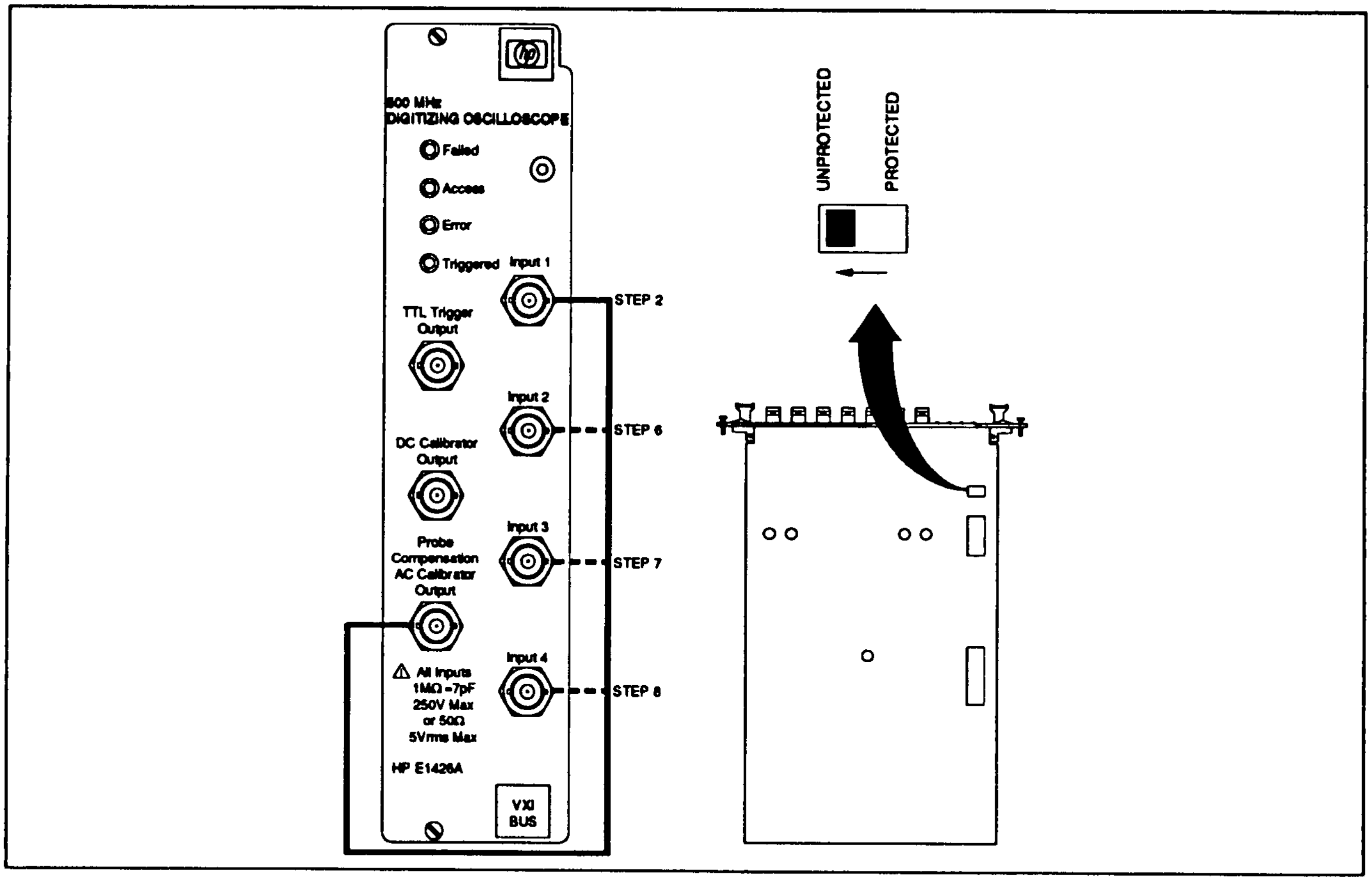


Figure 3-2. Example: Delay Calibration Setup

Example

The following example shows how to perform a delay calibration. The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10  OUTPUT 70905;"*CLS"              Clear status.
 20  OUTPUT 70905;"*RST"              Resets the Oscilloscope to its default
                                      state.
 30  OUTPUT 70905;"SUMM:PRES"         Presets the Oscilloscope.
 40  FOR I=1 TO 4                     Input testing loop.
 50  OUTPUT 70905;"CAL:SCAL:DEL CHAN"&VAL$(I)
                                      Selects input n delay calibration
                                      routine (n=input 1-4).
 60  OUTPUT 70905;"CAL:SCAL:BCAL"     Starts delay calibration routine.
 70  OUTPUT 70905;"SUMM:QUES:CAL?"    Read calibration event register.
 80  ENTER 70905;B                    Enter calibration event register results.
 90  PRINT B                          Print calibration event register results.
100  PAUSE                            Pause to connect next input.
110  NEXT I                           Repeat for inputs 2-4.
120  END                              Terminate program.
```

Time Null Cal Procedure

Time Null calibration is performed on channels one-two, one-three, and one-four (one at a time) using the following procedure:

1. Verify the CAL FACTOR PROTECT switch is set to **UNPROTECTED.**
2. Connect the Oscilloscope AC Calibrator Output connector to both the Input 1 and Input 2 connectors.

Note

Verify that the BNC cables are not longer than 1 meter and equal in length.

3. Perform clear status, reset, then preset the Oscilloscope.
4. Select and start the time null calibration routine. During calibration, the access and error LED's will be on.

Note

If the calibration time exceeds 1 minute, the calibration is invalid. Recycle power then repeat procedure. If the condition repeats the instrument is malfunctioning. Refer to the Service Manual for information on correcting the malfunction before proceeding.

5. After calibration is complete (LED's to off), read the calibration event register to verify that no errors were generated during the procedure.
   - If "0" is returned, the calibration was successful.
   - If "0" is not returned, the calibration attempt was unsuccessful. See Chapter 4, CALibrate Subsystem for more information.
6. Disconnect cable from input 2 and reconnect to input 3. Repeat steps 4 and 5 for input 3.
7. Disconnect cable from input 3 and reconnect to input 4. Repeat steps 4 and 5 for input 4.
8. When complete, disconnect BNC cable and perform the Logic Trigger Calibration Procedure (if necessary).

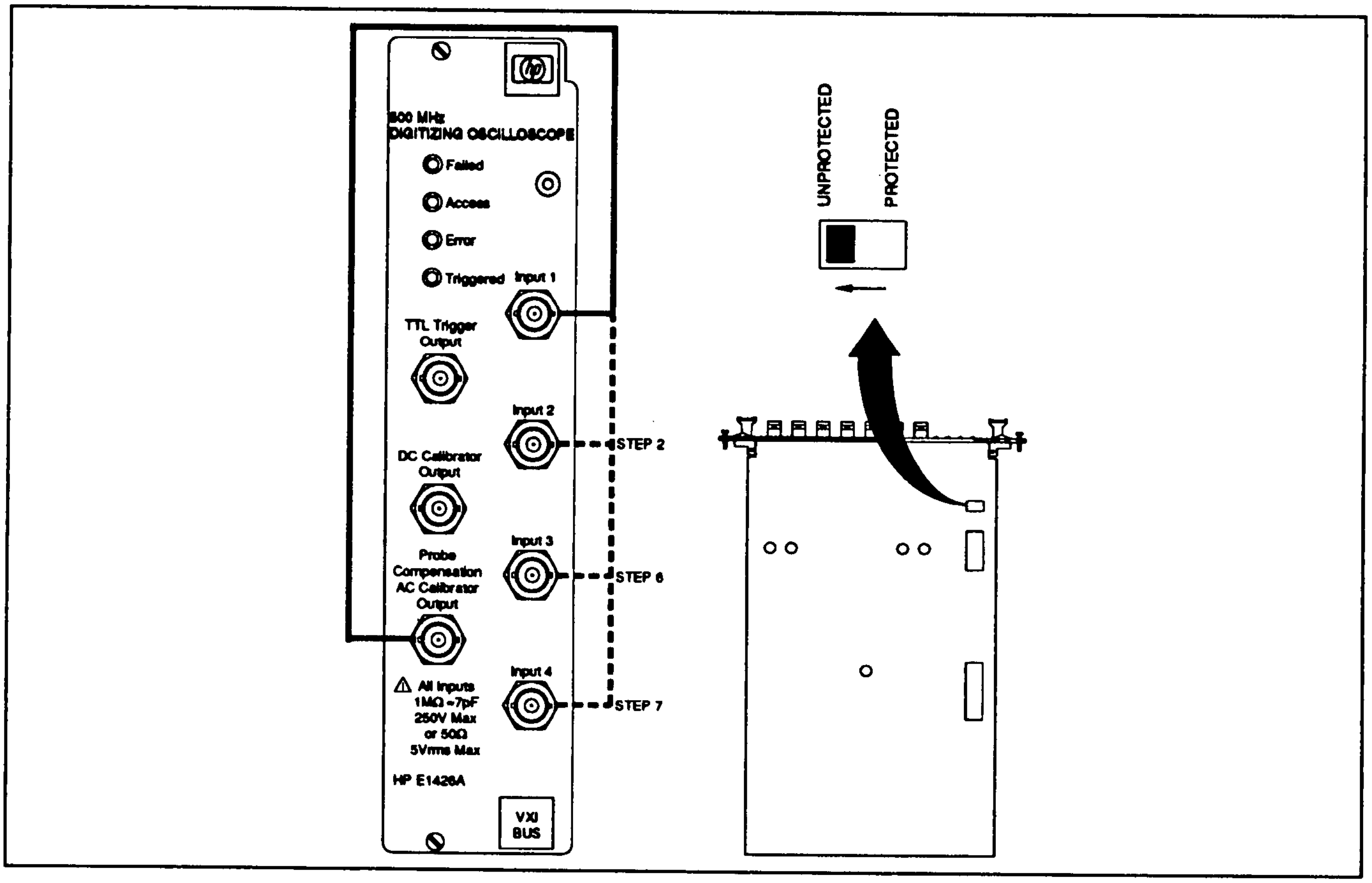


Figure 3-3. Example: Time Null Calibration Setup

Example

The following example shows how to perform a time null calibration. The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10 OUTPUT 70905;"*CLS"              Clear status.
 20 OUTPUT 70905;"*RST"              Resets the Oscilloscope to its default
                                     state.
 30 OUTPUT 70905;"SUMM:PRES"         Presets the Oscilloscope.
 40 FOR I=2 TO 4                     Input testing loop.
 50 OUTPUT 70905;"CAL:SCAL:TNUL CH1TO"&VAL$(I)
                                     Selects channel 1 to n time null
                                     calibration routine (n=input 2-4).
 60 OUTPUT 70905;"CAL:SCAL:BCAL"     Starts time null calibration routine.
 70 OUTPUT 70905;"SUMM:QUES:CAL?"    Read calibration event register.
 80 ENTER 70905;C                    Enter calibration event register results.
 90 PRINT C                          Print calibration event register results.
100 PAUSE                            Pause to connect next input.
110 NEXT I                           Repeat for inputs 3-4.
120 END                              Terminate program.
```

Logic Trigger Cal Procedure

Logic trigger calibration is performed on channel one using the following procedure:

Note

It is **NOT** necessary to perform this procedure unless very accurate delay by time triggering is required during measurements.

If this procedure is not performed, set the CAL FACTOR PROTECT switch to **PROTECTED**.

1. Verify the CAL FACTOR PROTECT switch is set to **UNPROTECTED**.
2. Connect the Oscilloscope AC Calibrator Output connector to the Input 1 connector using a BNC cable.

Note

Verify that the BNC cable is not longer than 1 meter.

3. Read the calibration register to verify that no errors are present.
   - If "0" is returned, proceed with step 5.
   - If "0" is not returned, the Logic Trigger calibration must be terminated, and the cause of the error corrected. See Chapter 4, CALibrate Subsystem for more information.
4. Perform clear status, reset, then preset the Oscilloscope.

5. Locate the logic trigger adjustment on the right side of the Oscilloscope module, and the error and access LED's on the front panel.

6. Select and start the logic trigger calibration routine. Verify that the triggered LED flashes.

   Observe the access and error LED's.

   - If both are on, no adjustment is required.
   - If only one is ON, rotate the logic trigger adjustment until both LED's remain ON.

   Approximately 15 seconds after no further rotation of the adjustment, the triggered LED will flash faster, then all LED's will go out.

7. After calibration is complete (LED's to off), read the calibration register to verify that no errors were generated during the procedure.

   - If "0" is returned, the calibration was successful.
   - If "0" is not returned, the calibration attempt was unsuccessful. See Chapter 4, CALibrate Subsystem for more information.

8. Disconnect BNC cable and set the CAL FACTOR PROTECT switch to **PROTECTED**.

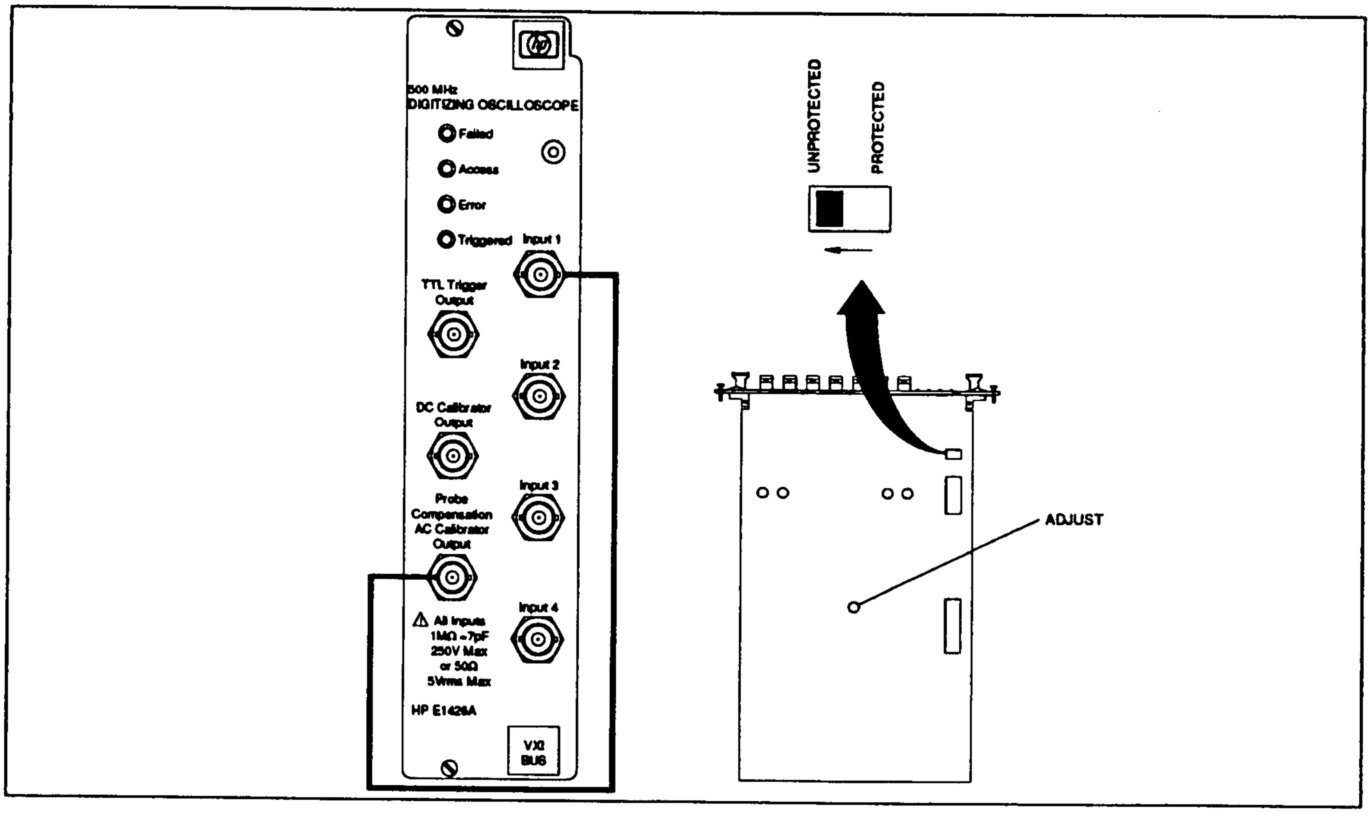


Figure 3-4. Example: Logic Trigger Calibration Setup

Example

The following example shows how to perform a logic trigger calibration. The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

`10 OUTPUT 70905;"*CLS"`	*Clear status.*
`20 OUTPUT 70905;"*RST"`	*Resets the Oscilloscope to its default state.*
`30 OUTPUT 70905;"SUMM:PRES"`	*Presets the Oscilloscope.*
`40 OUTPUT 70905;"SUMM:QUES:CAL?"`	*Read calibration event register.*
`50 ENTER 70905;D`	*Enter calibration event register results.*
`60 PRINT D`	*Print calibration event register results.*
`70 IF D<>0 THEN 130`	*Terminate if results not 0.*
`80 OUTPUT 70905;"CAL:SCAL:LTC"`	*Selects logic trigger calibration routine.*
`90 OUTPUT 70905;"CAL:SCAL:BCAL"`	*Starts logic trigger calibration routine.*
`100 OUTPUT 70905;"SUMM:QUES:CAL?"`	*Read calibration event register.*
`110 ENTER 70905;E`	*Enter calibration event register results.*
`120 PRINT E`	*Print calibration event register results.*
`130 END`	*Terminate program.*

Starting a Measurement

Before configuring the Oscilloscope to perform a specific measurement, it is very beneficial to know both the signal being measured, and the type of measurement being performed.

Unknown Input Signal

If the input signal is unknown, AUToscale can be used with MEASure:ALL? to quickly determine some of the critical input signal parameters.

The following example shows how to perform a measure all on a signal connected to Input 1. The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
5   DIM A$[500]                      String for data.
10  OUTPUT 70905;"*CLS"              Clear status.
20  OUTPUT 70905;"*RST"              Resets the Oscilloscope to its
                                     default state.
30  OUTPUT 70905;"SYST:HEAD ON"      Turn headers on so measurement
                                     results are identified.
40  OUTPUT 70905;"AUT"               Perform autoscale.
50  OUTPUT 70905;"MEAS:ALL?"         Measure input signal.
60  ENTER 70905;A$                   Enter measurement results.
70  PRINT A$                         Print measurement results.
80  END                              Terminate program.
```

Note

Some of the measured data returned will be invalid (+9.99999E+37) because the proper portion of the waveform was not present during measurement.

See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Measurement Considerations

In order to make a specific measurement, the portion of the waveform required for that measurement must be setup and present on the oscilloscope. For example, to measure:

- Period or frequency - a minimum of one complete cycle must be present.
- Pulse width - the entire pulse must be present.
- Rise time - the leading (positive-going) edge of the waveform must be present.
- Fall time - the trailing (negative-going) edge of the waveform must be present.

Performing this function on an oscilloscope with a display is a comparatively simple task. However, when the display is removed, certain steps must be taken to assure the correct Oscilloscope and measurement set-ups are performed prior to the actual measurement.

Oscilloscope Setup

Before a specific measurement can be performed, it is necessary to setup the Oscilloscope vertical, time base, and triggering controls. The settings are dependent on the input signal and the desired measurement being performed. There are two ways to set these controls:

- Automatic (using AUToscale).
- Manual - User enters desired values.

Automatic

When selected, the Oscilloscope automatically evaluates the input signals present at inputs 1-4, and then sets the vertical, time base, and triggering controls to present the signal. You setup the Oscilloscope with the following command:

```
AUToscale
```

Note

Autoscale should only be used with relatively stable input signals having a duty cycle of greater than 0.5% and a frequency greater than 50 Hz.

Manual

Instructions for manual selection of the Vertical, Time base, and Triggering controls are provided in this section.

Vertical Setup

This section discusses the vertical or channel controls you can program with the CHANnel<*n*> command. These controls allow the selection of:

- Input Coupling and Impedance
- Input Filter State
- Input Probe Attenuation
- Input Offset
- Input Range

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL channel commands available, see Chapter 4, CHANnel subsystem.

Input Coupling/Impedance

Coupling and impedance for each channel can be set to AC/1MΩ, DC/1MΩ, or DC/50Ω. DC/1MΩ is selected at reset. You select coupling with the following command:

```
CHANn:COUP xxx     (n=channel number and xxx = AC, DC, or
                   DCF)
```

Input Filter State

Two input filters are selectable to provide low-pass (BW≈30 MHz) or high-pass (BW≈450 Hz) filtering. Both filters are disabled at reset.

You enable the low-pass filter (high frequency reject) with the following command:

```
CHANn:HFR xxx   (n=channel number and xxx = ON or OFF)
```

You enable the high-pass filter (low frequency reject) with the following command:

```
CHANn:LFR xxx   (n=channel number and xxx = ON or OFF)
```

Input Probe Attenuation

Probe attenuation factor for each channel can be entered from 0.9:1 to 1000.0:1 to match the probe currently connected at the input. 1:1 is selected at reset. You enter probe attenuation factor with the following command:

```
CHANn:PROB xxx     (n=channel number and xxx = value :1)
```

Note

Changing probe attenuation from 1:1 will affect current settings of input range and offset.

Input Offset

Offset voltage for each channel can be entered to a level depending on the current RANGe selection. 0 volts is selected at reset. You enter offset with the following command:

```
CHANn:OFFS xxx     (n=channel number and xxx = value in
                   volts)
```

Input Range

Full scale (not per division) vertical axis for each channel can be entered from 8 mV to 40 V. 4 volts is selected at reset. You enter vertical full scale range with the following command:

```
CHANn:RANG xxx     (n=channel number and xxx = value in
                   volts)
```

Time Base Setup

This section discusses the time base or horizontal controls you can program with the TIMebase command. These controls allow the selection of:

- Delay
- Mode
- Range
- Reference

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available time base commands, see Chapter 4, TIMebase subsystem.

Delay

The time base delay (time interval between trigger event and sweep delay reference point) can be set to a value depending on the current RANGe selection. 0 seconds is selected at reset. See Reference in this section for more information. You select time base delay with the following command:

```
TIM:DEL xxx    (xxx = value in seconds + or –)
```

Mode

Three different sweep modes are available. TRIGgered requires a trigger event for a new sweep. SINGle requires a RUN command for a sweep. AUTo sweeps with or without a signal present. AUTo is selected at reset. You select time base mode with the following command:

```
TIM:MOD xxx    (xxx = TRIG, SING, or AUT)
```

Range

Full scale (not per division) horizontal axis can be entered from 2 ns to 50 seconds. 1 ms is selected at reset. You enter horizontal full scale range with the following command:

```
TIM:RANG xxx    (xxx = value in seconds)
```

Reference

Three different settings are available that control the sweep reference point. LEFT, CENTer, or RIGHt sets the reference to the left, center, or right of the sweep, respectively. CENTer is selected at reset. Refer to Delay above for more information. You enter reference with the following command:

```
TIM:REF xxx    (xxx = LEFT, CENT, or RIGH)
```

Trigger Setup

This section discusses the trigger controls you can program with the TRIGger command. These controls allow the selection of:

- Mode
- Holdoff
- Level
- Sensitivity
- Slope
- Source

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available triggering commands, see Chapter 4, TRIGger subsystem.

Mode

Five different trigger modes are available. See Chapter 4, TRIGger:MODe, for information on EDGe, PATTern, STATe, DELay, and TV trigger modes. EDGe is selected at reset. You select trigger mode with the following command:

```
TRIG:MOD xxx     (xxx = EDG, PATT, STAT, DEL, or TV)
```

Holdoff

Holdoff (disabling of trigger circuit for specific duration) can be set to either a time (40 ns to 320 ms) or event (2 to 16 million), and is valid in all modes except DELay. 40 ns is selected at reset. You select holdoff with the following command:

```
TRIG:HOLD  xxx,yyy  (xxx = TIME or EVENt and yyy = value in
                     seconds or events)
```

Level

Active trigger level voltage can be entered to a value dependent on CHANnel<*n*>:RANGe and OFFSet settings as follows:.

±0.75 of selected range from current offset

0 volts is selected at reset. You enter trigger level with the following command:

```
TRIG:LEV xxx     (xxx = value in volts)
```

Sensitivity

Trigger sensitivity or noise rejection can be turned on or off for the selected SOURce. Aids in eliminating false triggering. NORMal is noise reject off and LOW is noise reject on. NORMal is selected at reset. You enter trigger sensitivity with the following command:

```
TRIG:SENS xxx     (xxx = NORM or LOW)
```

Slope

The POSitive (rising) or NEGative (falling) edge of the input signal can be selected as the trigger event for the selected source. POSitive is selected at reset. You select trigger slope with the following command:

```
TRIG:SLOP xxx     (xxx = POS or NEG)
```

Source

The trigger source can be selected from one of the input signals (CHANnel 1-4), or from one of 10 bus lines (TTLTrg0-7 or ECLTrg 0-1). Only one trigger source can be specified at a time. CHANnel 1 is selected at reset. You select trigger source with the following command:

```
TRIG:SOUR  xxx      (xxx = CHAN1, CHAN2, CHAN3, CHAN4,
                     TTLT0, TTLT1, TTLT2, TTLT3, TTLT4,
                     TTLT5, TTLT6, TTLT7, ECLT0 or ECLT1)
```

Measurement Setup

After the vertical, time base, and triggering controls are setup, it is necessary to set any unique measurement parameters prior to performing the actual measurement. These controls allow the selection of:

- Mode
- Source

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available measurement commands, see Chapter 4, MEASure subsystem.

Mode

Allows the user to specify the measurement definitions and thresholds. STANdard selects IEEE measurement practice definitions and thresholds. USER allows the user to specify the definition and thresholds using the DEFine, LOWer, UPPer, and UNITs commands.

User definitions (MEASure:DEFine) are dependent on measurement type as follows (underlined parameter is the STANdard or default):

Delay	edge polarity (positive or negative) edge number (1 to 100) 1 for start, 2 for stop edge level (lower, mid, upper)
+ Pulse width	threshold (lower, mid, upper)
– Pulse width	threshold (lower, mid, upper)

User thresholds (MEASure:LOWer, UPPer, and UNITs) are as follows (underlined parameter is the STANdard or default):

Units	% or volts
Upper threshold	±250.0 kv/-25.00% to +125.0% (+90%)
Lower threshold	±250.0 kv/-25.00% to +125.0% (+10%)

See Chapter 4, MEASure:DEFine, LOWer, UPPer, and UNITs for additional information on user defined definitions and thresholds. STANdard is selected at reset. You setup the measurement mode with the following command:

```
MEAS:MODE xxx  (xxx = STAN or USER)
```

Source

Used to select the source(s) for measurement. Two sources can be specified, however, all measurements except DELay are made on the first source. Source can be CHANnels (inputs1-4), FUNCtions, or WMEMorys (waveform memories 1-4). CHAN1,CHAN1 is selected at turn-on or reset. You set up the measurement source(s) with the following command:

```
MEAS:SOUR xxx,xxx  (xxx = CHAN1-4, FUNC 1-2, or WMEM 1-4)
```

Digitizing Waveforms

Waveforms can be digitized to provide a waveform that fulfills user defined acquisition and completion criteria. The resulting waveform can be measured by the Oscilloscope or passed to the controller as a numerical representation. In addition, preamble data is passed to the controller so the digitized data can be interpreted. The user can specify exactly what the digitized information will contain, how the waveform is acquired, when the waveform is completed, and the format of the digitized information. Use the following sequence to assist in digitizing waveforms:

1. **Waveform Acquisition:** Defines the data type, completion criteria, number of averages, and number of data points for data acquisition. See Acquisition in this section for more information.
2. **Digitize the Waveform**: Acquires data on the specified channel, stores the data in the channel buffer, then stops the acquisition. See Digitize in this chapter for more information.
3. **Measure the Waveform**: All measurements made by the Oscilloscope are now performed on the same data. See Performing a Measurement in this chapter for more information.
4. **Waveform Disposition:** When the data is going to/from the bus, it defines where to get the data, the format the data is to be in, reads the digitized waveform data, and reads the preamble data. See Disposition in this section for more information.

Acquisition

This section discusses the acquisition controls you can program with the ACQuire command. These controls allow the selection of:

- Type
- Completion Criteria
- Count
- Points

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available acquisition commands, see Chapter 4, ACQuire subsystem.

Type

Three different modes define the type of acquisition that will take place when a DIGitize command is executed. See Chapter 4, ACQuire:TYPe, for information on NORMal, AVERage, and ENVelope modes. NORMal is selected at reset. You select acquisition type with the following command:

```
ACQ:TYP xxx     (xxx = NORM, AVER, ENV)
```

Completion Criteria

Completion criteria for an acquisition can be entered from 0 to 100%. 100% is selected at reset. You enter completion criteria with the following command:

```
ACQ:COMP xxx    (xxx = 0 to 100)
```

Count

The number of points to be averaged for each acquisition can be entered from 1 to 2048. 8 is selected at reset. You enter counts with the following command:

```
ACQ:COUN xxx    (xxx = 1 to 2048)
```

Points

The number of points for each acquisition record can be entered from 32 to 1024. 500 is selected at reset. You enter points with the following command:

```
ACQ:POIN xxx    (xxx = 32 to 1024)
```

Digitize

The DIGitize command causes an acquisition to take place on the specified channel(s) with the resulting data placed in the channel buffer. Upon completion, the data acquisition is stopped. See Chapter 4, DIGitize (Root Level Command) subsystem for more information. You digitize with the following command:

```
DIG xxx    (xxx = CHAN1-4)
```

Note

See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Disposition

This section discusses the disposition controls you can program with the WAVeform command. These controls allow the selection of:

- Source
- Format
- Reading Waveform Data
- Reading Preamble Data
- Using the Digitized Data

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available waveform commands, see Chapter 4, WAVeform subsystem.

Source

Select CHANnel 1-4, FUNCtion 1-2, or Waveform MEMory 1-4 as the source for all WAVeform commands. CHANnel 1 is selected at turn-on or reset. You select waveform source with the following command:

```
WAV:SOUR xxx    (xxx = CHAN1-4, FUNC1-2, or WMEM1-4)
```

Format

Three different formats are available to format digitized data when retrieved from the instrument. See Chapter 4, WAVeform:FORMat for information on WORD, BYTE, and COMPressed formats. BYTE is selected at turn-on or reset. You select format with the following command:

```
WAV:FORM xxx    (xxx = WORD, BYTE, or COMP)
```

Reading Waveform Data

The digitized data is read over the bus from the waveform memory or channel buffer specified using WAVeform:SOURce command. You read digitized data with the following query:

```
WAV:DATA?
```

Reading Preamble Data

The interpretation (preamble) data is read over the bus from the waveform memory or channel buffer specified using WAVeform:SOURce command. This data is needed to interpret the waveform data. You read preamble data with the following query:

```
WAV:PRE?
```

Using the Digitized Data

The returned data is read from the instrument starting at the left-most point on the waveform, and must be scaled for useful interpretation. The values needed to perform this task (x/y reference, origin, increment) are included in the preamble data. See Chapter 4, WAVeform subsystem for more information.

Example: Digitize a Waveform and Send Data to the Controller

This example uses the Oscilloscope module to digitize a waveform from CHANnel1, and send the results (waveform and preamble) to the controller. The waveform will be complete when 512 points have been averaged at least four times. The digitized data sent to the controller is to be in byte format.

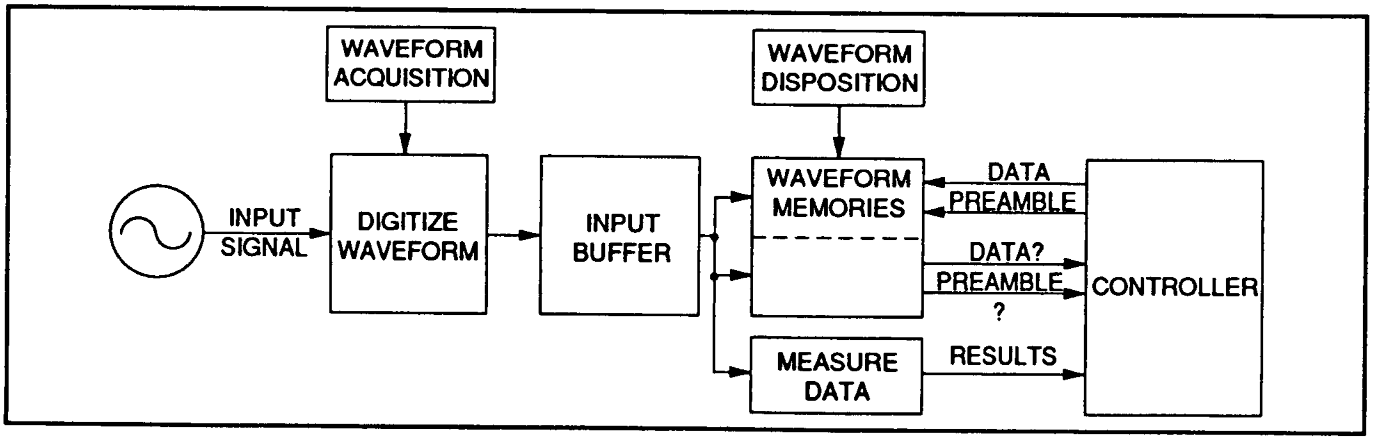


Figure 3-5. Example: Digitizing Waveforms

This example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10  DIM Pre$[100]                        String for preamble data.
 20  INTEGER Waveform(2000)               Temporary dimension for
                                          waveform data data.
 30  OUTPUT 70905;"SYST:HEAD ON"          System headers to on.
 40  OUTPUT 70905;"SYST:LONG ON"          System headers to long form.
 50  OUTPUT 70905;"*CLS"                  Clear status.
 60  OUTPUT 70905;"*RST"                  Resets the Oscilloscope to its
                                          default state (table 3-2).
 70  OUTPUT 70905;"AUT"                   Perform an autoscale.
 80  OUTPUT 70905;"ACQ:TYP AVER"          Acquisition type to average.
 90  OUTPUT 70905;"ACQ:COUN 4"            Number of averages to 4.
100  OUTPUT 70905;"ACQ:COMP 100"          Completion criteria to 100%.
110  OUTPUT 70905;"ACQ:POIN 512"          Data record to 512 points.
120  OUTPUT 70905;"DIG CHAN1"             Digitize channel 1 and place data
                                          in channel output buffer.
130  OUTPUT 70905;"WAV:SOUR CHAN1"        Waveform source is channel 1.
140  OUTPUT 70905;"WAV:FORM COMP"         Format for waveform data is 8-
                                          bits.
150  OUTPUT 70905;"WAV:PRE?"              Read preamble data.
160  ENTER 70905;Pre                      Enter preamble data.
170  OUTPUT 70905;"WAV:DATA?"             Read waveform data.
180  GOSUB Get_data                       Go to a subroutine that will read
                                          the header to determine the size of
                                          the waveform data, then re-
                                          dimension the waveform data
                                          array.
190  STOP                                 Stop main program.
200  Get_data: !                          Data retrieved routine.
210  ENTER 70905 USING "#,1A";One_char$   Enter one character at a time.
220  IF One_char$="#" THEN Found_pound
230  GOTO 210                             Loop to read next character.
240  Found_pound: !                       Read the record length routine.
250  ENTER 70905 USING "#,1D";Digits      Read and save first digit after "#".
260  ENTER 70905 USING "#,"&VAL$(Digits)&"D";Length
                                          Read the next XXX characters as
                                          the record length, where XXX is
                                          specified by Digits.
270  REDIM Waveform(1:Length)             Re-dimension Waveform to the
                                          actual record size.
280  ENTER 70905 USING "#,B";Waveform(*)  Enter waveform data.
290  ENTER 70905 USING "#,B";Crlf         Read carriage return.
300  RETURN                               Return to the main program.
310  END                                  Terminate program.
```

Comments

Block Data. Both preamble and waveform data is "definite-length block response data". This method allows any type of device-dependent data to be transmitted over the system interface as a series of 8-bit binary data types. This is particularly useful for sending large quantities of data or 8-bit extended ASCII codes. The syntax is a pound sign (#) followed by a non-zero digit representing the number of digits in the decimal integer. After the non-zero digit is the decimal integer that states the number of 8-bit data bytes being sent. This is followed by the actual data. For example, for transmitting 1024 bytes of data, the syntax is:

```
                       #41024<1024 bytes of data><term>
Number of digits  ___|  |              |               |
Number of bytes   ______|              |               |
Actual Data   _________________________|               |
Terminator   __________________________________________|
```

Reading Block Data. The example program (lines 200 to 300) finds the "#" sign, reads the number of digits and number of bytes, then redefines the field to the correct length. This method will work for all definite-length block response returned data (e.g., DISPlay:DATa? and WAVeform:DATa?).

When to Read Preamble. The preamble should be read prior to the waveform data.

When to Read Waveform Data. To obtain waveform data, you must specify the WAVeform parameters for the waveform data prior to sending the :DATA? query. After receiving the :DATA? query, the instrument will start passing the waveform information to the controller when addressed to talk.

Performing a Measurement

After the oscilloscope and measurement have been setup, and the waveform has been digitized, the actual measurement can be performed. Measurements include:

- Delay
- Fall time
- Frequency
- Period
- Pulse width
- Rise time

Notes

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of **ALL** available measurements, see Chapter 4, MEASure subsystem.

Remember, just because measurement results are obtained does not guarantee that they are correct. It is highly recommended that all measurement results be analyzed to verify authenticity. Incorrect oscilloscope and measurement setup can cause undesirable measurement results. **A returned +9.99999E+37 indicates an invalid measurement.**

See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Delay

Used to perform a delay measurement on the selected source(s). Delay time (in seconds) measured from:

- the first specified edge on one source to the next specified edge on the same source, or
- the first specified edge on one source to the first specified edge on another source.

Measurement definitions are specified by the MEASure:MODe command. You perform a delay measurement and return the measurement results with the following query:

```
MEAS:DEL?
```

Fall time

Used to perform a fall time measurement on the selected source. Fall time (in seconds) measured as time at lower threshold point minus time at upper threshold point. Measurement thresholds are specified by the MEASure:MODe command. You perform a fall time measurement and return the measurement results with the following query:

```
MEAS:FALL?
```

Frequency

Used to perform a frequency measurement on the selected source. Frequency (in hertz) of the first complete cycle is measured. Measurement thresholds are set to the 50% level (STANdard) or defined mid threshold (USER) as specified by the MEASure:MODe command. You perform a frequency measurement and return the measurement results with the following query:

```
MEAS:FREQ?
```

Period

Used to perform a period measurement on the selected source. Period (in seconds) of the first complete cycle is measured. Measurement thresholds are set to the 50% level (STANdard) or defined mid threshold (USER) as specified by the MEASure:MODe command. You perform a period measurement and return the measurement results with the following query:

```
MEAS:PER?
```

Pulse width

Used to perform a negative or positive pulse width measurements on the selected source. Pulse width of the first positive or negative pulse is measured. Measurement thresholds are set to the 50% level (STANdard) or a defined threshold (USER) as specified by the MEASure:MODe command.

You perform a negative pulse width measurement and return the measurement results with the following query:

```
MEAS:NWID?
```

You perform a positive pulse width measurement and return the measurement results with the following query:

```
MEAS:PWID?
```

Rise time

Used to perform a rise time measurement on the selected source. Rise time (in seconds) measured as time at upper threshold point minus time at lower threshold point. Measurement thresholds are specified by the MEASure:MODe command. You perform a rise time measurement and return the measurement results with the following query:

```
MEAS:RIS?
```

Notes

If the identical command is sent without the "?", the Oscilloscope will be placed in the continuous measurement mode, and start the specified measurement. Measurement results are returned using the MEASure:RESults? query. The query reads the measurement results but does not stop the continuous mode.

These commands are also used during limit testing. See Chapter 4, MEASure subsystem for more information.

Measurement Examples

The following is a list of examples provided to illustrate using COMP commands to perform basic measurements/functions using the Oscilloscope module.

- Autoscale Frequency Measurement
- Manual Fall Time Measurement
- Fall Time Measurement using the Expanded Window
- Using Delay Trigger
- Limit Testing

Caution

MAXIMUM INPUT VOLTAGE. The maximum voltage that can be applied to the four input connectors is 5 Vrms at 50Ω or ±250 V (dc+peak ac<10 kHz) at 1MΩ.

Note

The following examples are intended to provide only a brief overview of the necessary commands required for basic operation. However, these examples can be used to provide a good starting point for much more complex programs. For a complete list and description of ALL COMP commands, see Chapter 4.

All the examples in this section are written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Example: Autoscale Frequency Measurement

This example uses the Oscilloscope module to measure the frequency of an unknown signal connected to Input 2.

Caution

MAXIMUM INPUT VOLTAGE. The maximum voltage that can be applied to any of the four input connectors is 5 Vrms at 50Ω or ±250 V (dc+peak ac<10 kHz) at 1MΩ.

Note

Because of the Autoscale function, disconnect any input signal connected to input 1, 3, or 4.

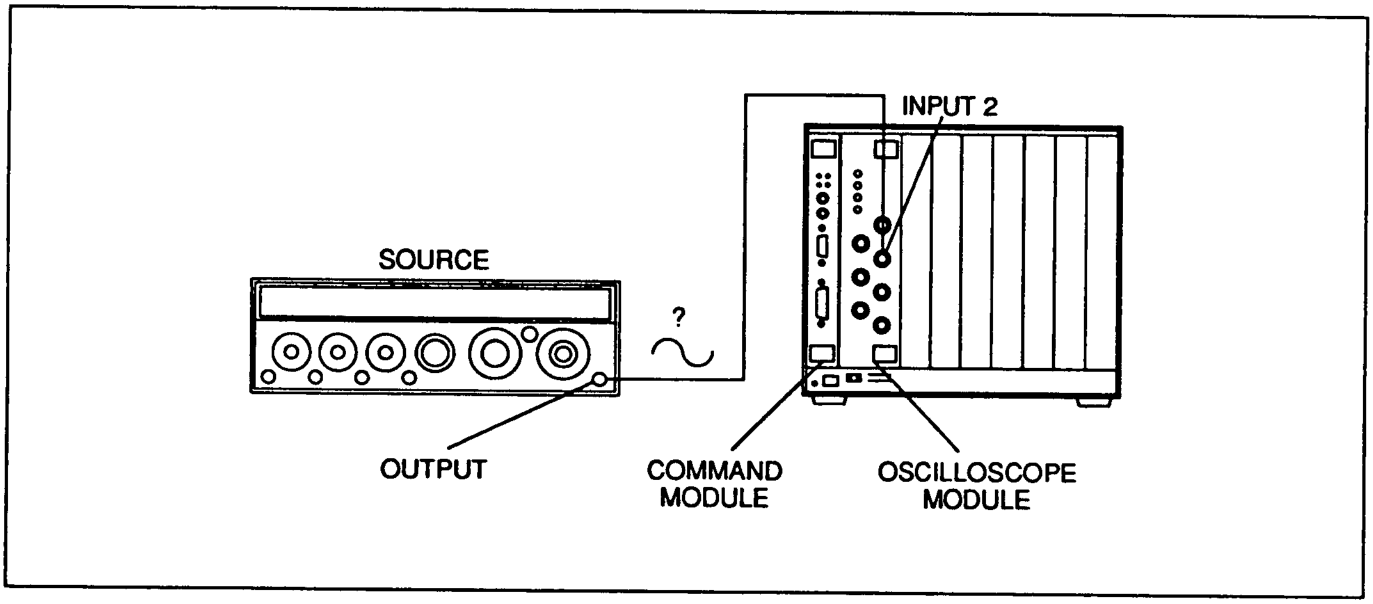


Figure 3-6. Example: Autoscale Frequency Measurement

Execute:

```
 10 DIM A$[25]                           String for measurement data and
                                         headers.
 20 OUTPUT 70905;"*CLS"                  Clear status.
 30 OUTPUT 70905;"*RST"                  Resets the Oscilloscope to its
                                         default state (table 3-2).
 40 OUTPUT 70905;"AUT"                   Perform autoscale.
 50 OUTPUT 70905;"SYST:HEAD ON"          Return headers.
 60 OUTPUT 70905;"DIG CHAN2"             Digitize channel 2 using default
                                         acquisition parameters (see table
                                         3-2).
 70 OUTPUT 70905;"MEAS:SOUR CHAN2"       Measurement source to channel
                                         2.
 80 OUTPUT 70905;"MEAS:FREQ?"            Read measurement results.
 90 ENTER 70905;A$                       Enter measurement results.
100 PRINT A$                             Print measurement results.
110 END                                  Terminate program.
```

Comments

Signal Connection. An input signal must be connected to the Oscilloscope before initiating autoscale, and remain connected until the measurement is terminated.

Multiple Signal Connection using Autoscale. When more than one Oscilloscope input is connected to a signal source, initiating autoscale will setup the Oscilloscope controls using the signal connected closest to channel 1. If a signal is not found on input 1 then input 2 is used, if a signal is not found on input 2 then input 3 is used, etc.

Measurement Time. The time it takes to complete a measurement is dependent on the input signal, acquisition type, and completion criteria. Under certain circumstances, a single measurement can take many hours to complete. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Numeric Results. Set SYSTem:HEADer to OFF to return numeric data.

Digitize. This measurement could be performed without digitizing the waveform, however the returned results will not be as accurate or as consistent as when using the digitize command.

Example: Manual Fall Time Measurement

This example uses the Oscilloscope module to measure the fall time of a signal connected to Input 3 using a 1MΩ 10:1 probe. The expected input is a 1.5 V clock at 10 MHz. The user is notified if the returned measurement results are not within specific limits.

Notes

When measuring fall time, the trailing (negative-going) edge of the waveform must be present. In order to obtain accurate results (example signal is a clock pulse-minimum fall time):

- the time base should be setup so the falling edge is maximized over the time base range, and
- the trigger should be set so the falling edge is centered.

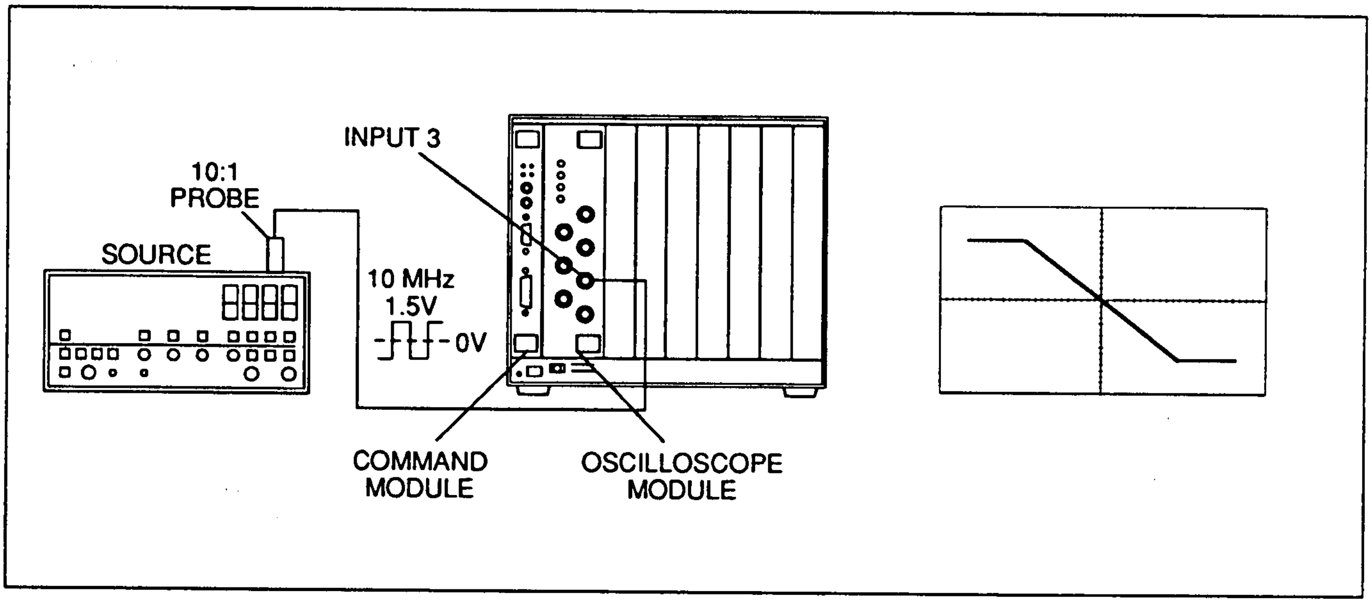


Figure 3-7. Example: Manual Fall Time Measurement

Execute:

`10 OUTPUT 70905;"SYST:HEAD OFF"`	*System headers off to read numeric data.*
`20 OUTPUT 70905;"*CLS"`	*Clear status.*
`30 OUTPUT 70905;"*RST"`	*Resets the Oscilloscope to its default state (table 3-2).*
`40 OUTPUT 70905;"CHAN3:PROB 10"`	*Set input 3 probe attenuation factor at 10:1.*
`50 OUTPUT 70905;"CHAN3:RANG 2"`	*Set input 3 full scale vertical range to 2 volts (expected input is 1.5V).*
`60 OUTPUT 70905;"TIM:RANG 100NS"`	*Set full scale horizontal range to 100 ns (expected period).*
`70 OUTPUT 70905;"TRIG:SOUR CHAN3"`	*Trigger source to channel 3.*
`80 OUTPUT 70905;"TRIG:SLOP NEG"`	*Trigger slope to negative (to ensure falling edge is acquired).*
`90 OUTPUT 70905;"TRIG:LEV .75"`	*Trigger level to .75 volts (approximately one-half the expected input).*
`100 OUTPUT 70905;"DIG CHAN3"`	*Digitize channel 3 using default acquisition parameters (see table 3-2).*
`110 OUTPUT 70905;"MEAS:SOUR CHAN3"`	*Measure channel 3.*
`120 OUTPUT 70905;"MEAS:FALL?"`	*Perform fall time measurement and return results.*
`130 ENTER 70905;Results`	*Enter measurement results.*
`140 IF Results<21E-9 THEN 160`	*Verify measurement results are less than than 21 nsec.*
`150 PRINT "Measurement out of Spec"`	*Print measurement flag.*
`160 END`	*Terminate program.*

Comments

Signal Connection. An input signal must be connected to the Oscilloscope before initiating a measurement, and remain connected until the measurement is terminated.

Probe Attenuation Factor. Probe attenuation factor should be entered before the range and offset parameters. Setting the probe attenuation factor does not change the input sensitivity, but does scale the vertical and trigger parameters to compensate for the entered factor.

Range. Both vertical (CHANnel) and horizontal (TIMebase) RANGe parameters are specified for full scale axis, and not per division values.

Window Command. An alternate method of presenting the waveform for measurement is to use the WINDow command. See Fall Time Measurement using the Expanded Window example in this section for more information.

Measurement Time. The time it takes to complete a measurement is dependent on the input signal, acquisition type, and completion criteria. Under certain circumstances, a single measurement can take many hours to complete. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Example: Manual Fall Time Measurement Using the Expanded Window

This example uses the Oscilloscope module to measure the fall time of a signal connected to Input 3. Measurement is taken using the expanded window. The expected input is a 1.5 V clock at 1 MHz. The user is notified if returned measurement results are not within specific limits.

Notes

When measuring fall time, the trailing (negative-going) edge of the waveform must be present. In order to obtain accurate results (example signal is a clock pulse-minimum fall time):

- the expanded window time base should be setup so the falling edge is maximized over the time base range, and
- the trigger should be set so the falling edge is centered.

When using the expanded window, the main sweep is set to present the waveform, then the window parameters are positioned on the main sweep to access the desired portion of the waveform being measured.

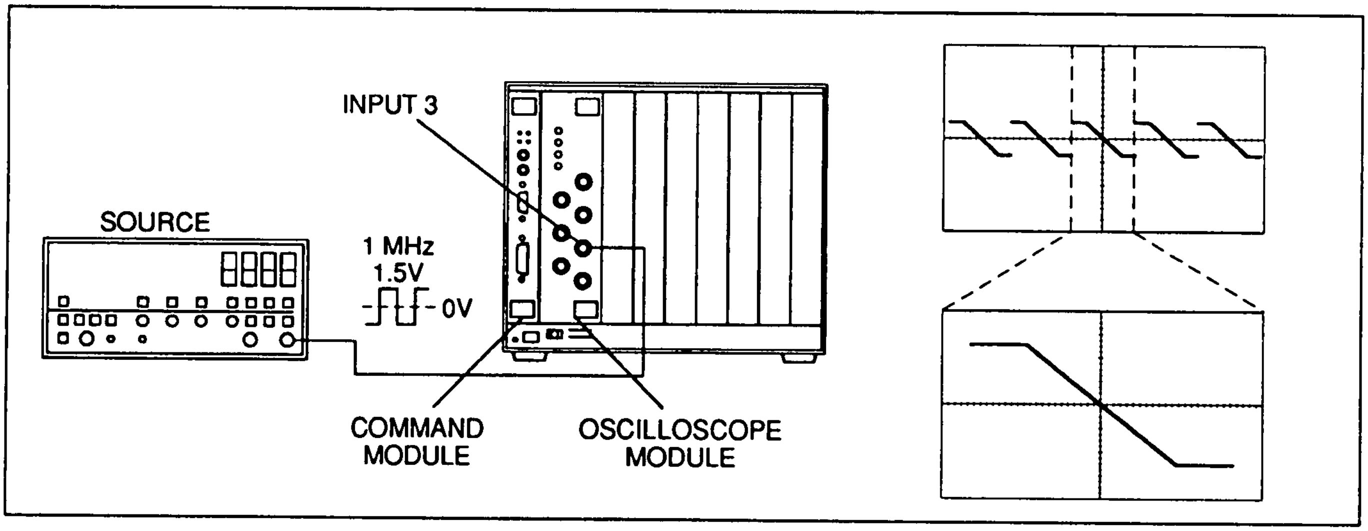


Figure 3-8. Example: Manual Fall Time Measurement with Expanded Window

Execute:

```
 10 OUTPUT 70905;"SYST:HEAD OFF"               System headers off to read
                                                numeric data.
 20 OUTPUT 70905;"*CLS"                        Clear status.
 30 OUTPUT 70905;"*RST"                        Resets the Oscilloscope to its
                                                default state (table 3-2).
 40 OUTPUT 70905;"BLAN CHAN1"                  Turn channel 1 off.
 50 OUTPUT 70905;"VIEW CHAN3"                  Turn channel 3 on.
 60 OUTPUT 70905;"RUN"                         Start data acquisition.
 70 OUTPUT 70905;"CHAN3:COUP DCF"              Set input 3 coupling to DC at 50Ω
 80 OUTPUT 70905;"CHAN3:RANG 2"                Set input 3 full scale vertical range
                                                to 2 volts (expected input is 1.5V).
 90 OUTPUT 70905;"TIM:RANG 2US"                Set full scale horizontal range to 2
                                                µs (5X the expected period).
100 OUTPUT 70905;"TRIG:SOUR CHAN3"             Trigger source to channel 3.
110 OUTPUT 70905;"TRIG:SLOP NEG"               Trigger slope to negative (to
                                                ensure falling edge is acquired).
120 OUTPUT 70905;"TRIG:LEV 0"                  Trigger level to 0 volts.
130 OUTPUT 70905;"TIM:WIND:DEL 0"              Window delay to 0 seconds
                                                (center of the waveform).
140 OUTPUT 70905;"TIM:WIND:RANG 100E-9"        Window range to 100nsec ( to 0
                                                seconds (the expected period).
150 OUTPUT 70905;"DIG CHAN3"                   Digitize channel 3 using default
                                                acquisition parameters (see table
                                                3-2).
160 OUTPUT 70905;"TIM:WIND ON"                 Expanded window on.
170 OUTPUT 70905;"MEAS:SOUR CHAN3"             Measure channel 3.
180 OUTPUT 70905;"MEAS:FALL?"                  Perform fall time measurement
                                                on expanded window and return
                                                results.
190 ENTER 70905;Results                        Enter measurement results.
200 IF Results<5.E-9 THEN 220                  Verify measurement results are
                                                less than 5 nsec.
210 PRINT "Measurement out of Spec"            Print measurement flag.
220 END                                        Terminate program.
```

Comments

Signal Connection. An input signal must be connected to the Oscilloscope before initiating a measurement, and remain connected until the measurement is terminated.

Probe Attenuation Factor. Probe attenuation factor should be entered before the range and offset parameters. Setting the probe attenuation factor does not change the input sensitivity, but does scale the vertical and trigger parameters to compensate for the factor.

Range. Both vertical (CHANnel) and horizontal (TIMebase) RANGe parameters are specified for full scale, and not per division values.

Window Control. Two commands are used to move the expanded window on the main sweep. WINDow:RANGe command is used to set the expanded window full scale horizontal axis, and WINDow:DELay sets the expanded window time base delay.

Measurement Time. The time it takes to complete a measurement is dependent on the input signal, acquisition type, and completion criteria. Under certain circumstances, a single measurement can take many hours to complete. See Appendix C, Optimizing Measurements, for additional information.

Example: Using Delay Trigger

The trigger is setup in the delay mode to qualify on the first rising edge of the first burst, delay for 2.5µsec to obtain a stable waveform, then trigger on the fifth edge (middle) of the second burst. The expected input is a 1 V burst pulse with ten 5 MHz bursts that repeat every 50µsec.

Note

For the example, the parameters are setup using autoscale, then adjusted as required.

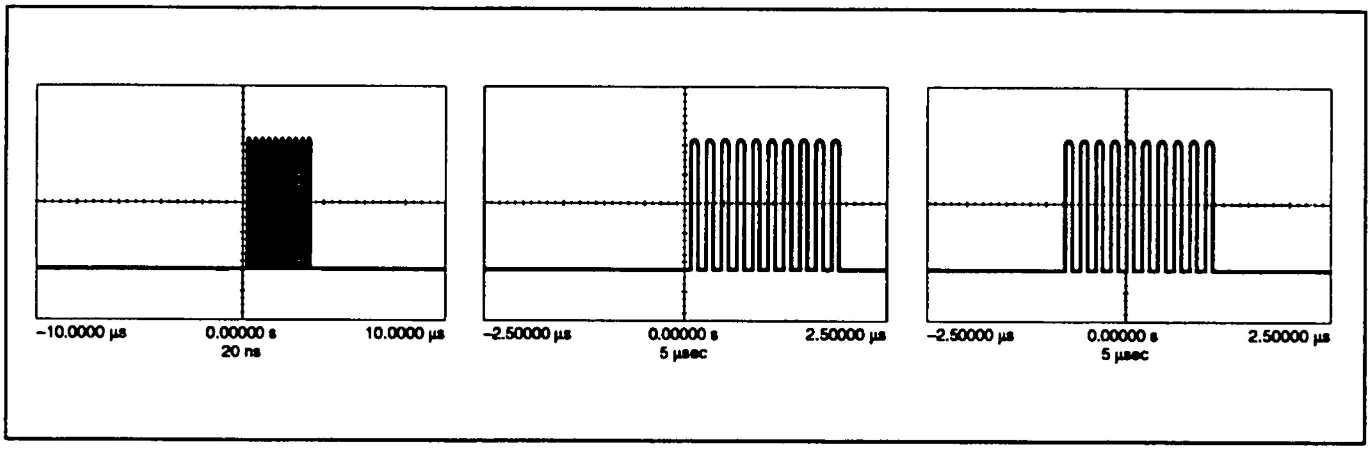


Figure 3-9. Example: Using the Delay Trigger

Execute:

```
 10  OUTPUT 70905;"*CLS"                    Clear status.
 20  OUTPUT 70905;"*RST"                    Resets the Oscilloscope to its
                                            default state (table 3-2).
 30  OUTPUT 70905;"AUT"                     Perform an autoscale.
 40  OUTPUT 70905;"TRIG:MOD DEL"            Delay trigger mode.
 50  OUTPUT 70905;"TRIG:QUAL:EDG"           Qualify mode to edge.
 60  OUTPUT 70905;"TRIG:DEL TIM,2.5US"      Delay time to 2.5µsec to obtain a
                                            stable trigger.
 70  OUTPUT 70905;"TRIG:OCC 1"              Trigger on first event.
 80  OUTPUT 70905;"TRIG:OCC:SLOP POS"       Trigger on rising edge.
 90  OUTPUT 70905;"TRIG:OCC:SOUR CHAN1"     Trigger source to channel 1.
100  OUTPUT 70905;"TRIG:SLOP POS"           Trigger on rising edge.
110  OUTPUT 70905;"TRIG:SOUR CHAN1"         Trigger source to channel 1.
120  OUTPUT 70905;"TIM:RANG 5US"            Time base to 5µsec full scale.
130  OUTPUT 70905;"TRIG:OCC 5"              Trigger on fifth event.
140  END                                    Terminate program.
```

Comments

Delay Trigger Qualify Modes. In delay mode, a trigger can be qualified using edge, pattern, or state mode before the delay is defined.

Multiple Signal Connection using Autoscale. When more than one Oscilloscope input is connected to a signal source, initiating autoscale will setup the Oscilloscope controls using the signal connected closest to channel 1. If a signal is not found on input 1 then input 2 is used, if a signal is not found on input 2 then input 3 is used, etc.

After Triggering. After a delay trigger is setup, any parameter can be measured, the waveform data can be digitized and measured, or read over the bus.

Measurement Time. The time it takes to complete a measurement is dependent on the input signal, acquisition type, and completion criteria. Under certain circumstances, a single measurement can take many hours to complete. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Example: Limit Testing

This example uses the Oscilloscope module to measure the frequency, duty cycle, and peak-to-peak voltage of a signal connected to Input 2. The expected input is a 1 Vp-p, 10 MHz, 50% duty cycle sinewave. If any of the the input signal measured values are not within the specified limits, the waveform will be saved in waveform memory 1, and the test will stop.

Note

For the example, the parameters are setup using autoscale, then adjusted as required.

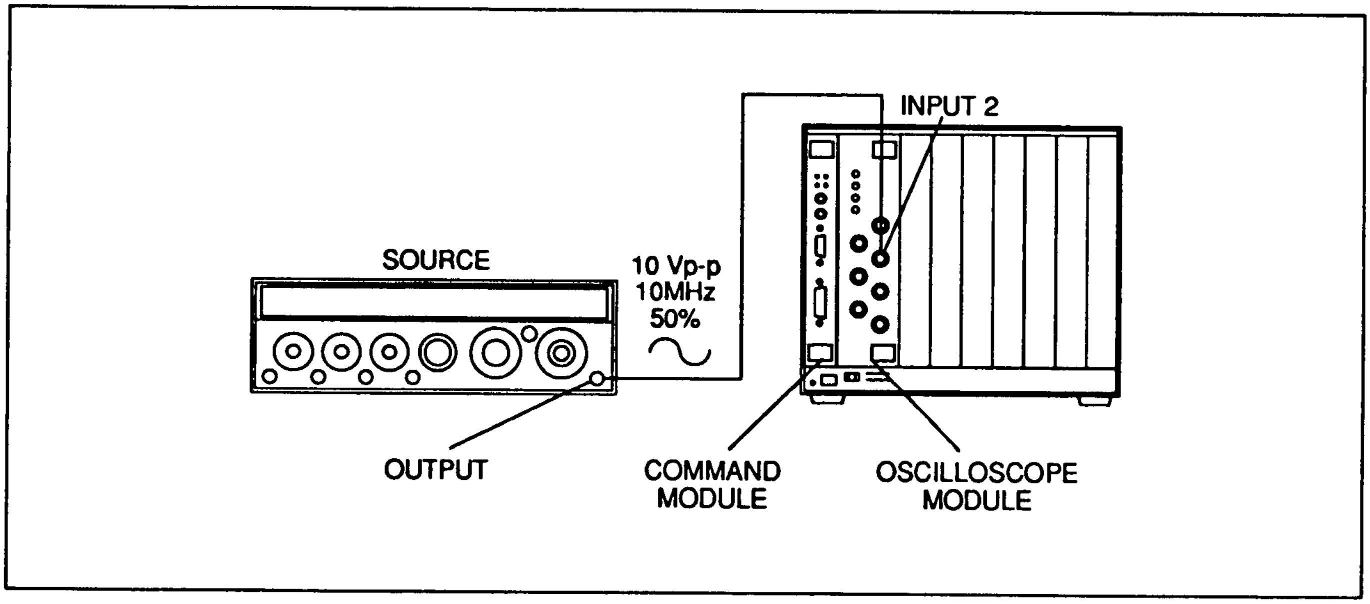


Figure 3-10. Example: Limit Testing

Execute:

```
05  DIM Results$[25]                      String for failure results.
10  OUTPUT 70905;"*CLS"                   Clear status.
20  OUTPUT 70905;"*RST"                   Resets the Oscilloscope to its
                                          default state (table 3-2).
30  OUTPUT 70905;"AUT"                    Perform an autoscale (no signal
                                          connected to channel 2).
40  OUTPUT 70905;"TIM:RANG 200E-9"        Set full scale horizontal range to
                                          200 ns (expected period).
50  OUTPUT 70905; "MEAS:SCR"              Clear measurement queue.
60  OUTPUT 70905; "MEAS:SOUR CHAN2"       Measure channel 2.
70  OUTPUT 70905; "WAV:SOUR CHAN2"        Violation source is channel 2.
80  OUTPUT 70905; "MEAS:FREQ"             Start a continuous frequency
                                          measurement on channel 2.
90  OUTPUT 70905; "MEAS:COMP FREQ,10.5E6,9.5E6"
                                          Configure a limit test for
                                          frequency measurement with
                                          acceptable results from 9.5 to
                                          10.5MHz.
```

```
100 OUTPUT 70905; "MEAS:DUT"                Start a continuous duty cycle
                                            measurement on channel 2.
110 OUTPUT 70905; "MEAS:COMP DUT,55,45"
                                            Configure a limit test for duty
                                            cycle measurement with
                                            acceptable results from 45 to 55%.
120 OUTPUT 70905; "MEAS:VPP"                Start a continuous peak-to-peak
                                            voltage measurement on channel
                                            2.
130 OUTPUT 70905; "MEAS:COMP VPP,1.2,0.8V"
                                            Configure a limit test for peak-to-
                                            peak voltage measurement with
                                            acceptable results from 0.8 to
                                            1.2Vp-p.
140 OUTPUT 70905; "MEAS:POST STOP"          Stop limit test after violation.
150 OUTPUT 70905; "MEAS:DEST WMEM1"         Waveform memory 1 is the
                                            destination for limit test
                                            violations.
160 OUTPUT 70905; "MEAS:LIM MEAS"           Start limit test.
165 OUTPUT 70905; "DIG CHAN1"               Acquire waveform data.
170 OUTPUT 70905; "LTER?"                   Has limit test failed?
180 ENTER 70905; Fail                       Read results.
190 IF Fail=0 THEN 170                      Read limit test event register until
                                            limit test fails.
200 OUTPUT 70905; "MEAS:RES?"               Query instrument to return failed
                                            measurement results.
280 ENTER 70905; Results$                   Read measurement results.
290 PRINT Results$                          Print measured results.
310 END                                     Terminate program.
```

Comments

Violation Choices: The limit test can be stopped or continued after a violation has been found. If desired, the data associated with the violation can be saved in pixel or waveform memory.

Determining Limit Test Status. Failures can be determined by reading the limit test event register (LTER?) and/or by analyzing the measurement results (RESults?).

Limit Test on More than One Channel. Limit tests can be performed on up to three different channel at the same time by stepping through the MEASure:SOURce parameter. Only one channel can be specified (using the WAVeform:SOURce command) as the source when saving the limit test violation data.

Multiple Signal Connection using Autoscale. When more than one Oscilloscope input is connected to a signal source, initiating autoscale will setup the Oscilloscope controls using the signal connected closest to channel 1. If a signal is not found on input 1 then input 2 is used, if a signal is not found on input 2 then input 3 is used, etc.

Measurement Time. The time it takes to complete a measurement is dependent on the input signal, acquisition type, and completion criteria. Under certain circumstances, a single measurement can take many hours to complete. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Recalling and Saving States

This section contains information about saving and recalling current Oscilloscope module states.

Storing States

The *SAV *<numeric_state>* command saves the current instrument state. The state number (1-4) is specified in the *<numeric_state>* parameter. All of the Oscilloscope and measurement setup parameters are saved.

Recalling States

The *RCL *<numeric_state>* command recalls a previously saved or existing state.

- Enter the number 0 in the *<numeric_state>* parameter to recall the configuration prior to executing the AUToscale, *RCL, CHANnel:TTL, or CHANnel:ECL commands.

- Enter the number (1-4) in the *<numeric_state>* parameter of the desired saved state. If *SAV was not previously executed using the selected number, the Oscilloscope module will generate an error.

Recalling and Storing Waveforms

This section contains information about recalling and storing current Oscilloscope module waveforms. These controls allow for:

- Storing
- Viewing
- Blanking
- Erasing
- Merging

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of these commands, see Chapter 4, STORe, VIEW, BLANk, and ERASe.

Storing Waveforms

The STORe command is used to save an active, previously stored, or function waveform in a non-volatile waveform memory location. The following waveforms are available for viewing:

CHANnel 1-4 - active waveform from input 1-4

WMEMory 1-4- stored in waveform memory (non-volatile)

FUNCtion 1-2 - function waveform (+,–,X,inverted,only)

You store a waveform with the following command:

```
STOR xxx,yyyy        (xxx = source waveform CHAN1-4, WMEM
                     1-4, or FUNC 1-2 and yyyy= memory
                     destination WMEM 1-4)
```

Note

Whatever is present at the specified source (waveform, baseline, etc) is what will be saved in the specified memory destination.

Viewing Waveforms

The VIEW command is used to present an active, previously stored, or function waveform. The following waveforms are available for viewing:

CHANnel 1-4 - active waveform from input 1-4
PMEMory 0-2 - stored in pixel memory (volatile)
WMEMory 1-4- stored in waveform memory (non-volatile)
FUNCtion 1-2 - function waveform (+,–,X,inverted,only)

You view a waveform with the following command:

```
VIEW xxx             (xxx = CHAN1-4, PMEM 0-2, WMEM 1-4, or
                     FUNC 1-2)
```

Blanking Waveforms

The BLANk command is used to stop presenting an active, previously stored, or function waveform. The following waveforms are available for blanking:

CHANnel 1-4 - active waveform from input 1-4

PMEMory 0-2 - stored in pixel memory (volatile)

WMEMory 1-4- stored in waveform memory (non-volatile)

FUNCtion 1-2 - function waveform (+,–,X,inverted,only)

You blank a waveform with the following command:

```
BLAN xxx            (xxx = CHAN1-4, PMEM 0-2, WMEM 1-4, or
                    FUNC 1-2)
```

Note

All unused channels should be blanked. See Appendix C, Optimizing Measurements, for additional information.

Erasing Waveforms

The ERASe command is used to erase the contents of pixel memory. You erase pixel memory with the following command:

```
ERAS xxx            (xxx = PMEM 0-2)
```

Merging Waveforms

The MERGe command is used to merge the contents of pixel memory 0 with the current contents of pixel memory 1 or 2. You merge pixel memories with the following command:

```
MERG xxx            (xxx = PMEM 1-2)
```

Querying the Oscilloscope

This section summarizes the query commands you can use to determine the configuration or state of the Oscilloscope. All commands end with the "?" which puts the data into the output buffer where you can retrieve it to your computer. See Chapter 4 for more information.

Unless otherwise specified, <n> is the channel number (1-4).

Query	Description
ACQuire	Acquire subsystem queries
:COMPlete?	Acquisition complete value
:COUNt?	Acquisition count value
:POINts?	Acquisition points value
:TYPe?	Acquisition type
BNC?	Probe compensation connector output
CALibrate	Calibrate subsystem queries
:REPort? CHANnel<n>	Calibration report
:TNULl?	Current time null values for channel pairs 1-2, 1-3, and 1-4.
CHANnel<n>	Channel subsystem queries
:COUPling?	Channel's coupling
:HFReject?	Channel's low pass filter state
:LFReject?	Channel's high pass filter state
:OFFSet?	Channel's offset value
:PROBe?	Channel's probe attenuation
:RANGe?	Channel's range value
DISPlay	Display subsystem queries
:DATa?	Display data (block)
:PERSistence?	Display persistence mode
:SOURce?	Display source setting
FUNCtion<n>	Function subsystem queries
:OFFSet?	Function offset value
:RANGe?	Function range value
LTER?	Limit test event register value
MEASure	Measure subsystem queries
:ALL?	Measure all parameters, return results
:COMPare?	Limit test compare mode and values
:CURSor? *<xxxx>*	Cursor (time/voltage) value, where xxxx is the type
:DEFine? *<xxxx>*	Measurement definition (delay and pulse width), where xxxx is measurement and parameters
:DELay?	Measure delay, return results
:DESTination?	Destination for limit test violation
:DUTycycle?	Measure duty cycle, return results
:ESTArt?	Edge start marker position
:ESTOp?	Edge stop marker position
:FALLtime?	Measure fall time, return results

Querying the Oscilloscope — Continued

Unless otherwise specified, <n> is the channel number (1-4).

Query	Description
MEASure—Continued	Measure subsystem queries
:FREQuency?	Measure frequency, return results
:LOWer?	Lower threshold value
:MODe?	Measurement mode definitions
:NWIDth?	Measure negative pulse width, return results
:OVERshoot?	Measure overshoot, return results
:PERiod?	Measure period, return results
:POSTfailure?	Limit test post-failure instruction
:PRECision?	Always returns COARse
:PREShoot?	Measure preshoot, return results
:PWIDth?	Measure positive pulse width, return results
:RESults?	Measurement queue contents
:RISetime?	Measure rise time, return results
:SOURce?	Measurement source setting
:STATistics?	Statistics mode
:TDELta?	Time difference between markers
:TMAX?	Time at maximum voltage
:TMIN?	Time at minimum voltage
:TSTArt?	Time start marker position
:TSTOp?	Time stop marker position
:TVOLt? *<xxxx>*	Time interval between trigger and defined event, where xxxx is defined event
:UNITs?	Threshold level units
:UPPer?	Upper threshold value
:VACRms?	Measure AC RMS voltage, return results
:VAMPlitude?	Measure voltage amplitude, return results
:VAVerage?	Measure average voltage, return results
:VBASe?	Measure base voltage, return results
:VDCRms?	Measure DC RMS voltage, return results
:VDELta?	Voltage difference between markers
:VMAX?	Measure maximum voltage, return results
:VMIN?	Measure minimum voltage, return results
:VPP?	Measure peak-to-peak voltage, return results
:VRELative?	Current relative voltage stop marker position
:VRMS?	Measure RMS voltage, return results
:VSTArt?	Voltage start marker position
:VSTOp?	Voltage stop marker position
:VTIMe? *<xxxx>*	Voltage level at specified time, where xxxx is time
:VTOP?	Measure top voltage, return results
MEMory	Memory subsystem queries
:VME	
:ADDRess?	Memory address selected
:MEASure	
:ADDRess?	Memory measurement address selected
:STATe?	Memory measure state
:SIZE?	Memory size
:STATe?	Overall VME memory state

Querying the Oscilloscope — Continued

Unless otherwise specified, <n> is the channel number (1-4).

Query	Description
OUTPut	Output subsystem queries
:ECLTrg<xxxx>	Select ECL trigger bus line (0-1)
[:STATe]?	Enable state of ECL trigger bus line selected
:EXTernal	Front panel trigger connector
[:STATe]?	Enable state of front panel trigger
[:STATe]?	Overall output state
:TTLTrg<xxxx>	Select TTL trigger bus line (0-7)
[:STATe]?	Enable state of TTL trigger bus line selected
SUMMary	Summary subsystem commands
:QUEStionable?	Questionable event register value
:CALibration?	Calibration event register value
:CHANnel<n>?	Channel's event register value
:AD?	Channel's AD event register value
:DELay?	Channel's delay event register value
:GAIN?	Channel's gain event register value
:HYSTeresis?	Channel's hysteresis event register value
:LTRigger?	Channel's logic trigger event register value
:OFFSet?	Channel's offset event register value
:TNULl?	Channel's time null event register value
:TRIGger?	Channel's trigger event register value
:DCALibration?	Default calibration event register value
:PROBe?	Probe event register value
:TEST?	Test event register value
:ACQuisition?	Acquisition test event register value
:AD?	Acquisition test AD event register value
:ATRigger?	Acquisition test analog trigger event register value
:DA?	Acquisition test DA event register value
:LTRigger?	Acquisition test logic trigger event register value
:TIMebase?	Acquisition test time base event register value
:INTerpolator?	Time base interpolator event register value
:RAM?	RAM test event register value
:ACQuisition?	Acquisition RAM test event register value
:DISPlay?	Display RAM test event register value
:NVOLatile?	Non-volatile RAM test event register value
:SYSTem?	System RAM test event register value
:ROM?	ROM test event register value
:NPRotect?	Non-protected RAM test event register value
:SYSTem?	System ROM test event register value
SYSTem	System subsystem queries
:ERRor? <xxxx>	Error number and messages, where xxxx specifies message type
:HEADer?	Command header state
:LANGuage?	Programming language selected
:LONGform?	Command header form
:SETup?	Setup data (block)

Querying the Oscilloscope — Continued

Unless otherwise specified, <n> is the channel number (1-4).

Query	Description
RUN?	Current acquisition state
STATus? <xxxx>	Channel, function, waveform memory, or pixel memory status, where xxxx is source
TER?	Trigger event register value
TIMebase	Time base subsystem queries
:DELay?	Time base delay value
:MODe?	Time base mode
:RANGe?	Time base range value
:REFerence?	Delay reference selection
:WINDow?	Expanded window state
:DELay?	Expanded window delay value
:RANGe?	Expanded window range value
TRIGger	Trigger subsystem queries
:CONDition?	Trigger condition and values
:DELay?	Trigger delay type (time or event) and value
:SLOPe?	Trigger delay slope selected
:SOURce?	Trigger delay source selected
:FIELd?	TV Trigger field selected
:HOLDoff?	Trigger holdoff type (time or events) and value
:LEVel?	Trigger level value
:LINe?	TV Trigger line selected
:LOGic?	Logic level selected
:MODe?	Trigger mode selected
:OCCurrence?	Occurrence value
:SLOPe?	Occurrence slope selected
:SOURce?	Occurrence source selected
:PATH?	Source selected for logic command
:POLarity?	TV Trigger polarity selected
:QUALify?	Qualify mode selected
:SENSitivity?	Trigger sensitivity selected
:SLOPe?	Trigger slope selected
:SOURce?	Trigger source selected
:STANdard?	TV Trigger standard selected
WAVeform	Waveform subsystem
:COUNt?	Always returns 1
:DATa?	Waveform data (block)
:FORMat?	Waveform data format selected
:POINts?	Waveform data points value
:PREamble?	Preamble (block)
:SOURce?	Waveform source selected
:TYPe?	Waveform data type selected
:XINCrement?	Time difference of data points
:XORigin?	Time of first data point
:XREFerence?	Always returns 0
:YINCrement?	Voltage difference of data points
:YORigin?	Voltage at center of waveform
:YREFerence?	Data point at y-origin

Contents

Chapter 4 — Oscilloscope HP 54503A Compatible Language Command Reference

4

Oscilloscope HP 54503A Compatible Language Command Reference

Using This Chapter

This chapter describes Hewlett-Packard 54503A Compatible Language (COMP) commands and summarizes IEEE 488.2 Common (*) commands applicable to the Oscilloscope module.

See the HP 54503A Programming Guide for additional information on COMP and common commands. This chapter contains the following sections:

Command Types

Commands are separated into two types: IEEE 488.2 Common Commands and Hewlett-Packard 54503A Compatible Language (COMP) Commands.

Common Command Format

The IEEE 488.2 standard defines the Common Commands that perform functions like reset, self-test, status byte query, etc. Common commands are four or five characters in length, always begin with the asterisk character (*), and may include one or more parameters. The command keyword is separated from the first parameter by a space character. Some examples of Common Commands are shown below:

```
*RST        *CLS        *STB?
```

COMP Command Format

The COMP commands perform functions like setting parameters, performing measurements, querying instrument states, and retrieving data. A subsystem command structure is a hierarchical structure that usually consists of a top level (or root) command, one or more lower level sub commands, and their parameters. The following example shows part of a typical subsystem:

```
TIMebase
        :RANGe <range>
        :WINDow ON|OFF
                :DELay?
```

TIMebase is the root command, :RANGe and :WINDow are the second level sub commands with *<range>* and ON|OFF as parameters, and :DELay? is a third level query.

Command Separator

A colon (:) always separates one command from the next lower level command as shown below:

```
TIMebase:WINDow:DELay?
```

Colons separate the root command from the second level command (TIMebase:WINDow), and the second level from the third level query (WINDow:DELay?).

Abbreviated Commands

The command syntax shows most commands as a mix of upper and lower case letters. The upper case letters indicate the abbreviated spelling for the command. For shorter program lines, send only the abbreviated form. For better program readability, you may send the entire command. The instrument will only accept either the abbreviated form or the entire command.

For example, if the command syntax shows TIMebase, then TIM and TIMEBASE are both acceptable forms. Other forms of TIMebase, such as TM will generate an error. You may use upper or lower case letters. Therefore, TIMEBASE, and TIMeBaSe are acceptable.

Implied Commands

Implied commands are those which appear in square brackets ([ ]) in the command syntax. (Note that the brackets are not part of the command and are not sent to the instrument.) Suppose you send a root level and second level command, but do not send the third level implied command. In this case, the instrument assumes you intend to use the implied command and it responds as if you had sent it. Examine the portion of the SUMMary subsystem shown below:

```
SUMMary
     :QUEStionable
          [:EVENt]?
```

The third level query [:EVENt]? is implied. To query the instruments operation event register, you can send either of the following command statements:

SUMM:QUES? *or* **SUMM:QUES:EVEN?**

Parameters

Parameter Types. The following table contains explanations and examples of parameter types you might see later in this chapter.

Parameter Type	Explanations and Examples
Numeric	Accepts all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation. 123 or 1.23E2; –123 or –1.23E2; .123, 1.23E–1, or 1.23000E–01. Accepts all commonly used suffixes with decimal representations of numbers including optional signs, and decimal points. .123S or 123MS; 1234OHM or 1.234KOHM. Voltage = "UV" for E–6, "MV" for E-3, "V" for E0, "KV" for E3. Percent = "PCT". Ohms = "OHM", "KOHM" for E3, "MOHM" for E6. Frequency = "HZ" for E0, "KHZ" for E3, "MHZ" for E6, "GHZ" for E9. Time = "PS" for E–12, "NS" for E-9, "US" for E–6, "MS" for E-3, "S" for E0. Special cases include MIN and MAX. MIN (selects minimum value available), and MAX (selects maximum value available).
Boolean	Represents a single binary condition that is either true or false. 1 or ON; 0 or OFF
Block	Definite block program data format specified in IEEE 488.2.
Discrete	Selects from a finite number of values. These parameters use mnemonics to represent each valid setting. An example is the TIMebase:MODe *<mode>* command where *<mode>* can be AUTo, TRIGgered, or SINGle.

Optional Parameters. Parameters shown within square brackets ([ ]) are optional parameters. (Note that the brackets are not part of the command and are not sent to the instrument.) If you do not specify a value for an optional parameter, the instrument chooses a default value. For example, consider the MEM:VME:ADDR? [<MIN|MAX>] command. If you send the command without specifying a parameter, the present external VME memory address is returned. If you send the MIN parameter, the command returns the lowest address available (2097152). If you send the MAX parameter, the command returns the maximum address available (14647294). Be sure to place a space between the command and the parameter.

Parameters Out of Range - Set to Limit. If an out of range parameter is automatically adjusted to an acceptable value, bit 10 in the SUMMary:QUEStionable register will be set true (1). For example, if CHANnel1:PROBe 0.1 is entered, the value is set to 0.9 (lowest available setting) and bit 10 is set to "1".

Linking Commands

Linking IEEE 488.2 Common Commands with COMP Commands. Use a semicolon between the commands. For example:

```
*CLS;*RST;CAL:REP? CHAN1
```

Linking Multiple COMP Commands. Use both a semicolon and a colon between the commands. For example:

```
CHAN1:COUP AC;:TIM:RANG?
```

COMP also allows several commands within the same subsystem to be linked with a semicolon. For example:

```
CAL:SCAL:VERT;:CAL:SCAL:BCAL
```

or

```
CAL:SCAL:VERT;BCAL
```

COMP Command Reference

This section describes the Hewlett-Packard 54503A Compatible Language (COMP) commands for the Oscilloscope module. Commands are listed alphabetically by subsystem and also alphabetically within each subsystem. Command guides are printed in the top margin of each page. The left guide indicates the first command listed on that page. The right guide indicates the last command listed on that page. Where only a single command appears on a page, the left and right guides will be the same.

ACQuire

The ACQuire command subsystem is used to set up the conditions to acquire waveform data prior to executing a DIGitize command. This subsystem selects the type of data, the number of averages, the number of data points, and the completion criteria.

Note

The term "Time Buckets" is defined as — the time range divided into a specific number of horizontal time points (as defined by ACQuire:POINts command), each with an equal and fixed time associated with it.

Subsystem Syntax

```
ACQuire
    :COMPlete <complete>
    :COMPlete?
    :COUNt <count>
    :COUNt?
    :POINts <points>
    :POINts?
    :TYPe <mode>
    :TYPe?
```

:COMPlete

ACQuire:COMPlete *<complete>* specifies the completion criteria for an acquisition. Specifies what percentage of the time buckets need to be "full" before an acquisition is considered complete.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
complete	numeric	0 to 100	PCT

Example

Set acquire complete to 50%

The following example illustrates the use of the ACQuire:COMPlete command only. Chapter 3 contains an example on performing a complete digitizing operation.

```
ACQ:COMP  50          Acquire complete is 50%
```

Comments

- **Time Buckets = "full":** A time bucket is considered "full" dependent on the ACQuire:TYPe selected as follows:

  ACQuire:TYPe NORMal: The instrument only needs one data point per time bucket for that time bucket to be considered full.

  ACQuire:TYPe AVERage or ENVelope: A specified number of data points per time bucket (set using ACQuire:COUNt) must be acquired.

- **Recommended Completion Value:** 60% is the recommended completion criteria. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.
- **Completion of 0%:** If the complete value is set to 0, then one acquisition cycle will take place.
- **Related Commands:** ACQuire:TYPe, COUNt.
- ***RST Condition:** Defaults to 100%.

:COMPlete?

ACQuire:COMPlete? returns the completion value (in percent) for the currently selected mode. The value is sent to the output buffer.

Example

Querying acquire complete

```
ACQ:COMP  50          Acquire complete is 50%
ACQ:COMP?             Query instrument to return acquire
                      complete value
enter  statement      Enter value into computer
```

:COUNt

ACQuire:COUNt ***<count>*** selects the number of values to be averaged for each time bucket before the acquisition (for that time bucket) is considered complete.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
count	numeric	1 to 2048	none

Example

Set acquire count to 64

The following example illustrates the use of the ACQuire:COUNt command only. Chapter 3 contains an example on performing a complete digitizing operation.

`ACQ:COUN 64`	*Acquire count is 64*

Comments

- **Acquire Type:** Count values accepted are dependent on the ACQuire:TYPe currently selected, as follows:

  ACQuire:TYPE NORMal: Count is not used in this mode. Any value from 1 to 2048 can be entered, but is disregarded. Query will always return a 1.

  ACQuire:TYPE AVERage: When selected, the acceptable values are from 1 to 2048, however the entry will be rounded to the nearest power of 2.

  ACQuire:TYPE ENVelope: Count is not used in this mode. Any value from 1 to 2048 can be entered, but is disregarded. Query will return entered value.

- **Related Commands:** ACQuire:COMPlete, TYPe.
- ***RST Condition:** Defaults to 8.

:COUNt?

ACQuire:COUNt? returns the currently selected count value. The value is sent to the output buffer.

Example

Querying acquire count

`ACQ:COUN 64`	*Acquire count is 64*
`ACQ:COUN?`	*Query instrument to return acquire count value*
`enter statement`	*Enter value into computer*

Comments

- **Acquire Type Normal:** When ACQuire:TYPe NORMal is selected, a count query will always return a 1.

:POINts

ACQuire:POINts <*points*> selects the number of time buckets for each acquisition record.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
points	numeric	32|64|128|256|500|512|1024	none

Example

Set acquire points to 1024

The following example illustrates the use of the ACQuire:POINts command only. Chapter 3 contains an example on performing a complete digitizing operation.

```
ACQ:POIN  1024
```
Acquire points are 1024

Comments

- **Entering Points:** Any value between 32 and 1024 can be entered, however entry will be rounded to the nearest acceptable value.
- **Waveform Points:** To determine the ACTUAL number of time buckets acquired, send the WAVeform:POINts? query.
- **Related Commands:** WAVeform:POINts?.
- ***RST Condition:** Defaults to 500.

:POINts?

ACQuire:POINts? returns the currently selected points value. The value is sent to the output buffer.

Example

Querying Acquire Points

`ACQ:POIN  1024`	*Acquire points are 1024*
`ACQ:POIN?`	*Query instrument to return acquire points value*
`enter  statement`	*Enter value into computer*

:TYPe

ACQuire:TYPe ***<mode>*** selects the type of acquisition that will take place when a DIGitize command is executed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	NORMal|AVERage|ENVelope	none

Example

Set acquire type to average

The following example illustrates the use of the ACQuire:TYPe command only. Chapter 3 contains an example on performing a complete digitizing operation.

```
ACQ:TYP  AVER              Acquisition type is average
```

Comments

- **Selecting Mode:** Mode is used to select how the acquisitions are used when generating the waveform. See the WAVeform:TYPe? query for more information on selecting acquisition type.

  NORMal: Used for general purpose type measurements. The waveform reflects the last data point (hit) in each time bucket. ACQuire:COUNt has no effect in this mode.

  AVERage: Used when reduction of signal noise and improved resolution is desired. The waveform reflects a minimum of *n* acquisitions averaged per time bucket, where *n* is the current ACQuire:COUNt value.

  ENVelope: Used when measuring voltage or time jitter. The waveform reflects the minimum and maximum data points (hit) in each time bucket. ACQuire:COUNt has no effect in this mode.
- **Related Commands:** ACQuire:COUNt, DIGitize, WAVeform:TYPe?.
- ***RST Condition:** Defaults to NORMal.

:TYPe?

ACQuire:TYPe? returns the currently selected acquisition type. The data is sent to the output buffer. See ACQuire:TYPe command for more information on available types.

Example

Querying acquire type

```
Dimension  statement   String for data
ACQ:TYP  AVER          Acquisition type is average
ACQ:TYP?               Query instrument to return
                       acquisition type
enter  statement       Enter value into computer
```

CALibrate

The CALibrate command subsystem contains commands to perform probe/self calibrations, and set channel-to-channel time nulls.

Subsystem Syntax

```
CALibrate
    :PCALibration
        :ATTenuation  CHANnel<number>
        :BCALibration
        :TNULl
            :CH1TO<number> <time>
    :REPort? <channel>
    :SCALibration
        :BCALibration
        :DCALibration
        :DELay <channel>
        :DOUTput <level>
        :LTCalibration
        :TNULl <channel_skew>
        :VERTical
    :TNULl <value1>,<value2>,<value3>
    :TNULl?
```

:PCALibration:BCALibration

CALibrate:PCALibration:BCALibration performs an attenuation calibration on the channel number specified by the CAL:PCAL:ATT CHAN command. Instrument calibrates channel gain at the point connected to the DC Calibrator Output connector (probe, cable, etc). Probe attenuation is then calculated from the results, and a correction is automatically entered in the correct CHANnel<*n*>:PROBe setting.

Example **Perform attenuation calibration on channel 4**

This example calibrates the channel gain on input 4. For the example, a 10:1 attenuator probe is connected to the DC Calibrator Output connector from the Input 4 connector.

`CAL:PCAL:ATT  CHAN4`	*Attenuation calibration channel 4*
`pause`	*To connect probe to DC Calibrator Output from Input 4 connector*
`CAL:PCAL:BCAL`	*Perform attenuation calibration. Correction automatically stored in CHAN4:PROB*

Comments

- **Valid Calibration:** Channel gain is corrected using calculated probe attenuation values from 0.9:1 to 250:1. If the measured results cause the calculated attenuation factor to be out of this range, an error will be generated.
- **Related Commands:** CAL:PCAL:ATT CHANnel, CHANnel<n>:PROBe.

:PCALibration:ATTenuation CHANnel

CALibrate:PCALibration:ATTenuation CHANnel<*number*> selects the channel number that will be calibrated when the CAL:PCAL:BCAL command is executed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example **Set attenuation calibration channel to 4**

```
CAL:PCAL:ATT   CHAN4
```
Attenuation calibration channel to 4

Comments

- **Related Commands:** CAL:PCAL:BCALibration.

:PCALibration:TNULl:CH1TO

CALibrate:PCALibration:TNULl:CH1TO<*number*> <*time*> is used to set the timing of channels 2, 3, <u>**OR**</u> 4 to correspond with channel 1. Use to eliminate any time discrepancies between channels and minimize channel to channel skew variations. Use to manually adjust any differences in cable length.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number *time*	numeric numeric	2 to 4 -50NS to +70NS	none S

Example **Set time null from channel 1 to 3 to 25 ns.**

```
CAL:PCAL:TNUL:CH1TO3   25E-9
```
Channel 1 to 3 time null to 25 nsec

Comments

- **Query Time Null:** Use the CALibrate:TNULl? query to return current time null settings.
- **CALibrate:TNULl Command:** This command is similar to the CALibrate:TNULl? query, except the three time null values can be entered separately.
- **Related Commands:** CALibrate:TNULl.

:REPort?

CALibrate:REPort? *<channel>* is used to query the current calibration status of the instrument. Each channel's status is queried separately. The data is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel	numeric	CHANnel*n* (*n*=1 to 4)	none

Example

Query channel 2 calibration results

```
dimension  statement     String to hold data
CAL:REPort?  CHAN2       Query channel 2 calibration results
enter  statement         Enter value into computer
```

Comments

- **Returned Format:** The calibration results are returned in the following format:

  CHANnel1 **A/D X, Gain X, Offset X, Hysteresis X, Trigger X, Delay X, Logic Trigger X**

  CHANnel2-4 **A/D X, Gain X, Offset X, Hysteresis X, Trigger X, Delay X, Time Null X**

  Where X is "P"=Passed, "F"=Failed, "D"=Defaulted, "C"=Corrupted. If X prefixed by a "*", indicates a new ROM revision without a recalibration.

- **Related Commands:** CAL:SCAL:VERTical, CAL:SCAL:DCALibration, CAL:SCAL:DELay, CAL:SCAL:LTCalibration, CAL:SCAL:TNULl.

:SCALibration:BCALibration

CALibrate:SCALibration:BCALibration is used to begin a self calibration routine. The routine that is performed is dependent on the SCALibration command configured prior to executing the BCALibration command.

Example

Begin a Logic Trigger Calibration

```
CAL:SCAL:LTC             Configure logic trigger calibration
CAL:SCAL:BCAL            Begin logic trigger calibration
```

Comments

- **Self Calibration:** If the BCALibration command is executed without first defining the SCALibration routine to be performed, a bit will be set in the SUMMary:QUEStionable register.
- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a SCALibration routine.
- **Related Commands:** CAL:SCAL:VERTical, CAL:SCAL:DCALibration, CAL:SCAL:DELay, CAL:SCAL:LTCalibration, CAL:SCAL:TNULl, CAL:TNULl, SUMMary:QUEStionable.

:SCALibration:DCALibration

CALibrate:SCALibration:DCALibration is used to load "default" calibration data. Default calibration data is set at the factory and is dependent on the ROM revision currently installed. This command should only be used by service personnel. Procedures for performing this calibration are provided in the Service Manual.

Example

Overwrite all existing calibration data with default calibration data

`CAL:SCAL:DCAL`	*Configure for default calibration routine*
`CAL:SCAL:BCAL`	*Load default calibration data*

Comments

- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a default calibration routine.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?.

:SCALibration:DELay

CALibrate:SCALibration:DELay *<channel>* performs a delay calibration on all four inputs, one at a time. Each input must be connected to the AC Calibrator Output prior to executing the calibration routine for that channel. The results are stored and used by the instrument to maintain measurement accuracy.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel	numeric	CHANnel*n* (*n*=1 to 4)	none

Example

Chapter 3 contains an example on performing a delay calibration

Comments

- **Calibration Results:** Delay calibration results can be reviewed using the Calibrate:REPort? query.
- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a calibration routine.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?.

:SCALibration:DOUTput

CALibrate:SCALibration:DOUTput ***<level>*** is used to set the output level of the DC Calibrator output connector to 0 volts (ZVOLt) or 5 volts (FVOLt).

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
level	discrete	ZVOLt|FVOLt	none

Example **Set DC calibrator output connector to 5V**

```
CAL:SCAL:DOUT  5        DC calibrator output to 5V
```

Comments

- ***RST Condition:** Defaults to ZVOLt (0 volts).

:SCALibration:LTCalibration

CALibrate:SCALibration:LTCalibration performs a logic trigger calibration. Input 1 must be connected to the AC Calibrator Output prior to executing the calibration routine. The results are stored and used by the instrument to maintain measurement accuracy.

Example **Chapter 3 contains an example on performing a logic trigger calibration**

Comments

- **Prior to Logic Trigger Calibration Execution:** Prior to executing the logic trigger calibration routine, the calibration results must be reviewed using the CALibrate:REPort? query. All four channel calibration results must indicate "P" before the logic trigger calibration can be executed.
- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a calibration routine.
- **Calibration Results:** Logic trigger calibration results can be reviewed using the CALibrate:REPort? query.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?.

:SCALibration:TNULl

CALibrate:SCALibration:TNULl *<channel_skew>* performs a time null calibration on one set of channels at a time. The results are stored and used by the instrument to maintain measurement accuracy.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_skew	discrete	CH1TO2|CH1TO3|CH1TO4	none

Example

Chapter 3 contains an example on performing a time null calibration

Comments

- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a calibration routine.
- **Calibration Results:** Time null calibration results can be reviewed using the CALibrate:REPort? query.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?.

:SCALibration:VERTical

CALibrate:SCALibration:VERTical performs a vertical calibration on all four inputs simultaneously. All inputs must be connected to the DC Calibrator Output prior to executing the calibration routine. The results are stored and used by the instrument to maintain measurement accuracy.

Example

Chapter 3 contains an example on performing a vertical calibration

Comments

- **Calibration Results:** Vertical calibration results can be reviewed using the CALibrate:REPort? query.
- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a calibration routine.
- **Related Commands:** Calibrate:SCALibration:BCALibration, Calibrate:REPort?.

:TNULl

CALibrate:TNULl *<value_1>,<value_2>,<value_3>* is used to set the timing of channels 2, 3, AND 4 to correspond with channel 1. Use to eliminate any time discrepancies between channels and minimize channel to channel skew variations. Use to manually adjust any differences in cable length.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
value_1	numeric	–50NS to +70NS	S
value_2	numeric	–50NS to +70NS	S
value_3	numeric	–50NS to +70NS	S

Example **Set time null to 25 ns from channel 1 to 4**

`CAL:TNUL 0,0,25E-9` *Channel 1 to 4 time null to 25 nsec*

Comments

- **Query Time Null:** Use the CALibrate:TNULl? query to return current time null settings.
- **CALibrate:TNULl Command:** This command is similar to the CALibrate:PCALibration:ATTenuation:TNULl CH1TO command, except all three time null values must be entered.
- **Related Commands:** CAL:PCAL:TNULl CH1TO.

:TNULl?

CALibrate:TNULl? returns the currently selected time nulls (in seconds) for channels 1 to 2, 1 to 3, and 1 to 4 respectively. The data is sent to the output buffer. See CALibrate:TNULl command for more information.

Example **Querying time nulls**

`CAL:TNUL 0,0,25E-9` *Set channel 1 to 4 time null to 25 nsec*

`CAL:TNUL?` *Query instrument to return time nulls*

`enter statement` *Enter value into computer*

Comments

- **Related Commands:** CAL:SCAL:TNULl CH1TO, CAL:TNULl.

CHANnel

The CHANnel command subsystem is used to select a specific channel's vertical or Y-axis controls. Channels 1, 2, 3, and 4 are independently programmable for all offset, probe, coupling, and range functions.

See VIEW and BLANk commands for information on channel presentation.

Subsystem Syntax

```
CHANnel<number>
    :COUPling <type>
    :COUPling?
    :ECL
    :HFReject <mode>
    :HFReject?
    :LFReject <mode>
    :LFReject?
    :OFFSet <value>
    :OFFSet?
    :PROBe <atten>
    :PROBe?
    :RANGe <range>
    :RANGe?
    :TTL
```

:COUPling

CHANnel<*number*>:COUPling <*type*> is used to select the input coupling for the channel specified. The coupling for each channel can be set to AC, DC or DCFifty.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	None
type	discrete	AC|DC|DCFifty	None

Example

Set channel 1 coupling to AC

```
CHAN1:COUP  AC            Channel 1 coupling to AC
```

Comments

- **Impedance:** AC is 1MΩ, DC is 1MΩ, and DCFifty is 50Ω.
- ***RST Condition:** Defaults to DC on all channels.
- **Related Commands:** CHANnel<*n*>:COUPling?

:COUPling?

CHANnel<*number*>:COUPling? returns the currently selected coupling type for the channel specified. The data is sent to the output buffer. See CHANnel<*n*>:COUPling for more information on coupling types.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example

Querying channel 1 coupling

`Dimension statement`	*String for data*
`CHAN1:COUP AC`	*Channel 1 coupling to AC*
`CHAN1:COUP?`	*Query instrument to return channel 1 coupling selection*
`enter statement`	*Enter value into computer*

:ECL

CHANnel<*number*>:ECL sets the specified channel's vertical controls for optimum viewing of ECL signals. CHANnel<*n*>:RANGe is set to 1.6 volts full scale, CHANnel<*n*>:OFFSet and TRIGger:LEVel are set to –1.3 volts, and CHANnel<*n*>:COUPling is set to DC (1MΩ or 50Ω). TRIGger:SLOPe and impedance (coupling) is not changed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example

Set channel 2 to view ECL signal

`CHAN2:ECL`	*Channel 2 for ECL signal*

Comments

- **Probe Attenuation:** The current CHANnel<n>:PROBe setting (other than 1:1) will affect CHANnel<n>:RANGe and OFFSet settings.
- **Related Commands:** CHANnel<*n*>:OFFSet, LEVel, RANGe, COUPling.

:HFReject

CHANnel<*number*>:HFReject <*mode*> is used to select an internal low pass filter to reject high frequencies. When ON, the bandwidth of the specified channel is limited to approximately 30MHz. The bandwidth limit filter may be used with all coupling selections.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none
mode	boolean	OFF|0|ON|1	none

Example **Enable low pass filter on channel 1**

```
CHAN1:HFR 1
```

Channel 1 low pass filter to on

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Related Commands:** CHANnel<*n*>:COUPling.
- ***RST Conditions:** Defaults to OFF for all channels.

:HFReject?

CHANnel<*number*>:HFReject? returns a number to show whether the internal low pass filter is enabled or disabled for the specified channel. "1" = ON, "0" is OFF. The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example **Querying channel 1 low pass filter state**

`CHAN1:HFR 1`	*Channel 1 low pass filter to on*
`CHAN1:HFR?`	*Query instrument to return channel 1 low pass filter state*
`enter statement`	*Enter value into computer*

:LFReject

CHANnel<*number*>:LFReject <*mode*> is used to select an internal high pass filter to reject low frequencies. When ON, the bandwidth of the specified channel is limited to approximately 450 Hz. The bandwidth limit filter may be used only with AC coupling.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none
mode	boolean	OFF|0|ON|1	none

Example

Enable filter on channel 3

```
CHAN3:LFR 1                Channel 3 filter to on
```

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Related Commands:** CHANnel<*n*>:COUPling.
- ***RST Conditions:** Defaults to OFF for all channels.

:LFReject?

CHANnel<*number*>:LFReject? returns a number to show whether the internal filter is enabled or disabled for the specified channel. "1" = ON, "0" is OFF. The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example

Querying channel 3 high pass filter state

```
CHAN3:LFR 1                Channel 3 filter to on
CHAN3:LFR?                 Query instrument to return channel 3
                           filter state
enter statement            Enter value into computer
```

:OFFSet

CHANnel<*number*>:OFFSet <*value*> sets the voltage that is represented at the center of the current range for the selected channel number.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number *value*	numeric numeric	1 to 4 Depends on CHANnel<n>:RANGe	none V

Example

Set channel 2 offset to 10V

```
CHAN2:OFFS  10          Channel 2 offset to 10 volts
```

Comments

- **Entering Offset:** The range of acceptable OFFSet values is dependent on the current CHANnel<*n*>:RANGe setting as follows:

CHANnel<n>:RANGe	CHANnel<n>:OFFSet limits
8mV to 400mV	±2V
>400mV to 2.0V	±10V
>2.0V to 10.0V	±50V
>10.0V to 40.0V	±250V

  If OFFSet is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.

- **Probe Attenuation:** Changing CHANnel<n>:PROBe settings after selecting CHANnel<n>:OFFSet will cause the offset parameter to change.
- **Related Commands:** CHANnel<*n*>:RANGe, PROBe.
- ***RST Condition:** Defaults to 0 volts.

:OFFSet?

CHANnel<*number*>:OFFSet? returns the current offset value for the channel number specified. The value (in volts) is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example

Querying channel 2 offset value

```
CHAN2:OFFS  10          Channel 2 offset to 10 volts
CHAN2:OFFS?             Query instrument to return channel 2
                        offset value in volts
enter  statement        Enter value into computer
```

:PROBe

CHANnel<*number*>:PROBe <*atten*> is used to enter a probe's attenuation factor for the channel specified. The selection does not change the actual input sensitivity of the instrument, it changes the reference constants for scaling the vertical range and offset, automatic measurements, trigger levels, etc.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none
atten	numeric	0.9 to 1000.0	none

Example

Set channel 1 probe attenuation to 10:1

```
CHAN1:PROB  10          Channel 1 probe attenuation to 10:1
```

Comments

- **Entering Attenuation:** If *atten* is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.
- **Range and Offset:** Changing CHANnel<n>:PROBe will effect the current settings of CHANnel<n>:RANGe and OFFSet.
- **Related Commands:** CALibrate:PCALibration:ATTenuation, CHANnel<n>:RANGe, OFFSet.
- ***RST Conditions:** Defaults to 1:1 on all channels.

:PROBe?

CHANnel<*number*>:PROBe? returns the current probe attenuation factor for the channel specified. The value (a ratio :1) is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example

Querying channel 1 probe attenuation

```
CHAN1:PROB  10          Channel 1 probe attenuation to 10:1
CHAN1:PROB?             Query instrument to return channel 1
                        probe attenuation factor
enter  statement        Enter value into computer
```

:RANGe

CHANnel<*number*>:RANGe <*range*> is used to define the full scale vertical axis, or "Y-axis" of the channel specified.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none
range	numeric	8MV to 40.0V	V

Example

Set channel 2 range to 10 volts (full scale)

```
CHAN2:RANG  10
```
Channel 2 range to 10 volts

Comments

- **Range versus Probe Attenuation:** Range values can be set from 8mV to 40.0 V when CHANnel<*n*>:PROBe is set to 1:1. If the CHANnel<*n*>:PROBe value is changed, the CHANnel<*n*>:RANGe value is multiplied by the probe attenuation factor.
- **Range versus Trigger Level:** Changing CHANnel<n>:RANGe could affect trigger level currently selected.
- **Probe Attenuation:** Changing CHANnel<n>:PROBe settings after selecting CHANnel<n>:RANGe will cause the range parameter to change.
- **Related Commands:** CHANnel<*n*>:PROBe, OFFSet.
- ***RST Condition:** Defaults to 4 volts on all channels.

:RANGe?

CHANnel<*number*>:RANGe? returns the current range setting for the channel specified. The value (in volts) is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example

Querying channel 2 full scale range setting

```
CHAN2:RANG  10
CHAN2:RANG?
enter  statement
```

Channel 2 range to 10 volts

Query instrument to return channel 2 range setting

Enter value into computer

:TTL

CHANnel<*number*>:TTL sets the specified channels vertical controls for optimum viewing of TTL signals. CHANnel<*n*>:RANGe is set to 8.0 volts full scale, CHANnel<*n*>:OFFSet is set to 2.5 volts, TRIGger:LEVel is set to 1.4 volts, and CHANnel<*n*>:COUPling is set to DC (1MΩ or 50Ω). TRIGger:SLOPe and impedance (coupling) is not changed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example

Set channel 2 to view TTL signal

```
CHAN2:TTL                  Channel 2 for TTL signal
```

Comments

- **Probe Attenuation:** The current CHANnel<n>:PROBe setting (other than 1:1) will effect CHANnel<n>:RANGe and OFFSet settings.
- **Related Commands:** CHANnel<*n*>:RANGe, OFFSet, LEVel, COUPling.

DISPlay

The DISPlay command subsystem is used to control the display of data in pixel memory.

Subsystem Syntax

```
DISPlay
    :DATa <block>
    :DATa?
    :PERSistence <period>
    :PERSistence?
    :SOURce <source>
    :SOURce?
```

:DATa

DISPlay:DATa ***<block>*** is used to write a block of binary waveform data from the bus to one of the pixel memory locations. Only Pixel MEMories 1 and 2 may be written to, and are selected using the DISPlay:SOURce command.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
block	block	binary block data	none

Example

Write waveform data to pixel memory 1

In this example, the block data is represented by "#516576..."

`DISP:SOUR PMEM1`	*Select pixel memory 1*
`DISP:DAT #516576...`	*Write binary waveform data*

Comments

- **Block Data:** The data is in the IEEE 488.2 definite block form with 16576 bytes of data preceded by seven block header bytes. The block header contains the ASCII characters "#516576" and is sent prior to the data being sent.
- **Display Data versus Waveform Data:** The DISPlay and WAVeform DATa commands are similar, except that DISPlay:DATa transfers data to the pixel memories and WAVeform:DATa transfers data to the Waveform memories.
- **Related Commands:** DISPlay:DATa?, SOURce.

:DATa?

DISPlay:DATa? is used to write a block of binary waveform data from one of the pixel memory locations to the output buffer. Pixel MEMories 0, 1, or 2 may be written from, and are selected using the DISPlay:SOURce command. See DISPlay:DATa command for more information on waveform data.

Example **Transfer active waveform to pixel memory 1**

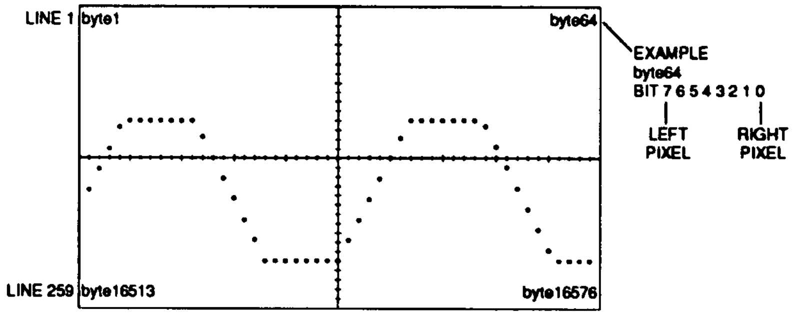
In this example, the retrieved block data is represented by "Xxxxx". See Chapter 3 Digitize example, for the procedure used to read the block length then re-define a string to hold the data.

`dimension statement`	*String to hold data (16576 bytes)*
`VIEW PMEM0`	*Enables pixel memory 0*
`DISP:SOUR PMEM0`	*Select pixel memory 0*
`DISP:DAT?`	*Read binary waveform data from pixel memory 0 (active waveform)*
`enter statement`	*Enter value into computer (see Chapter 3, Digitize example, for the procedure to read block length)*
`DISP:SOUR PMEM1`	*Select pixel memory 1*
`DISP:DAT Xxxxx`	*Write binary waveform data to pixel memory 1*

Comments

- **Pixel Memories 0-2:** PMEM0 represents the active display. PMEM1 and PMEM2 contain data entered using the DISPlay:DATa and MERGe commands. PMEM source is specified using the DISPlay:SOURce command.
- **Display Data versus Waveform Data:** The DISPlay and WAVeform DATa? commands are similar, except that DISPlay:DATa? transfers data from the pixel memories and WAVeform:DATa? transfers data from the waveform memories, channel buffers, or function buffers.
- **Pixel Data Plotting:** The active display or pixel memory 0 can be plotted using the DISPlay:DATa? query. See the example on the next page for more information.
- **Related Commands:** MERGe, DISPlay:DATa, SOURce.

Example The following example shows how to plot pixel memory 0-2. Pixel memory 0 is the active waveform, and pixel memory 1 and 2 are volatile storage locations for waveform data. The pixel data is contained in a total of 16,576 bytes. There are 259 lines of data, and each line contains 64 bytes. Each byte consists of 8 pixels, where bit 7 is the left pixel and bit 0 is the right pixel. Bytes are counted top to bottom, left to right. See the illustration below for more information.



The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10  INTEGER Pixel(1:16576)
 20  INTEGER Pixels
 30  S=70901
 40  OUTPUT S;"*CLS"
 50  OUTPUT S;"DIG CHAN1"
 60  OUTPUT S;"SYST:HEAD OFF"
 70  OUTPUT S;"VIEW PMEM0"
 80  OUTPUT S;"DISP:SOUR PMEM0"
 90  OUTPUT S;"DISP:DAT?"
100  ENTER S USING "#,1A";Pound$
110  ENTER S USING "#,1D";Number
120  ENTER S USING "#,"&VAL$(Number)&"D";Bytes
130  ENTER S USING "#,B";Pixel(*)
140  ENTER S USING "B";Crif
150  GRAPHICS ON
160  GINIT
170  PLOTTER IS CRT, "INTERNAL"
180  GCLEAR
190  VIEWPORT 0,RATIO*100,0,100
200  WINDOW 0,512,260,0
210  Position=0
220  Pline=0
230  Ppos=1
240  PEN -1
250  FOR K=1 to 259
260    FOR I=1 TO 64
270      IF Pixel(Ppos)<>0 THEN
280        FOR J=7 TO 0 STEP -1
290          IF BIT(Pixel(Ppos),J) THEN
300            MOVE Position,Pline
310            PEN 1
320            PLOT Position,Pline
330            PEN -1
340          END IF
350          Position=Position+1
360        NEXT J
370      ELSE
390        Position=Position+8
400      END IF
410      Ppos=Ppos+1
420    NEXT I
430    Pline=Pline+1
440    Position=0
450  NEXT K
460  END
```

:PERSIstence

DISPlay:PERSistence ***<period>*** sets the display persistence for pixel memory 0 (active display).

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
period	discrete	INFinite|SINGle	none

Example **Set display persistence to infinite**

```
DISP:PERS  INF          Persistence to infinite
```

Comments

- **Selecting Period:** When SINGle is selected, the contents of pixel memory 0 are updated to reflect the last data point hit in a time bucket. When INFinite is selected, all the contents of pixel memory 0 are retained and current waveform data is added.
- ***RST Condition:** Defaults to SINGle.

:PERSistence?

DISPlay:PERSistence? returns the current display persistence setting. The data (SINGle or INFinite) is sent to the output buffer.

Example **Query display persistence**

```
DISP:PERS  INF          Persistence to infinite
DISP:PERS?              Query instrument to return display
                        persistence setting
enter  statement        Enter value into computer
```

:SOURce

DISPlay:SOURce ***<source>*** is used to select the pixel memory location for the DISPlay:DATa query and command.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	PMEMory*n* (*n*=0 to 2)	none

Example

Set pixel memory 2 as the source

```
DISP:SOUR  PMEM2
```
Source is pixel memory 2

Comments

- **Selecting Source:** PMEMory0 represents the active display. PMEMory1 and PMEMory2 are volatile memory locations. Specifies the location to store pixel data using the DISPlay:DATa command, or retrieve pixel data using the DISPlay:DATa? query.
- **Related Commands:** DISPlay:DATa, DATa?, SOURce?.
- ***RST Condition:** Defaults to PMEMory0.

:SOURce?

DISPlay:SOURce? returns the current display source setting. The setting (PMEM0, PMEM1, or PMEM2) is sent to the output buffer. See DISPlay:SOURce command for more information.

Example

Query display source

```
DISP:SOUR  PMEM2
```
Source is pixel memory 2

```
DISP:SOUR?
```
Query instrument to return display source setting

```
enter  statement
```
Enter value into computer

FUNCtion

The FUNCtion command subsystem defines six functions that use signals acquired on CHANnels 1 to 4 and/or stored in Waveform MEMories 1 to 4 as operands to create altered or duplicate waveforms. The selected CHANnel<*n*> or WMEMory<*n*> is enabled when defined as an operand. Two locations are provided for the results (FUNCtion1 and 2).

Subsystem Syntax

FUNCtion<*number*>
  :ADD <*source*>,<*source*>
  :INVert <*source*>
  :MULTiply <*source*>,<*source*>
  :OFFSet <*value*>
  :OFFSet?
  :ONLY <*source*>
  :RANGe <*range*>
  :RANGe?
  :SUBTract <*source*>,<*source*>

:ADD

FUNCtion<*number*>:ADD <*source*>,<*source*> is used to algebraically sum two defined operands. Results are retained in the FUNCtion *number* specified.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 or 2	none
source	discrete	CHANnel*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)	none

Example

Algebraically sum channel 1 with waveform memory 4 and retain results in function 1

```
FUNC1:ADD  CHAN1,WMEM4 Add channel 1 to waveform
                       memory 4, retain as function 1
```

Comments

- **FUNCtion Number:** Used to specify where function results are retained. Range and offset are automatically selected to accommodate the resulting waveform. Resulting waveform can be measured, turned on/off, stored, and/or viewed.
- **Related Commands:** FUNCtion<*n*>:OFFSet, RANGe, MEASure subsystem, BLANk, STORe, VIEW.
- ***RST Condition:** FUNCtion 1 and 2 default to ADD CHANnel1 + CHANnel1.

:INVert

FUNCtion<*number*>:INVert *<source>* is used to invert the defined operand. Result is retained in the FUNCtion *number* specified.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 or 2	none
source	discrete	CHANNel*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)	none

Example

Invert channel 1 and retain results in function 2

```
FUNC2:INV  CHAN1
```

Invert channel 1 waveform, retain in function 2

Comments

- **FUNCtion Number:** Used to specify where function result is retained. Range and offset are automatically selected to accommodate the resulting waveform. Resulting waveform can be measured, turned on/off, stored, and/or viewed.
- **Related Commands:** FUNCtion<*n*>:OFFSet, RANGe, MEASure subsystem, BLANk, STORe, VIEW.

:MULTiply

FUNCtion<*number*>:MULTiply *<source>,<source>* is used to algebraically multiply two defined operands. Results are retained in FUNCtion number specified.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 or 2	none
source	discrete	CHANNel*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)	none

Example

Algebraically multiply channel 1 with waveform memory 4 and retain results in function 1

```
FUNC1:MULT  CHAN1,WMEM4
```

Multiply channel 1 to waveform memory 4, retain as function 1

Comments

- **FUNCtion Number:** Used to specify where function results are retained. Range and offset are automatically selected to accommodate the resulting waveform. Resulting waveform can be measured, turned on/off, stored, and/or viewed.
- **Related Commands:** FUNCtion<*n*>:OFFSet, RANGe, MEASure subsystem, BLANk, STORe, VIEW.

:OFFSet

FUNCtion<*number*>:OFFSet <*value*> sets the voltage that is represented at the center of the current range for the selected function number. Useful in scaling function 1 and 2 results.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 or 2	none
value	numeric	Depends on FUNCtion<*n*>:RANGe	V

Example

Set function 2 offset to 10V

```
FUNC2:OFFS  10            Function 2 offset to 10 volts
```

Comments

- **Entering Offset:** The maximum range of acceptable OFFSet values is ± the current FUNCtion<*n*>:RANGe setting. If OFFSet is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.

  When to Enter Offset: Because offset is automatically selected to accommodate the resulting waveform (depending on the function selected), offset values must be specified **AFTER** the function result is retained.
- **Related Commands:** FUNCtion<*n*>:RANGe.
- ***RST Condition:** Defaults to 0 volts on both functions.

:OFFSet?

FUNCtion<*number*>:OFFSet? returns the current offset value for the function number specified. The value (in volts) is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 or 2	none

Example

Querying function 2 offset value

```
FUNC2:OFFS  10            Function 2 offset to 10 volts
FUNC:OFFS?                Query instrument to return function 2
                          offset value in volts
enter  statement          Enter value into computer
```

:ONLY

FUNCtion<*number*>:ONLY <*source*> is used to copy the defined operand. Result is retained in the FUNCtion number specified.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number *source*	numeric discrete	1 or 2 CHANnel*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)	none none

Example

Duplicate channel 1 in function 2

```
FUNC2:ONLY   CHAN1          Copy channel 1 waveform, retain in
                            function 2
```

Comments

- **FUNCtion Number:** Used to specify where function results are retained. Range and offset are automatically selected to accommodate the resulting waveform. Resulting waveform can be measured, turned on/off, stored, and/or viewed.
- **Scaling:** Use the ONLY command to duplicate a channel or memory waveform, then use the FUNCtion<*n*>:RANGe and OFFSet commands to scale the function waveform.
- **Related Commands:** FUNCtion<*n*>:OFFSet, RANGe, MEASure subsystem BLANk, STORe, VIEW.

:RANGe

FUNCtion<*number*>:RANGe <*range*> is used to define the full scale vertical axis of the function number specified. Useful in scaling function 1 and 2 results.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number *range*	numeric numeric	1 or 2 Depends on CHANnel<n>:RANGe OFFSet, and PROBe	none V

Example

Set function 2 range to 10 volts (full scale)

```
FUNC2:RANG   10          Function 2 range to 10 volts
```

Comments

- **Entering Range:** Because range is automatically selected to accommodate the resulting waveform (depending on the function selected), range values must be specified **AFTER** the function result is retained.

  Range is automatically adjusted when the operands and/or functions are changed from the default settings (CHAN1 + CHAN1). If the default function is used, and then turned on using the VIEW command, the range will **NOT** be adjusted.

- **Related Commands:** VIEW, FUNCtion<*n*>:OFFSet.
- ***RST Condition:** Defaults to 4 volts on both functions.

:RANGe?

FUNCtion<*number*>:RANGe? returns the current range setting for the function number specified. The value (in volts) is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 or 2	none

Example

Querying function 2 full scale range setting

`FUNC2:RANG 10`	*Function 2 range to 10 volts*
`FUNC2:RANG?`	*Query instrument to return function 2 range setting*
`enter statement`	*Enter value into computer*

:SUBTract

FUNCtion<*number*>:SUBTract <*source*>,<*source*> is used to algebraically subtract two defined operands. Results are retained in the FUNCtion number specified.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number *source*	numeric discrete	1 or 2 CHANnel*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)	none none

Example

Algebraically subtract channel 4 with waveform memory 1 and retain the results in function 2

`FUNC2:SUBT CHAN4,WMEM1`	*Subtract channel 4 from waveform memory 1, retain as function 2*

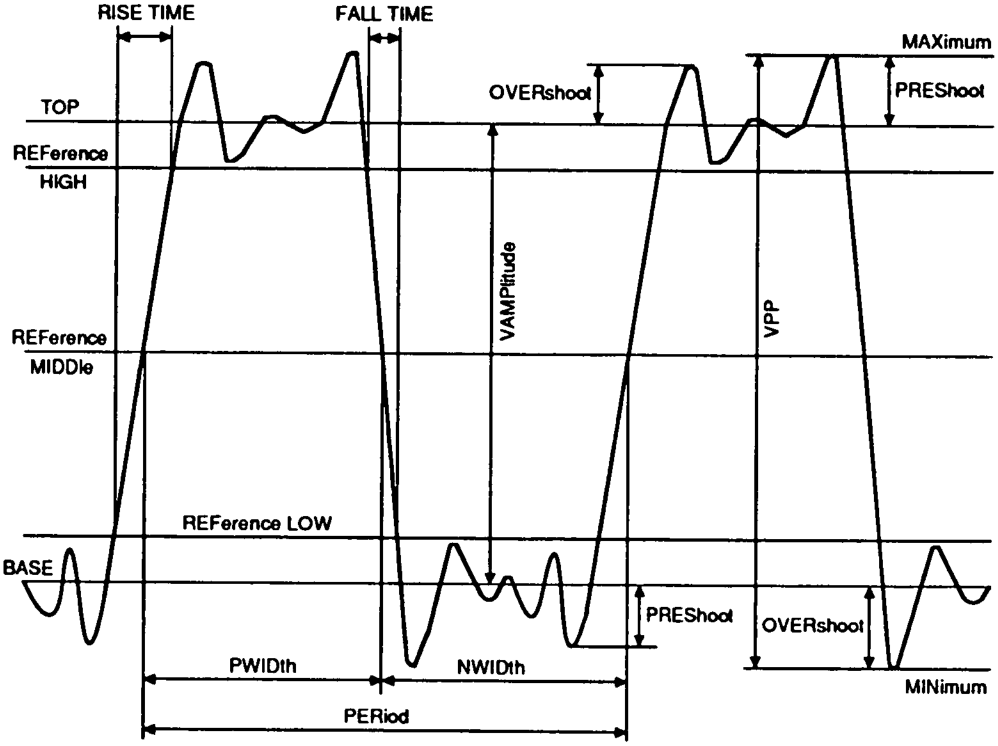
Comments

- **FUNCtion Number:** Used to specify where function results are retained. Range and offset are automatically selected to accommodate the resulting waveform. Resulting waveform can be measured, turned on/off, stored, and/or viewed.
- **Related Commands:** FUNCtion<*n*>:OFFSet, RANGe, MEASure, BLANk, STORe, VIEW.

MEASure

The MEASure command subsystem is used to make parametric measurements on the specified source, and to return the measured voltage and time values. Measurement results (up to eight) are retained in a measurement queue. Voltage, time, and event markers are automatically positioned during measurement, or can be manually set to specific voltages, times, or events.

When TIMebase:WINDow is ON, measurements are **ONLY** applied to the expanded portion of the waveform.



Subsystem Syntax

```
MEASure
    :ALL?
    :COMPare <measurement>,<upper_limit>,<lower_limit>
    :COMPare?
    :CURSor? <type>
    :DEFine <measure_spec>
    :DEFine? <measure_spec>
    :DELay
    :DELay?
    :DESTination <location>
    :DESTination?
    :DUTycycle
    :DUTycycle?
    :ESTArt <edge>
    :ESTArt?
    :ESTOp <edge>
    :ESTOp?
    :FALLtime
    :FALLtime?
```

Subsystem Syntax

```
MEASure — Continued
    :FREQuency
    :FREQuency?
    :LIMittest <mode>
    :LOWer <value>
    :LOWer?
    :MODe <mode>
    :MODe?
    :NWIDth
    :NWIDth?
    :OVERshoot
    :OVERshoot?
    :PERiod
    :PERiod?
    :POSTfailure <mode>
    :POSTfailure?
    :PRECision <coarse>
    :PRECision?
    :PREShoot
    :PREShoot?
    :PWIDth
    :PWIDth?
    :RESults?
    :RISetime
    :RISetime?
    :SCRatch
    :SOURce <source>[,<source>]
    :SOURce?
    :STATistics <mode>
    :STATistics?
    :TDELta?
    :TMAX?
    :TMIN?
    :TSTArt <time>
    :TSTArt?
    :TSTOp <time>
    :TSTOp?
    :TVOLt? <voltage>,<slope><occurrence>
    :UNITs <unit>
    :UNITs?
    :UPPer <value>
    :UPPer?
    :VACRms
    :VACRms?
    :VAMPlitude
    :VAMPlitude?
    :VAVerage
    :VAVerage?
    :VBASe
    :VBASe?
    :VDCRms
    :VDCRms?
```

Subsystem Syntax

```
MEASure — Continued
    :VDELta?
    :VFIFty
    :VMAX
    :VMAX?
    :VMIN
    :VMIN?
    :VPP
    :VPP?
    :VRELative <percent>
    :VRELative?
    :VRMS
    :VRMS?
    :VSTArt <voltage>
    :VSTArt?
    :VSTOp <voltage>
    :VSTOp?
    :VTIMe? <time>
    :VTOP
    :VTOP?
```

:ALL?

MEASure:ALL? makes a set of measurements on the present signal and sends the measurement results to the output buffer. The following measurements are performed:

- :FREQuency
- :PERiod
- :PWIDth
- :NWIDth
- :RISetime
- :FALLtime
- :VAMPlitude
- :VPP
- :PREShoot
- :OVERshoot
- :DUTycycle
- :VACRMS
- :VMAX
- :VMIN
- :VTOP
- :VBASe
- :VAVerage
- :VDCRMS

Example **Perform measure all on channel 3 and return results**

`dimension statement`	*String to hold data*
`MEAS:SOUR CHAN3`	*Select channel 3 for measurement*
`MEAS:ALL?`	*Perform measure all, query instrument to return results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** When performing measure all, a minimum of one full cycle must be present.
- **Returned Format:** The measurement results are returned as follows:

```
[MEASure:FREQuency] <result>;[PERiod] <result>;[PWIDth]
<result>;[NWIDth] <result> ;[RISetime] <result> ;[FALLtime]
<result> ;[VAMPlitude] <result> ;[VPP] <result> ;[PREShoot]
<result> ;[OVERshoot] <result> ;[DUTycycle] <result>
;[VACRms] <result> ;[VMAX] <result> ;[VMIN] <result> ;[VTOP]
<result> ;[VBASe] <result> ;[VAVerage] <result> ;[VDCRms]
<result>
```

Where: *<result>* :: = individual measurement results
[MEASure:XXXX] is returned if SYSTem:HEADer is ON

The measurement values can be returned to numeric variables instead of the string variables as shown. If numeric variables are used, SYSTem:HEADer must be turned OFF.

- **Individual MEASure Commands:** Refer to the individual commands for information on how the measurements are made and the returned format of the measurement results. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

:COMPare

MEASure:COMPare ***<measurement>,<upper_limit>,<lower_limit>*** is used to configure for a measurement comparison or limit test. When configuring a limit test, the user specifies both the desired measurement and acceptable limits of the test.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
measurement	discrete	RISetime|FALLtime|FREQuency| PERiod|PWIDth|NWIDth| VAMPlitude|VBASe|VTOP|VPP| VAVerage|VMAX|VMIN|VACRms| DUTycycle|DELAY|VDCRms	none
upper_limit	numeric	Depends on measurement selected	V/S/HZ/PCT
lower_limit	numeric	Depends on measurement selected	V/S/HZ/PCT

Example **Configure a limit test on frequency with acceptable results from 1 kHz to 1 MHz**

See the MEASure:LIMittest command for an example of configuring and performing a limit test

```
MEAS:COMP   FREQ,1E6,1E3
```
Configure limit test

Comments

- **Selecting Upper and Lower Limits:** Both upper and lower limits must be within the range of the measurement selected. Refer to the individual measurement commands for information on selecting available limits.
- **Starting a Limit Test:** The individual MEASure subsystem commands (not queries) are used to place the instrument in the continuous measure mode (e.g., MEAS:FREQ), then the MEASure:LIMitest command is used to start a configured limit test.
- **Related Commands:** MEASure:COMPare?, LIMittest, POSTfailure.

:COMPare?

MEASure:COMPare? ***<measurement>*** returns the current limit test configuration for the selected measurement. The data is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
measurement	discrete	RISetime|FALLtime|FREQuency| PERiod|PWIDth|NWIDth| VAMPlitude|VBASe|VTOP|VPP| VAVerage|VMAX|VMIN|VACRms| DUTycycle|DELAY|VDCRms	none

Example **Querying limit test configuration**

```
dimension  statement    String to hold data
MEAS:COMP  FREQ,1E6,1E3 Configure limit test
MEAS:COMP?  FREQ        Query instrument to return limit
                        test configuration
enter  statement        Enter data into computer
```

Comments

- **Returned Format:** The measurement configuration is returned as follows:

  <measurement>,<upper_value>,<lower_value>

  See MEASure:COMPare command for more information on returned data.

:CURSor?

MEASure:CURSor? *<type>* returns the time and voltage values of the specified marker as an ordered pair of time/voltage values. The data is sent to the output buffer. When the CURSor? query is sent, no measurement is made and the cursors are not moved.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
type	discrete	DELTa|STARt|STOP	none

Example **Query the positions of the start marker and V Marker 1**

```
dimension  statement   String to hold data
MEAS:CURS?  STAR       Query instrument to return start
                       cursor
enter  statement       Enter data into computer
```

Comments

- **Selecting Type:** The data returned *<time>,<voltage>* is dependent on the type of cursor selected as follows:

  DELTa: Returns the value of delta V and delta T.

  STARt: Returns the positions of the start time marker and start voltage marker (VMarker 1).

  STOP: Returns the positions of the stop time marker and stop voltage marker (VMarker 2).

- **Delta V/T:** The values for delta V and delta T are calculated as follows:

  delta V = Vmarker 2 – Vmarker 1.
  delta T = stop marker – start marker.

- **Moving Cursors:** See MEASure:TSTARt and:TSTOp commands for moving time start/stop time markers, and MEASure:VSTArt and:VSTOp commands for moving voltage start/stop markers.

- **Related Commands:** MEASure:TSTARt, TSTOp, VSTArt, VSTOp.

:DEFIne

MEASure:DEFine ***<measure_spec>*** **sets up the measurement standards for a USER defined measurement. Selects the definitions that DELay, PWIDth, and NWIDth measurements will use when MEASure:MODe is set to USER. Provides the option of making measurements based on signal width, or delay settings, or threshold parameters.**

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
measure_spec	discrete	DELay <polarity>,<edge>,<level>, <polarity>,<edge>,<level> *OR*	none
	discrete	PWIDth MIDDle|UPPer|LOWer *OR*	none
	discrete	NWIDth MIDDle|UPPer|LOWer	none

Example

User defined DELay measurement

This example will set the parameters for a user defined time measurement from the **first positive edge** at the **upper threshold** level to the **second negative edge** at the **middle threshold** level. If one source is specified (MEASure:SOURce), both parameters apply to that signal. If two sources are specified, the measurement is from the first positive edge on source 1 to the second negative edge on the source 2.

```
MEAS:DEF   DEL,POS,1,UPP,NEG,2,MIDD
```

Comments

- **Entering Measurement Specifications for DELay Measurements:** Delay *measure_spec* entries define the DELay measurement standards.

  DELay specifies that the following parameters are defining delay measurements.

  <polarity>,<edge>,<level> (following DELay) specifies the starting (from) slope, edge count, and transition point.

  ,<polarity>,<edge>,<level> (second group) specifies the stopping (to) slope, edge count, and transition point.

  Available entries of **<polarity>,<edge>,<level>** are as follows:

  <polarity> = POSitive or NEGative

  <edge> = 1 to 100 (excluding 0) specify an edge

  <level> = MIDDle, UPPer, or LOWer (UPPer level is set using MEASure:UPPer command, LOWer level is set using MEASure:LOWer command, MIDDle level is set to the center of the currently selected upper and lower values)

- **Entering Measurement Specifications for PWIDth Measurements:** PWIDth *measure_spec* entries define the Positive Pulse Width measurement standards.

  PWIDth specifies that the following parameters are defining positive pulse width measurements.

  MIDDle|UPPer|LOWer specifies the point on the waveform transition to measure when making the positive width of the active waveform.

  UPPer is set using MEASure:UPPer command

  LOWer is set using MEASure:LOWer command

  MIDDle is calculated at the center of the currently selected lower and upper values

- **Entering Measurement Specifications for NWIDth Measurements:** NWIDth *measure_spec* entries are identical to PWIDth measurement entries, except prefix with **NWIDth.**

- **Related Commands:** MEASure:LOWer, UPPer, UNITs.

:DEFine?

MEASure:DEFine? *<measure_spec>* returns the currently selected measurement definitions. The data is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
measure_spec	discrete	DELay|PWIDth|NWIDth	none

Example

Query the delay measurement user definitions

`dimension statement`	*String for data*
`MEAS:DEF? DEL`	*Query instrument to return delay definitions*
`enter statement`	*Enter data into computer*

Comments

- **Returned Format:** The data returned is dependent on *measure_spec* selected as follows:

  DELay: Returns the delay measurement definitions **DELay <polarity>,<edge>,<level>,<polarity>,<edge>,<level>.** See MEASure:DEFine command for more information.

  PWIDth: Returns the positive pulse width measurement definitions **PWIDth UPPer|LOWer|MIDDle.** See MEASure:DEFine command for more information.

  NWIDth: Returns the negative pulse width measurement definitions **NWIDth UPPer|LOWer|MIDDle.** See MEASure:DEFine command for more information.

:DELay

MEASure:DELay is used to place the instrument in the continuous measurement mode and start a Delay measurement.

Example

Start a Delay measurement

```
MEAS:DEL                Start a Delay measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One or two sources are specified using the MEASure:SOURce command.
- **Defining Measurements:** The measurement can be performed using standard or user defined parameters as specified by the MEASure:MODe command.

  STANdard parameters are the first rising edge at mid threshold to the second rising edge at mid threshold.

  USER defined parameters set the "from" slope, edge count, and transition points; the "to" slope, edge count, and transition points; and the upper, lower and mid threshold values (mid calculated at center of entered upper and lower threshold values).
- **Jitter Measurements:** Jitter measurements can be performed by selecting the ACQuire:TYPe to ENVelope, and specifying the two delay parameters the same. For example,

  ```
  MEAS:SOUR CHAN1,CHAN1
  MEAS:MODe USER
  MEAS:DEF DEL,POS,1,MID,POS,1,MID
  ACQ:TYP ENV
  ```
- **Executing the Measurement:** When the measurement is executed, the instrument will measure the delay from:

  the first specified edge on one source to the next specified edge on the same source (when one source is specified),

  the first specified edge on one source to the first specified edge on another source (when two sources are specified).
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:DELay command prior to the MEASure:COMPare command when configuring for a delay limit test.
- **Related Commands:** MEASure:DELay?, COMPare, LIMittest, MODe, RESults?, SOURce.

:DELay?

MEASure:DELay? turns continuous measurement mode off, performs a Delay measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform Delay measurement on channel 2 and return results

`dimension statement`	*String for data*
`MEAS:SOUR CHAN2`	*Measure channel 2*
`MEAS:DEL?`	*Perform Delay measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** In order to perform a Delay measurement, the selected edge must be present. All edges are counted from the first edge of the acquired data, not at the reference point. If the edge is not present, an error will be generated.
- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing measured delay time (in seconds).

:DESTination

MEASure:DESTination ***<location>*** specifies the destination used when a limit test violation is found. Used to save the data associated with a limit test failure.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
location	discrete	WMEMory*n* (*n*=1 to 4)| PMEMory*n* (*n*=1 to 2)| OFF	none

Example

Set destination to pixel memory 1

See the MEASure:LIMittest command for an example of configuring and performing a limit test.

`MEAS:DEST  PMEM1`	*Pixel memory 1 is the destination for limit test violations*

Comments

- **Specifying Waveform Memories:** Only one source can be stored in Waveform Memory. If Waveform Memory is specified, the source must be set up separately using the WAVeform:SOURce command. When Waveform Memory is selected, the most current data will overwrite the memory each time a violation is found. In the example below, the source is CHAN1, and the destination is WMEM2:

  ```
  WAV:SOUR CHAN1
  MEAS:DEST WMEM2
  ```

- **Specifying Pixel Memories:** If Pixel Memory is specified, an accumulated save occurs each time a violation is found. Measurements cannot be made on pixel memory.
- **Disable Destination:** OFF is specified to disable the destination function.
- **Related Commands:** MEASure:LIMittest.
- ***RST Conditions:** Defaults to OFF.

:DESTination?

MEASure:DESTination? returns the currently selected destination (WMEM1-4, PMEM1-2, or OFF) for limit test violations. The data is sent to the output buffer.

Example

Query the violation destination

`dimension  statement`	*String for data*
`MEAS:DEST  PMEM1`	*Set destination to pixel memory 1*
`MEAS:DEST?`	*Query instrument to return destination*
`enter  statement`	*Enter data into computer*

:DUTycycle

MEASure:DUTycycle is used to place the instrument in the continuous measurement mode and start a Duty Cycle measurement.

Example

Start a Duty Cycle measurement

```
MEAS:DUT                    Start a Duty Cycle measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Defining Measurements:** The measurement can be performed using standard or user defined parameters as specified by the MEASure:MODe command.

  STANdard parameters are to measure at 50% levels.

  USER defined parameters set the upper, lower and mid threshold values (mid calculated at center of entered upper and lower threshold values).
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the Duty cycle of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:DUTycycle command prior to the MEASure:COMPare command when configuring for a duty cycle limit test.
- **Related Commands:** MEASure:DUTycycle?, COMPare, LIMittest, MODe, RESults?, SOURce.

:DUTycycle?

MEASure:DUTycycle? turns continuous measurement mode off, performs a Duty Cycle measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform duty cycle measurement on channel 3 and return results

`dimension statement`	*String for data*
`MEAS:SOUR CHAN3`	*Measure channel 3*
`MEAS:DUT?`	*Perform Duty Cycle measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** In order to perform a Duty Cycle measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Measurement Method:** The method the instrument uses to determine duty cycle is to measure PWIDth and PERiod, then present duty cycle in percent as ratio of the positive pulse width to period as follows:

  duty cycle = + pulse width/period

- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing measured duty cycle. If the signal is not present, 9.99999E+37 is returned.

:ESTArt

MEASure:ESTArt ***<edge>*** used to position the start marker on the specified edge and slope of the waveform. All edges must be present and are counted from the first edge of the acquired data, not at the reference point. The start marker is positioned at the point where VMarker 1 (set using MEASure:VSTArt command) intersects the waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
edge	numeric	–32,768 to +32,767	none

Example **Set edge start marker**

This example places the start marker at the **second positive-going** intersection of the waveform and VMarker 1.

```
MEAS:ESTA  +2
```
Set start marker

Comments

- **Selecting Edge:** The desired edge number (counted from the first edge of the acquired data) is specified by sending an integer value.
  - If a positive integer is selected (+ sign, space, or blank), the instrument will place the start marker on a positive-going waveform edge.
  - If a negative integer is sent (– sign), the start marker will be placed on a negative-going waveform edge.
  - If the value is out of range, an error will be generated.
- **Related Commands:** MEASure:VSTArt.

:ESTArt?

MEASure:ESTArt? returns the edge and slope of the edge start marker. The value is sent to the output buffer. Sign indicates selected slope. +, blank, or space indicates positive going, and – indicates negative going. The number returned specifies the edge number as counted from the first edge of the acquired data.

Example **Query the edge start marker**

`MEAS:ESTA  2`	*Set start marker*
`MEAS:ESTA?`	*Query instrument to return start marker*
`enter  statement`	*Enter data into computer*

:ESTOp

MEASure:ESTOp ***<edge>*** used to position the stop marker on the specified edge and slope of the acquired waveform. All edges must be present and are counted from the first edge of the acquired data, not at the reference point. The stop marker is positioned at the point where VMarker 2 (set using MEASure:VSTOp command) intersects the waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
edge	numeric	–32768 to +32767	none

Example **Set edge stop marker**

This example places the stop marker at the **second negative-going** intersection of the waveform at VMarker2.

```
MEAS:ESTO  -2            Set stop marker
```

Comments

- **Selecting Edge:** The desired edge number (counted from the first edge of the acquired data) is specified by sending an integer value.
  - If a positive integer is selected (+ sign, space, or blank), the instrument will place the stop marker on a positive-going waveform edge.
  - If a negative integer is sent (– sign), the stop marker will be placed on a negative-going waveform edge.
  - If the value is out of range, an error will be generated.
- **Related Commands:** MEASure:VSTOp.

:ESTOp?

MEASure:ESTOp? returns the edge and slope of the stop marker. The value is sent to the output buffer. Sign indicates selected slope. "+", blank, or space indicates positive going, and "–" indicates negative going. The number returned specifies the edge number as counted from the first edge of the acquired data.

Example **Query the edge stop marker**

```
MEAS:ESTO  -2            Set stop marker
MEAS:ESTO?               Query instrument to return stop
                         marker
enter  statement         Enter data into computer
```

:FALLtime

MEASure:FALLtime is used to place the instrument in the continuous measurement mode and start a Fall Time measurement.

Example

Start a Fall Time measurement

```
MEAS:FALL              Start a Fall Time measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Defining Measurements:** The measurement can be performed using standard or user defined parameters as specified by the MEASure:MODe command.

  STANdard parameters measure at 10%/90% threshold levels.

  USER defined parameters set the upper and lower threshold values.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the fall time of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:FALLtime command prior to the MEASure:COMPare command when configuring for a fall time limit test.
- **Related Commands:** MEASure:FALLtime?, COMPare, LIMittest, MODe, RESults?, SOURce.

:FALLtime?

MEASure:FALLtime? turns continuous measurement mode off, performs a Fall Time measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform Fall Time measurement on channel 4 and return results

`dimension statement`	*String for data*
`MEAS:SOUR CHAN4`	*Measure channel 4*
`MEAS:FALL?`	*Perform Fall Time measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** In order to perform a Fall Time measurement, the falling edge of the waveform must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. For best measurement accuracy, set the sweep speed as fast as possible. If the horizontal scaling is questionable when performing a Fall Time measurement, an error will be generated.
- **Measurement Method:** The method the instrument uses to determine fall time is to measure the time at the upper threshold then the lower threshold of the first falling (negative-going) edge present. Measurement results are used to calculate fall time as follows:

  fall time = lower threshold time – upper threshold time
- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing measured fall time (in seconds).

:FREQuency

MEASure:FREQuency is used to place the instrument in the continuous measurement mode and start a Frequency measurement.

Example **Start a Frequency measurement**

```
MEAS:FREQ                   Start a Frequency measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Defining Measurements:** The measurement can be performed using standard or user defined parameters as specified by the MEASure:MODe command.

  STANdard parameters measure at 50% levels.

  USER defined parameters set the mid threshold level (center of entered upper and lower threshold values).
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the frequency of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:FREQuency command prior to the MEASure:COMPare command when configuring for a frequency limit test.
- **Related Commands:** MEASure:FREQuency?, COMPare, LIMittest, MODe, RESults?, SOURce.

:FREQuency?

MEASure:FREQuency? turns continuous measurement mode off, performs a Frequency measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform Frequency measurement on waveform memory 1 and return results

`dimension statement`	*String for data*
`MEAS:SOUR WMEM1`	*Measure WMEMory1*
`MEAS:FREQ?`	*Perform Frequency measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** In order to perform a Frequency measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Measurement Method:** The method the instrument uses to determine frequency is to measure the time of the first complete cycle, then calculate frequency as follows:

  If first edge of waveform is rising, then
  frequency = 1/(time at second rising edge – time at first rising edge)

  If first edge of waveform is falling, then:
  frequency = 1/(time at second falling edge – time at first falling edge)
- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing measured frequency (in hertz). If the signal is not present, 9.99999E+37 is returned.

:LIMittest

MEASure:LIMittest <mode> used to perform a measurement comparison or limit test on up to three measurements. The user specifies the measurements to be performed, and the acceptable range of measurement results. The user can also specify what happens if a measurement result falls outside of the acceptable range.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	MEASure|OFF	none

Example

Frequency Limit Test

This example performs a **frequency** limit test on channel 1 with violations **saved** in **waveform memory 1.** Test will **not stop** after violations.

```
MEAS:SCR                  Clear measurement queue
MEAS:SOUR  CHAN1          Measure channel 1
WAV:SOUR  CHAN1           Violation source is channel 1
MEAS:FREQ                 Start a continuous frequency
                          measurement on channel 1
MEAS:COMP  FREQ,1E6,1E3   Start a limit test for frequency
                          measurement with acceptable
                          results from 1 kHz to 1 MHz
MEAS:POST  CONT           Continue limit test after violation
MEAS:DEST  WMEM1          Waveform memory 1 is the
                          destination for limit test violations
MEAS:LIM  MEAS            Start limit test
loop  beginning           Loop beginning
LTER?                     Has limit test failed?
loop  end                 Loop end
MEAS:RES?                 Query instrument to return
                          measurement results
enter  statement          Enter measurement results into
                          computer
MEAS:LIM  OFF             Stop limit test
```

Comments

- **Limit Test Sequence:** A limit test is configured using three MEASure subsystem commands:

  COMPare - is used to set the desired measurement and acceptable limits of the test. See MEASure:COMPare command for more information.

  POSTfailure - is used to specify what will occur (continue or stop) after a violation or measurement out of range has been found. See MEASure:POSTfailure command for more information.

  DESTination - is used to specify the destination where data associated with a limit test failure is stored. See MEASure:DESTination command for more information.

- **Limit Test Status:** Failures can be determined by one of two commands:

  LTER? Used to return if the limit test has failed. See LTER? query for more information.

  MEASure:RESults? Used to return the current, minimum, maximum, and pass ratio values for the limit test. See MEASure:RESults? query for more information.

- **Related Commands:** LTER?, MEASure:COMPare, DESTination, POSTfailure, RESults?.
- ***RST Conditions:** Defaults to OFF.

:LOWer

MEASure:LOWer *<value>* is used to set the lower measurement threshold level. The *value* that is sent will be in the units currently selected with the MEASure:UNITs command.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
value	numeric	–250,000 to +250,000	V
	numeric	–25.00 to +125.0	PCT

Example

Set lower threshold to 25 V

```
MEAS:UNIT  VOLT          Units to volts
MEAS:LOW  25             Lower threshold is 25V
```

Comments

- **Entering Value:** The MEASure:UNITs command should precede the MEASure:LOWer command to set units to desired value. When VOLTs are selected as the unit, all entries will be rounded to the nearest millivolt. When PERCent is selected as the unit, all entries will be rounded to the nearest tenth percent.
- **MIDDle Threshold:** When MEASure:MODe USER is selected, all "MID" threshold levels used for measurements are calculated to the center of the present UPPer and LOWer threshold levels. Changing LOWer threshold level will effect the MIDDle threshold level.
- **Related Commands:** MEASure:UNITs, MODe, UPPer.
- ***RST Condition:** Defaults to 10 (%).

:LOWer?

MEASure:LOWer? returns the currently selected lower measurement threshold level. The value is sent to the output buffer. The value that is returned will be in the units currently selected with the MEASure:UNITs command. See MEASure:LOWer command for more information.

Example

Query the lower threshold level

```
MEAS:UNIT  VOLT          Units to volts
MEAS:LOW  25             Lower threshold is 25V
MEAS:UNIT?               Query instrument to return units
enter  statement         Enter data into computer
MEAS:LOW?                Query instrument to return lower
                         threshold
enter  statement         Enter data into computer
```

Comments

- **Query Value:** The MEASure:UNITs? query should precede the MEASure:LOWer? query to determine the current threshold units.
- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing lower threshold (in volts or percent).
- **Determining MIDDle Threshold Level:** MIDDle threshold can be determined by querying both the LOWer and UPPer threshold values, and calculating the center point. For example, if LOWer = 20% and UPPer = 30% then MIDDle is 25%.
- **Related Commands:** MEASure:UNITs, MODe, UPPer.

:MODe

MEASure:MODe *<mode>* is used to set the standards (definitions and thresholds) under which the measurement will be performed. Allows the user to perform the measurement using "standard" parameters, or with "user defined" parameters.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	STANdard|USER	none

Example

Set mode so user can define measurement standards

```
MEAS:MOD  USER          Set mode to USER
```

Comments

- **Selecting Mode:** Measurement mode is selected as follows:

  STANdard - Measurements are performed using default parameters that follow "IEEE" measurement techniques. When measurements are requested, the instrument first determines the top (100%) and base (0%) voltages of the waveform. From this information, thresholds of 90% (upper), 50% (middle), and 10% (lower) are determined. Rise time and Fall time measurements are made at the 90% (upper) and 10% (lower) levels. All other measurements are made using the 50% (middle) level. Delay measurements are made from the the first positive edge to the second positive edge.

  USER - Measurements are performed to user specified parameters, allowing measurements to be based on selectable signal width, delay settings, and/or threshold levels.

  Use the MEASure:LOWer, and UPPer commands to set desired threshold levels. MIDDle is set to the center of the current upper and lower threshold levels.

  Use the MEASure:DEFine command to enter delay setting and signal width parameters.

- **Related Commands:** MEASure:DEFine, LOWer, UPPer.
- ***RST Condition:** Defaults to STANdard.

:MODe?

MEASure:MODe? returns the currently selected mode under which the measurements will be performed. The data is sent to the output buffer. Returns STANdard when "IEEE" parameters are used. Returns USER when "user defined" parameters are used. See MEASure:MODe command for more information.

Example

Query current measurement mode

```
dimension  statement    String for data
MEAS:MOD  USER          Set mode to USER
MEAS:MOD?               Query instrument to return mode
enter  statement        Enter data into computer
```

:NWIDth MEASure:NWIDth is used to place the instrument in the continuous measurement mode and start a Negative Pulse Width measurement.

Example **Start a Negative Pulse Width measurement**

```
MEAS:NWID            Start a Negative Pulse Width
                     measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Defining Measurements:** The measurement can be performed using standard or user defined parameters as specified by the MEASure:MODe command.

  STANdard parameters measure at 50% levels.

  USER defined parameters select the transition (upper, middle, lower) to measure when making the measurement. Determined by the current upper, lower and mid threshold values (mid calculated at center of entered upper and lower threshold values).
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the negative pulse width of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:NWIDth command prior to the MEASure:COMPare command when configuring for a negative pulse width limit test.
- **Related Commands:** MEASure:NWIDth?, COMPare, LIMittest, MODe, RESults?, SOURce.

:NWIDth?

MEASure:NWIDth? turns continuous measurement mode off, performs a Negative Pulse Width measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform Negative Pulse Width measurement on channel 1 and return results

`dimension statement`	*String for data*
`MEAS:SOUR CHAN1`	*Measure channel 1*
`MEAS:NWID?`	*Perform Negative Pulse Width measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** In order to perform a Negative Pulse Width measurement, a minimum of one negative pulse must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Measurement Method:** The method the instrument uses to determine pulse width is to measure the time of the first complete negative pulse, then calculate pulse width as follows:

  If first edge of waveform is rising, then
  pulse width = time at second rising edge – time at first falling edge

  If first edge of waveform is falling, then:
  pulse width = time at first rising edge – time at first falling edge

- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing measured negative pulse width (in seconds). If the signal is not present, 9.99999E+37 is returned.

:OVERshoot

MEASure:OVERshoot is used to place the instrument in the continuous measurement mode and start an Overshoot measurement.

Example

Start an Overshoot measurement

```
MEAS:OVER                  Start an  Overshoot measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the Overshoot of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:OVERshoot command prior to the MEASure:COMPare command when configuring for an overshoot limit test.
- **Related Commands:** MEASure:OVERshoot?, COMPare, LIMittest, RESults?, SOURce.

:OVERshoot?

MEASure:OVERshoot? turns continuous measurement mode off, performs an Overshoot measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform Overshoot measurement on channel 2 and return results

```
dimension  statement   String for data
MEAS:SOUR  CHAN2       Measure channel 2
MEAS:OVER?             Perform Overshoot measurement,
                       query instrument to return
                       measurement results
enter  statement       Enter measurement results into
                       computer
```

Comments

- **Oscilloscope Setup:** In order to perform an Overshoot measurement, a minimum of one edge must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Measurement Method:** The method the instrument uses to determine overshoot is to make three different voltage measurements, then calculate overshoot as follows:

  If first edge of waveform is rising, then:
  overshoot = (VMAX - VTOP)/VAMPlitude

  If first edge of waveform is falling, then:
  overshoot = (VBASe - VMIN)/VAMPlitude

- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing measured overshoot. If the signal is not present, 9.99999E+37 is returned.
- **Related Commands:** MEASure:VBASe, VMIN, VMAX, VTOP, VAMPlitude.

:PERiod

MEASure:PERiod is used to place the instrument in the continuous measurement mode and start a Period measurement.

Example

Start a Period measurement

```
MEAS:PER                    Start a Period measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Defining Measurements:** The measurement can be performed using standard or user defined parameters as specified by the MEASure:MODe command.

  STANdard parameters measure at 50% levels.

  USER defined parameters set the mid threshold level (center of entered upper and lower threshold values).
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the period of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:PERiod command prior to the MEASure:COMPare command when configuring for a period limit test.
- **Related Commands:** MEASure:PERiod?, COMPare, LIMittest, MODe, RESults?, SOURce.

:PERiod?

MEASure:PERiod? turns continuous measurement mode off, performs a Period measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example Perform Period measurement on function 1 and return results

`dimension`	`statement`	*String for data*
`MEAS:SOUR`	`FUNC1`	*Measure function 1*
`MEAS:PER?`		*Perform Period measurement, query instrument to return measurement results*
`enter`	`statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** In order to perform a Period measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Measurement Method:** The method the instrument uses to determine period is to measure the time of the first complete cycle, then calculate period as follows:

  If first edge of waveform is rising, then
  period = time at second rising edge – time at first rising edge

  If first edge of waveform is falling, then:
  period = time at second falling edge – time at first falling edge
- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing measured period (in seconds). If the signal is not present, 9.99999E+37 is returned.

:POSTfailure

MEASure:POSTfailure ***<mode>*** **is used to** specify what will occur (limit test continue or stop) after a violation has been found during a limit test.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	CONTinue|STOP	none

Example

Set to continue limit test after violation

See the MEASure:LIMittest command for an example of configuring and performing a limit test.

```
MEAS:POST  CONT         Continue limit test after violation
```

Comments

- **Selecting Mode:** After a violation (entered limits exceeded) the limit test will:

  STOP the limit test.

  CONTinue to look for another violation. If MEASure:DESTination is not OFF, the violation will be written to the selected memory location.

- **Related Commands:** MEASure:LIMittest, DESTination.
- ***RST Conditions:** Defaults to STOP.

:POSTfailure?

MEASure:POSTfailure? returns the currently selected failure instructions for limit test violations. The data is sent to the output buffer. If CONTinue is returned, the limit test will continue. If STOP is returned, the limit test will stop.

Example

Query the limit test failure instruction

```
dimension  statement    String for data
MEAS:POST  CONT         Continue limit test after violation
MEAS:POST?              Query instrument to return post
                        failure instruction
enter  statement        Enter data into computer
```

:PRECision

MEASure:PRECision has no effect on instrument operations, and is only included for compatibility with other instruments.

:PRECision?

MEASure:PRECision? always returns COARse. This query has no effect on instrument operations, and is only included for compatibility with other instruments.

:PREShoot

MEASure:PREShoot is used to place the instrument in the continuous measurement mode and start a Preshoot measurement.

Example **Start a Preshoot measurement**

```
MEAS:PRES                    Start a Preshoot measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the preshoot of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:PREShoot command prior to the MEASure:COMPare command when configuring for a preshoot limit test.
- **Related Commands:** MEASure:PREShoot?, COMPare, LIMittest, RESults?, SOURce.

:PREShoot?

MEASure:PREShoot? turns continuous measurement mode off, performs a Preshoot measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example **Perform Preshoot measurement on channel 4 and return results**

```
dimension  statement  String for data
MEAS:SOUR  CHAN4      Measure channel 4
MEAS:PRES?            Perform Preshoot measurement,
                      query instrument to return
                      measurement results
enter  statement      Enter measurement results into
                      computer
```

Comments

- **Oscilloscope Setup:** In order to perform a Preshoot measurement, a minimum of one edge must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Measurement Method:** The method the instrument uses to determine preshoot is to make three different voltage measurements, then calculate preshoot as follows:

  If first edge of waveform is rising, then
  preshoot = (VBASe - VMIN)/VAMPlitude

  If first edge of waveform is falling, then:
  preshoot = (VMAX - VTOP)/VAMPlitude

- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing measured preshoot.
- **Related Commands:** MEASure:VBASe, VMIN, VMAX, VTOP, VAMPlitude.

:PWIDth

MEASure:PWIDth is used to place the instrument in the continuous measurement mode and start a Positive Pulse Width measurement.

Example

Start a Positive Pulse Width measurement

```
MEAS:PWID
```

Start a Positive Pulse Width measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Defining Measurements:** The measurement can be performed using standard or user defined parameters as specified by the MEASure:MODe command.

  STANdard parameters measure at 50% levels.

  USER defined parameters select the transition (upper, middle, lower) to measure when making the measurement. Determined by the current upper, lower and mid threshold values (mid calculated at center of entered upper and lower threshold values).
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the positive pulse width of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:PWIDth command prior to the MEASure:COMPare command when configuring for a positive pulse width limit test.
- **Related Commands:** MEASure:PWIDth?, COMPare, LIMittest, MODe, RESults?, SOURce.

:PWIDth?

MEASure:PWIDth? turns continuous measurement mode off, performs a Positive Pulse Width measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform Positive Pulse Width measurement on channel 1 and return results

`MEAS:SOUR CHAN1`	*Measure channel 1*
`MEAS:PWID?`	*Perform Positive Pulse Width measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** In order to perform a Positive Pulse Width measurement, a minimum of one positive pulse must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Measurement Method:** The method the instrument uses to determine pulse width is to measure the time of the first complete positive pulse, then calculate pulse width as follows:

  If first edge of waveform is rising, then
  pulse width = time at second falling edge – time at first rising edge

  If first edge of waveform is falling, then:
  pulse width = time at first falling edge – time at first rising edge

- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing measured positive pulse width (in seconds). If the signal is not present, 9.99999E+37 is returned.

:RESults?

MEASure:RESults? returns the currently active measurement results.

Example **Return active measurement results**

`dimension statement`	*String to hold data*
`MEAS:RES?`	*Query instrument to return results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Returned Format:** The measurement results are returned as follows:

  <No. of Meas> [;*<measurement_name measurement_result>*]

  No. of Meas is a numeric value representing the number of measurements present in the measurement queue. Up to eight measurements are saved.

  measurement_name indicates the type of measurement.

  measurement_result is the measured value(s).

  If the measurement queue is empty, "0" is returned.

- **RESults? versus STATistics:** When MEASure:STATistics is selected, executing the RESult? query returns one of the following results depending on the MEASure:LIMittest selection:

  LIMittest OFF: The current, minimum, maximum, and average values for each measurement are returned.

  LIMittest MEASure: The current, minimum, maximum, and pass ratio values for the limit test are returned.

- **System Headers:** The measurement result is returned as a numeric value when SYSTem:HEADer is set to OFF.

- **Related Commands:** MEASure:STATistics, LIMittest, POSTfailure.

:RISetime

MEASure:RISetime is used to place the instrument in the continuous measurement mode and start a Rise Time measurement.

Example

Start a Rise Time measurement

```
MEAS:RIS                Start a Rise Time measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Defining Measurements:** The measurement can be performed using standard or user defined parameters as specified by the MEASure:MODe command.

  STANdard parameters measure at 10%/90% threshold levels.

  USER defined parameters set the upper and lower threshold values.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the rise time of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:RISetime command prior to the MEASure:COMPare command when configuring for a rise time limit test.
- **Related Commands:** MEASure:RISetime?, COMPare, LIMittest, RESults?, SOURce.

:RISetime?

MEASure:RISetime? turns continuous measurement mode off, performs a Rise Time measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example **Perform Rise Time measurement on channel 3 and return results**

`dimension statement`	*String for data*
`MEAS:SOUR CHAN3`	*Measure channel 3*
`MEAS:RIS?`	*Perform Rise Time measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** In order to perform a Rise Time measurement, the rising edge of the waveform must be present. For best measurement accuracy, set the sweep speed as fast as possible. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. If the horizontal scaling is questionable when performing a Rise Time measurement, an error will be generated.
- **Measurement Method:** The method the instrument uses to determine rise time is to measure the time at the lower threshold then the upper threshold of the first rising (positive-going) edge present. Measurement results are used to calculate rise time as follows:

  rise time = upper threshold time – lower threshold time
- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing measured rise time (in seconds). If the signal is not present, 9.99999E+37 is returned.

:SCRatch

MEASure:SCRatch is used to clear the measurement results from the measurement queue.

Example **Clear measurement results from the measurement queue**

`MEAS:SCR`	*Clear measurement queue*

Comments

- **Related Commands:** MEASure:RESults?.

:SOURce

MEASure:SOURce ***<source>*[,*<source>*]** is used to select the source(s) for measurement. The source(s) specified become the source(s) for all the MEASure subsystem commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
source	discrete	CHANnel*n* (*n*=1 to 4)| FUNCtion*n* (*n*=1 or 2) WMEMory*n* (*n*=1 to 4)	none

Example

Set source to waveform memory 3

```
MEAS:SOUR  WMEM3
```
WMEMory 3 specified as source for all MEASure commands

Comments

- **Specifying Source:** Two different sources can be specified with this command, however, all measurements except DELay are made on the first specified source only.

  SOURce and DELay: The DELay measurement will use two sources if two have been specified. If only one source is specified, the DELay measurement will use that source for both parameters.

- ***RST Condition:** Defaults to CHANnel1,CHANnel1.

:SOURce?

MEASure:SOURce? returns the currently selected source (CHAN1-4, FUNC1-2, WMEM1-4) for measurement.

Example

Query selected source(s) for MEASure subsystem

`dimension statement`	*String for data*
`MEAS:SOUR?`	*Query instrument to return selected measurement source(s)*
`enter statement`	*Enter measurement results into computer*

Comments

- **Returned Format:** The measurement source(s) returned are dependent on how many different sources are selected.

  If the specified source(s) are different, both will be returned.

  If the specified source(s) are the same, only one will be returned.

  See MEASure:SOURce command for additional information on available source(s).

:STATistics

MEASure:STATistics ***<mode>*** select the statistics mode. When ON the minimum, maximum, average (or pass ratio), and current measurement results are placed in the measurement queue. When OFF, only the current measurement results are placed in the measurement queue. Measurements must be in the continuous mode.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
mode	boolean	OFF|0|ON|1	none

Example

Enable statistics mode

```
MEAS:STAT 1                  Statistics mode enabled
```

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Measurement Queue:** The measurement queue holds up to eight measurements, however executing the RESults? query will only return the last three measurements.
- **Read Statistics:** Use the MEASure:RESult? query to read measurement results. MEASure:STATistics? query only reads state (ON/OFF).
- **Average or Pass Ratio:** Average is replaced by pass ratio when MEASure:LIMittest is selected and MEASure:POSTfailure is CONTinue. Pass ratio lists the percentage of times a certain test has passed.
- **Related Commands:** MEASure:RESults?.
- ***RST Conditions:** Defaults to OFF.

:STATistics?

MEASure:STATistics? returns a number to show whether mode is enabled or disabled. "1" = ON, "0" = OFF. The value is sent to the output buffer.

Example

Querying statistics state

```
MEAS:STAT 1                  Statistics mode enabled
MEAS:STAT?                   Query instrument to return statistics
                             mode state
enter statement              Enter value into computer
```

:TDELta?

MEASure:TDELta? returns the time difference between the start and stop time markers. The value is sent to the output buffer.

Example **Query the time difference between start and stop time markers**

`dimension statement`	*String for data*
`MEAS:TDEL?`	*Query instrument to return time difference*
`enter statement`	*Enter data into computer*

Comments

- **Measurement Method:** Time difference is calculated as follows:

  TDELta = TSTOp – TSTArt

  TSTOp is the current time stop marker position as set by the MEASure:TSTOp command. TSTArt is the current time start marker position as set by the MEASure:TSTArt command.

- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing time difference (in seconds). A (–) negative number indicates the stop marker is preceding the start marker.

- **Related Commands:** MEASure:TSTArt, TSTOp.

:TMAX?

MEASure:TMAX? returns the time at which the first maximum voltage occurred on the acquired waveform. The value is sent to the output buffer.

Example **Query the first maximum voltage time**

`dimension statement`	*String for data*
`MEAS:TMAX?`	*Query instrument to return maximum voltage time*
`enter statement`	*Enter data into computer*

Comments

- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing time (in seconds) that the maximum voltage occurred. The trigger point is used as the reference (time 0).

:TMIN?

MEASure:TMIN? returns the time at which the first minimum voltage occurred on the acquired waveform. The value is sent to the output buffer.

Example **Query the first minimum voltage time**

`dimension statement`	*String for data*
`MEAS:TMIN?`	*Query instrument to return minimum voltage time*
`enter statement`	*Enter data into computer*

Comments

- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Returned Format:** The measurement result is returned as a numeric value (SYSTem:HEADer to OFF) representing time (in seconds) that the minimum voltage occurred. The trigger point is used as the reference (time 0).

:TSTArt

MEASure:TSTArt ***<time>*** is used to position the time start marker at a specified time with respect to trigger time.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
time	numeric	−200 to + 200	S

Example

Set time start marker to 25 nsec

```
MEAS:TSTA  25E-9        Set time start marker to 25 nsec
```

Comments

- **Selecting Time:** The desired time is specified using trigger time as the reference (time 0). Entry can be a positive (post-trigger) or negative (pre-trigger) number.
- **Related Commands:** MEASure:TSTOp, TDELta?.

:TSTArt?

MEASure:TSTArt? returns the current position of the time start marker. The value is sent to the output buffer. The number returned specifies the position of the time start marker in seconds from trigger time.

Example

Query the time start marker

```
dimension  statement   String for data
MEAS:TSTA  25E-9       Set time start marker to 25 nsec
MEAS:TSTA?             Query instrument to return time start
                       marker
enter  statement       Enter data into computer
```

:TSTOp

MEASure:TSTOp ***<time>*** is used to position the time stop marker at a specified time with respect to trigger time.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
time	numeric	−200 to + 200	S

Example

Set time stop marker to 50 nsec

```
MEAS:TSTO  50E-9        Set time stop marker to 50 nsec
```

Comments

- **Selecting Time:** The desired time is specified using trigger time as the reference (time 0). Entry can be a positive (post-trigger) or negative (pre-trigger) number.
- **Related Commands:** MEASure:TSTArt, TDELta?.

:TSTOp?

MEASure:TSTOp? returns the current position of the time stop marker. The value is sent to the output buffer. The number returned specifies the position of the time stop marker in seconds from trigger time.

Example

Query the time stop marker

```
dimension  statement    String for data
MEAS:TSTO  50E-9        Set time stop marker to 50 nsec
MEAS:TSTO?              Query instrument to return time stop
                        marker
enter  statement        Enter data into computer
```

:TVOLt?

MEASure:TVOLt? ***<voltage>,<slope><occurrence>*** is used to search the selected source for a defined voltage level and transition. The time interval between the trigger event and this defined occurrence is returned as the response to this query.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
voltage	numeric	–9.99999E+37 to + 9.99999E+37	V
slope	numeric	+ or space|–	none
occurrence	numeric	±9.99999E+37	none

Example

Perform TVOLt function

For this example, return the time between the trigger event and the **third** time the waveform crosses **–25V** in the **positive** direction.

```
MEAS:TVOL?  -25,+3
```

Comments

- **Specifying the Parameters:** Parameters are entered as follows:

  ***<voltage>*:** The desired level to be reported is entered. Use a minus (–) sign or positive (use a space or +) as required.

  <slope>: Specify (+) or a space to select the rising edge, and (–) to select the falling edge.

  <occurrence>: Specify the occurrence to be reported.

- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing the time in seconds with the trigger point as the reference when all the specified parameters were true.

  True would be defined as the waveform crossing the specified voltage for the specified number of times in the specified direction.

  If all the specified parameters were never true, +9.99999E+37 is returned.

:UNITs

MEASure:UNITs ***<unit>*** sets the measurement threshold units when the user defined measurement mode is selected.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
unit	discrete	PERCent|VOLTs	none

Example

Set threshold units to volts

```
MEAS:UNIT  VOLT          Threshold units are in volts
```

Comments

- **Changing Units:** When units are changed from the existing setting, the stored upper, lower, and middle threshold values will automatically be changed to the same value for the new units.
- **Related Commands:** MEASure:LOWer, UPPer, MODe.
- ***RST Condition:** Defaults to PERCent.

:UNITs?

MEASure:UNITs? returns the currently selected measurement threshold units (PERCent or VOLTs). The data is sent to the output buffer.

Example

Query the current units

```
dimension  statement     String for data
MEAS:UNIT  VOLT          Units to volts
MEAS:UNIT?               Query instrument to return units
enter  statement         Enter data into computer
```

:TVOLt?

MEASure:TVOLt? ***<voltage>,<slope><occurrence>*** is used to search the selected source for a defined voltage level and transition. The time interval between the trigger event and this defined occurrence is returned as the response to this query.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
voltage	numeric	−9.99999E+37 to + 9.99999E+37	V
slope	numeric	+ or space⊢	none
occurrence	numeric	±9.99999E+37	none

Example

Perform TVOLt function

For this example, return the time between the trigger event and the **third** time the waveform crosses **−25V** in the **positive** direction.

```
MEAS:TVOL?  -25,+3
```

Comments

- **Specifying the Parameters:** Parameters are entered as follows:

  <voltage>: The desired level to be reported is entered. Use a minus (–) sign or positive (use a space or +) as required.

  <slope>: Specify (+) or a space to select the rising edge, and (–) to select the falling edge.

  <occurrence>: Specify the occurrence to be reported.

- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing the time in seconds with the trigger point as the reference when all the specified parameters were true.

  True would be defined as the waveform crossing the specified voltage for the specified number of times in the specified direction.

  If all the specified parameters were never true, +9.99999E+37 is returned.

:UNITs

MEASure:UNITs ***<unit>*** sets the measurement threshold units when the user defined measurement mode is selected.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
unit	discrete	PERCent|VOLTs	none

Example

Set threshold units to volts

```
MEAS:UNIT  VOLT          Threshold units are in volts
```

Comments

- **Changing Units:** When units are changed from the existing setting, the stored upper, lower, and middle threshold values will automatically be changed to the same value for the new units.
- **Related Commands:** MEASure:LOWer, UPPer, MODe.
- ***RST Condition:** Defaults to PERCent.

:UNITs?

MEASure:UNITs? returns the currently selected measurement threshold units (PERCent or VOLTs). The data is sent to the output buffer.

Example

Query the current units

```
dimension  statement   String for data
MEAS:UNIT  VOLT        Units to volts
MEAS:UNIT?             Query instrument to return units
enter  statement       Enter data into computer
```

:UPPer

MEASure:UPPer ***<value>*** **is used to set the upper measurement threshold level. The** ***value*** **that is sent will be in the units currently selected with the MEASure:UNITs command.**

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
value	numeric numeric	–250,000 to +250,000 –25.00 to +125.0	V PCT

Example **Set upper threshold to 25 %**

```
MEAS:UNIT  PERC        Units to percent
MEAS:UPP  25           Upper threshold is 25%
```

Comments

- **Entering Value:** The MEASure:UNITs command should precede the MEASure:UPPer command to set units to a desired value. When VOLTs are selected as the unit, all entries will be rounded to the nearest millivolt. When PERCent is selected as the unit, all entries will be rounded to the nearest tenth percent.
- **MIDDle Threshold:** When MEASure:MODe USER is selected, all "MID" threshold levels used during measurements are calculated to the center of the present UPPer and LOWer threshold levels. Changing UPPer threshold level will affect the MIDDle threshold level.
- **Related Commands:** MEASure:UNITs, MODe, LOWer.
- ***RST Condition:** Defaults to 90 (%).

:UPPer?

MEASure:UPPer? returns the currently selected upper measurement threshold level. The value is sent to the output buffer. The value that is returned will be in the units currently selected with the MEASure:UNITs command. See MEASure:UPPer command for more information.

Example **Query the upper threshold level**

```
MEAS:UNIT  PERC        Units to percent
MEAS:UPP  25           Upper threshold is 25%
MEAS:UNIT?             Query instrument to return units
enter  statement       Enter data into computer
MEAS:UPP?              Query instrument to return lower
                       threshold
enter  statement       Enter data into computer
```

Comments

- **Query Value:** The MEASure:UNITs? query should precede the MEASure:UPPer? query to determine the current threshold units.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing upper threshold (in volts or percent).
- **Determining MIDDle Threshold Level:** MIDDle threshold can be determined by querying both the LOWer and UPPer threshold values, and calculating the center point. For example, if LOWer = 20% and UPPer = 30% then MIDDle is 25%.
- **Related Commands:** MEASure:UNITs, MODe, LOWer.

:VACRms

MEASure:VACRms is used to place the instrument in the continuous measurement mode and start an AC RMS Voltage measurement.

Example **Start an AC RMS Voltage measurement**

```
MEAS:VACRms                Start an AC RMS Voltage
                           measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the AC RMS voltage of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:VACRms command prior to the MEASure:COMPare command when configuring for an AC RMS voltage limit test.
- **VACRms versus VRMS:** The MEASure:VACRms command is identical to the MEASure:VRMS command.
- **Related Commands:** MEASure:VACRms?, COMPare, LIMittest, RESults?, SOURce, VRMS.

:VACRms?

MEASure:VACRms? turns continuous measurement mode off, performs an AC RMS Voltage measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example **Perform an ACRMS amplitude measurement on channel 3 and return results**

```
dimension statement        String for data
MEAS:SOUR CHAN3            Measure channel 3
MEAS:VACR?                 Perform an AC RMS Voltage
                           measurement, query instrument to
                           return measurement results
enter statement            Enter measurement results into
                           computer
```

Comments

- **Oscilloscope Setup:** The AC RMS Voltage measurement is made using the the first cycle present. If a complete cycle is not present, the AC RMS value of all data points is calculated.
- **Measurement Method:** The method the instrument uses to determine AC RMS voltage is to measure VAVerage, subtract it from each data point, then calculate AC RMS voltage.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing measured AC RMS voltage (in volts), with 0 volts as the reference.
- **VACRms? versus VRMS?:** The MEASure:VACRms? query is identical to the MEASure:VRMS? query .
- **Related Commands:** MEASure:VRMS?.

:VAMPlitude

MEASure:VAMPlitude is used to place the instrument in the continuous measurement mode and start an Amplitude Voltage measurement.

Example

Start an Amplitude Voltage measurement

```
MEAS:VAMP                  Start an Amplitude Voltage
                           measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the amplitude voltage of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:VAMPlitude command prior to the MEASure:COMPare command when configuring for an amplitude voltage limit test.
- **Related Commands:** MEASure:VAMPlitude?,VTOP, VBASe, COMPare, LIMittest, RESults?, SOURce.

:VAMPlitude?

MEASure:VAMPlitude? turns continuous measurement mode off, performs an Amplitude Voltage measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform amplitude voltage measurement on function 2 and return results

```
dimension  statement  String for data
MEAS:SOUR  FUNC2      Measure function 2
MEAS:VAMP?            Perform Amplitude Voltage
                      measurement, query instrument to
                      return measurement results
enter  statement      Enter measurement results into
                      computer
```

Comments

- **Oscilloscope Setup:** Amplitude Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Measurement Method:** The method the instrument uses to determine voltage amplitude is to measure VTOP and VBASe, then calculate voltage amplitude as follows:

  voltage amplitude = VTOP − VBASe
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing measured voltage amplitude (in volts) with 0 volts as the reference.
- **Measuring Pulse Signals:** The measured VAMPlitude value will not normally be the same as the "peak-to-peak value" if the input signal is a pulse.

:VAVerage

MEASure:VAVerage is used to place the instrument in the continuous measurement mode and start an Average Voltage measurement.

Example

Start an Average Voltage measurement

```
MEAS:VAV                 Start an Average Voltage
                         measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the average voltage of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:VAVerage command prior to the MEASure:COMPare command when configuring for an average voltage limit test.
- **Related Commands:** MEASure:VAVerage?, COMPare, LIMittest, RESults?, SOURce.

:VAVerage?

MEASure:VAVerage? turns continuous measurement mode off, performs an Average Voltage measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform voltage amplitude measurement on waveform memory 3 and return results

```
dimension  statement     String for data
MEAS:SOUR  WMEM3         Measure waveform memory 3
MEAS:VAV?                Perform Average Voltage
                         measurement, query instrument to
                         return measurement results
enter  statement         Enter measurement results into
                         computer
```

Comments

- **Oscilloscope Setup:** Average Voltage measurement is made using the the first cycle present. If a complete cycle is not present, all currently acquired data points are averaged.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing measured average voltage (in volts) with 0 volts as the reference.

:VBASe

MEASure:VBASe is used to place the instrument in the continuous measurement mode and start a Base Voltage measurement.

Example **Start a Base Voltage measurement**

```
MEAS:VBAS              Start a Base Voltage measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the voltage value at the base of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:VBASe command prior to the MEASure:COMPare command when configuring for a base voltage limit test.
- **Related Commands:** MEASure:VBASe?, COMPare, LIMittest, RESults?, SOURce.

:VBASe?

MEASure:VBASe? turns continuous measurement mode off, performs a Base Voltage measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example **Perform base voltage measurement on channel 4 and return results**

```
dimension  statement   String for data
MEAS:SOUR  CHAN4       Measure channel 4
MEAS:VBAS?             Perform Base Voltage measurement,
                       query instrument to return
                       measurement results
enter  statement       Enter measurement results into
                       computer
```

Comments

- **Oscilloscope Setup:** Base Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing measured base voltage (in volts), with 0 volts as the reference.
- **Measuring Pulse Signals:** The measured VBASe value will not normally be the "minimum value" if the input signal is a pulse.

:VDCRms

MEASure:VDCRms is used to place the instrument in the continuous measurement mode and start a DC RMS Voltage measurement.

Example

Start a DC RMS Voltage measurement

`MEAS:VDCRms`	*Start a DC RMS Voltage measurement*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the DC RMS voltage of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:VDCRms command prior to the MEASure:COMPare command when configuring for a DC RMS voltage limit test.
- **Related Commands:** MEASure:VDCRms?, COMPare, LIMittest, RESults?, SOURce.

:VDCRms?

MEASure:VDCRms? turns continuous measurement mode off, performs a DC RMS Voltage measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform a DC RMS measurement on channel 1 and return results

`dimension statement`	*String for data*
`MEAS:SOUR CHAN1`	*Measure channel 1*
`MEAS:VDCR?`	*Perform a DC RMS Voltage measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** The DC RMS Voltage measurement is made using the the first cycle present. If a complete cycle is not present, the instrument calculates the DC RMS value of all data points.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing measured DC RMS voltage (in volts), with 0 volts as the reference.

:VDELta?

MEASure:VDELta? returns the time difference between the start (VMarker1) and stop (VMarker2) voltage markers. The value is sent to the output buffer.

Example **Query the voltage difference between start and stop voltage markers**

`dimension statement`	*String for data*
`MEAS:VDEL?`	*Query instrument to return voltage difference*
`enter statement`	*Enter data into computer*

Comments

- **Measurement Method:** Voltage difference is calculated as follows:

  VDELta = VSTOp – VSTArt

  VSTOp is the current stop marker position, and VSTArt is the current start marker position. Markers are automatically set during a measurement, or by the MEASure:VSTOp and MEASure:VSTArt commands.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing voltage difference (in volts). A (–) negative number indicates the start marker is higher than the stop marker.
- **Related Commands:** MEASure:VSTArt, VSTOp.

:VFIFty

MEASure:VFIFty is used to find the top and base values of the specified waveform(s), then places the voltage markers at the 50% voltage point on the specified source(s).

Example **Set voltage markers (Vmarker1 and 2) at 50% levels**

`MEAS:VFIF`	*Voltage markers set at 50% levels*

Comments

- **Selecting Source:** Up to two source(s) are specified using the MEASure:SOURce command.

  When one source is specified, both voltage markers (VMarker1 and VMarker2) are set to the 50% voltage level on that source.

  When two sources are specified, VMarker1 is set to the 50% level of the first source and VMarker2 is set to the 50% level of the second source.
- **Voltage Marker Query:** There is not a query for the VFIFty command. Marker values can be returned using the VSTArt? and VSTOp? commands.
- **Related Commands:** MEASure:SOURce, VSTArt?, VSTOp?.

:VMAX

MEASure:VMAX is used to place the instrument in the continuous measurement mode and start a Maximum Voltage measurement.

Example **Start a Maximum Voltage measurement**

`MEAS:VMAX`	*Start a Maximum Voltage measurement*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the absolute maximum voltage of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:VMAX command prior to the MEASure:COMPare command when configuring for a maximum voltage limit test.
- **Related Commands:** MEASure:VMAX?, COMPare, LIMittest, RESults?, SOURce.

:VMAX?

MEASure:VMAX? turns continuous measurement mode off, performs a Maximum Voltage measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example **Perform maximum voltage measurement on channel 2 and return results**

`dimension statement`	*String for data*
`MEAS:SOUR CHAN2`	*Measure channel 2*
`MEAS:VMAX?`	*Perform Maximum Voltage measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** Maximum Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing measured maximum voltage (in volts), with 0 volts as the reference.

:VMIN

MEASure:VMIN is used to place the instrument in the continuous measurement mode and start a Minimum Voltage measurement.

Example **Start a Minimum Voltage measurement**

```
MEAS:VMIN                   Start a Min Voltage measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the absolute minimum voltage of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:VMIN command prior to the MEASure:COMPare command when configuring for a minimum voltage limit test.
- **Related Commands:** MEASure:VMIN?, COMPare, LIMittest, RESults?, SOURce.

:VMIN?

MEASure:VMIN? turns continuous measurement mode off, performs a Minimum Voltage measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example **Perform minimum voltage measurement on function 1 and return results**

```
dimension  statement   String for data
MEAS:SOUR  FUNC1       Measure function 1
MEAS:VMIN?             Perform Minimum Voltage
                       measurement, query instrument to
                       return measurement results
enter  statement       Enter measurement results into
                       computer
```

Comments

- **Oscilloscope Setup:** Minimum Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing measured minimum voltage (in volts), with 0 volts as the reference.

:VPP

MEASure:VPP is used to place the instrument in the continuous measurement mode and start a Peak-to-Peak Voltage measurement.

Example **Start a Peak-to-Peak Voltage measurement**

```
MEAS:VPP                    Start a Vp-p measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the peak-to-peak voltage of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:VPP command prior to the MEASure:COMPare command when configuring for a peak-to-peak voltage limit test.
- **Related Commands:** MEASure:VPP?, VMAX, VMIN, COMPare, LIMittest, RESults?, SOURce.

:VPP?

MEASure:VPP? turns continuous measurement mode off, performs a Peak-to-Peak Voltage measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example **Perform peak-to-peak voltage measurement on waveform memory 4 and return results**

```
dimension statement         String for data
MEAS:SOUR WMEM4             Measure waveform memory 4
MEAS:VPP?                   Perform Vp-p measurement, query
                            instrument to return measurement
                            results
enter statement             Enter measurement results into
                            computer
```

Comments

- **Oscilloscope Setup:** Peak-to-Peak Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Measurement Method:** The method the instrument uses to determine peak-to-peak voltage is to measure VMAX and VMIN, then calculate voltage amplitude as follows:

  peak-to-peak voltage = VMAX - VMIN
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing measured peak to peak voltage (in volts), with 0 volts as the reference.

:VRELative

MEASure:VRELative ***<percent>*** is used to move the voltage markers (Vmarkerl and Vmarker2) to the specified *percentage* points of their last established position. The location of the voltage markers may not necessarily be on the waveform currently presented.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Unit
percent	numeric	0 to 100	PCT

Example

Move voltage markers

For the following examples, the current position of Vmarkerl is at the base (0%) of the signal and VMarker2 is at the top (100%).

After execution of this example, VMarkerl will move to the 10% level and VMarker2 to the 90% level of the signal.

```
MEAS:VREL  10            Move voltage markers 10%
```

After execution of this example, VMarkerl will move to the 20% level and VMarker2 to the 80% level of the signal.

```
MEAS:VREL  20            Move voltage markers 20%
```

Comments

- **Entering Percent:** Any value between 0 and 100 can be used, however the markers cannot cross positions.

  If 0 is sent, the markers are not moved.

  If values of ≤50 are sent, both markers are moved the amount specified up to 50%.

  If values of >50 are sent, both markers are moved the specified amount – 50%. This is because VMarkerl is always in the range of 0% to 50% and Vmarker2 is in the range of 50% to 100%. For example, if 60% is entered, both markers are moved 10% (60% – 50% = 10%).

- **VRELative versus VSTArt and VSTOp:** Both commands are used to specify voltage marker positions.

  MEASure:VSTArt and VSTOp commands are used to position the voltage markers to a specified voltage level. Each is specified separately, and knowledge of the current marker positions is not necessary.

  MEASure:VRELative command moves both voltage markers a specific *percentage* point from their last established position. The starting position of the markers must be known for this command to be meaningful. The markers can be set to a known position on the selected waveform using the MEASure:VAMPlitude? query (set to 0% and 100%).

- **Starting Marker Position:** When the instrument performs a voltage measurement, the voltages markers are automatically moved to perform the measurement, then read. Because of this, the markers can be set to a known position on the selected waveform using the measure voltage commands. For example, use the MEASure:VAMPlitude? query to set the markers to 0% and 100%.
- **Upper and Lower Threshold:** The VRELative command does not affect the UPPer and LOWer threshold levels as selected by the MEASure:UPPer and LOWer commands.
- **Related Commands:** MEASure:VSTArt, VSTOp.

:VRELative?

MEASure:VRELative? returns the current relative position of the voltage stop marker (Vmarker2). The value is sent to the output buffer. The number returned specifies the position in percent from the last established position of the voltage stop marker.

Example

Query the voltage stop marker relative position

`dimension statement`	*String for data*
`MEAS:VREL 10`	*Move voltage markers 10%*
`MEAS:VREL?`	*Query instrument to return voltage stop marker relative position (90%)*
`enter statement`	*Enter data into computer*

:VRMS

MEASure:VRMS is used to place the instrument in the continuous measurement mode and start a RMS Voltage measurement.

Example

Start a RMS Voltage measurement

`MEAS:VRMS`	*Start a RMS Voltage measurement*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the RMS voltage of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **VACRms versus VRMS:** The MEASure:VACRms command is identical to the MEASure:VRMS command.
- **Related Commands:** MEASure:VRMS?, VACRms, SOURce, RESults?.

:VRMS?

MEASure:VRMS? turns continuous measurement mode off, performs a RMS Voltage measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example

Perform RMS amplitude measurement on channel 2 and return results

`dimension statement`	*String for data*
`MEAS:SOUR CHAN2`	*Measure channel 2*
`MEAS:VRMS?`	*Perform RMS Voltage measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** RMS Voltage measurement is made using the first cycle present. If a complete cycle is not present, all data points are averaged.
- **Measurement Method:** The method the instrument uses to determine RMS voltage (ACRMS) is to measure VAVerage, subtract it from each data point, then calculate RMS voltage.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing measured RMS voltage (in volts), with 0 volts as the reference.

:VSTArt

MEASure:VSTArt ***<voltage>*** is used to position the voltage start marker (VMarker1) at a specified voltage with respect to 0 volts.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
voltage	numeric	–4E+10 to +4E+10	V

Example

Set voltage start marker to 5 volts

```
MEAS:VSTA  5                Set Vmarker1 to 5 volts
```

Comments

- **Selecting Voltage:** The desired voltage is specified using 0 volts as the reference. Positive and negative values are acceptable. If *voltage* is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.
- **Related Commands:** MEASure:VSTOp, VDELta?.

:VSTArt?

MEASure:VSTArt? returns the current position of the voltage start marker (VMarker 1). The value is sent to the output buffer. The number returned specifies the position of the voltage start marker (in volts) from 0 volts.

Example

Query the voltage start marker

```
dimension  statement        String for data
MEAS:VSTA  5                Set Vmarker1 to 5 volts
MEAS:VSTA?                  Query instrument to return voltage
                            start marker
enter  statement            Enter data into computer
```

:VSTOp

MEASure:VSTOp ***<voltage>*** is used to position the voltage stop marker (VMarker2) at a specified voltage with respect to 0 volts.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
voltage	numeric	–4E+10 to +4E+10	V

Example

Set voltage stop marker to –5 volts

```
MEAS:VSTO  -5          Set Vmarker2 to –5 volts
```

Comments

- **Selecting Voltage:** The desired voltage is specified using 0 volts as the reference. Positive and negative values are acceptable. If *voltage* is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.
- **Related Commands:** MEASure:VSTArt, VDELta?.

:VSTOp?

MEASure:VSTOp? returns the current position of the voltage stop (VMarker 2) marker. The value is sent to the output buffer. The number returned specifies the position of the voltage stop marker (in volts) from 0 volts.

Example

Query the voltage start marker

```
dimension  statement   String for data
MEAS:VSTO  -5          Set Vmarker2 to –5 volts
MEAS:VSTO?             Query instrument to return voltage
                       stop marker
enter  statement       Enter data into computer
```

:VTIMe?

MEASure:VTIMe? ***<time>*** returns the voltage level at a specified time. The time is referenced to the trigger event and must be on the acquired waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
time	numeric	±9.99999E+37	S

Example

Measure voltage on waveform 100μsec from trigger event

```
MEAS:VTIM?  100E-6     Set to measure voltage 100μsec from
                       trigger event
```

Comments

- **Oscilloscope Setup:** In order to perform measurement, the voltage measurement point must be present.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing volts (referenced to 0 volts), at the time specified (referenced to the trigger event).

:VTOP

MEASure:VTOP is used to place the instrument in the continuous measurement mode and start a Top Voltage measurement.

Example **Start a Top Voltage measurement**

`MEAS:VTOP`	*Start a Top Voltage measurement*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting Source:** One source is specified using the MEASure:SOURce command.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the voltage value at the top of the source specified.
- **Reading Measurement Results:** Use the MEASure:RESults? query to return measurement results without stopping the measurement.
- **Limit Test:** Execute the MEASure:VTOP command prior to the MEASure:COMPare command when configuring for a top voltage limit test.
- **Related Commands:** MEASure:VTOP?, COMPare, LIMittest, RESults?, SOURce.

:VTOP?

MEASure:VTOP? turns continuous measurement mode off, performs a Top Voltage measurement one time on the signal present, and then sends the measurement results to the output buffer.

Example **Perform top voltage measurement on channel 2 and return results**

`dimension statement`	*String for data*
`MEAS:SOUR CHAN2`	*Measure channel 2*
`MEAS:VTOP?`	*Perform Top Voltage measurement, query instrument to return measurement results*
`enter statement`	*Enter measurement results into computer*

Comments

- **Oscilloscope Setup:** Top Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Returned Format:** The measurement results are returned as a numeric value (SYSTem:HEADer to OFF) representing measured top voltage (in volts), with 0 volts as the reference.

MEMory

The MEMory command subsystem enables the Oscilloscope to use external A24 VME memory space for routing acquired data from the internal buffers to the external buffer. It also enables measurements to be performed using data from the internal buffer or external buffer.

Subsystem Syntax

```
MEMory
   :VME
      :ADDRess <address>
      :ADDRess? [MINimum|MAXimum]
      :MEASure
         :ADDRess <address>
         :ADDRess? [MINimum|MAXimum]
         :STATe <mode>
         :STATe?
      :SIZE <bytes>
      :SIZE?
      :STATe <mode>
      :STATe?
```

The MEMory command allows the Oscilloscope to use external A24 VME memory for data acquisition, when the VME STATe is ON. Once the data has been acquired, it is available in "raw" format. Data is processed depending on type and format selected using the following guidelines:

- **ACQuire:TYPe NORMal** — The acquired data is 16 bits in length. The starting location, in VME space, of each INPUT buffer is to be calculated as follows:

```
INPUT1 (CHANnel1) = MEM:VME:ADDR
INPUT2 (CHANnel2) = MEM:VME:ADDR + 8192
INPUT3 (CHANnel3) = MEM:VME:ADDR + 16384
INPUT4 (CHANnel4) = MEM:VME:ADDR + 24576
```

Data — 16 bits

```
1111110000000000
5432109876543210
xxxxxxxxxxxxxxxx
```

```
         BYTE                      WORD                     COMPRESSED
15                   0  15                    0  15                          0
xxxxxxx--------->  xxxxxxxxxxxxxxxx>  xxxxxxxxxx-------->
   shift right 9 bits         shift right 1 bit          shift right 8 bits
```

*NOTE: x = bits to shift. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the WAVeform:PREamble? query.

- **ACQuire:TYPe AVERage** — The acquired data is 32 bits in length. The starting location, in VME space, of each INPUT buffer is to be calculated as follows:

  INPUT1 (CHANnel1) = MEM:VME:ADDR
  INPUT2 (CHANnel2) = MEM:VME:ADDR + 8192
  INPUT3 (CHANnel3) = MEM:VME:ADDR + 16384
  INPUT4 (CHANnel4) = MEM:VME:ADDR + 24576

Data — 32 bits

```
                  33222222222211111111110000000000
                  10987654321098765432109876543210
                          xxxxxxxxxxxxxxxx

        BYTE                    WORD                    COMPRESSED
15                 0  15                  0  15                        0
xxxxxxx--------->  xxxxxxxxxxxxxxx>  xxxxxxxxxx-------->
   shift right 9 bits      shift right 1 bit        shift right 8 bits
```

*NOTE: x = bits to shift. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the WAVeform:PREamble? query.

- **ACQuire:TYPe ENVelope** — The acquired data is 16 bits in length. The starting location, in VME space, of each INPUT buffer is be to calculated as follows:

If data is digitized:

MINIMUM

INPUT1 (CHANnel1) = MEM:VME:ADDR
INPUT2 (CHANnel2) = MEM:VME:ADDR + 8192
INPUT3 (CHANnel3) = MEM:VME:ADDR + 16384
INPUT4 (CHANnel4) = MEM:VME:ADDR + 24576

MAXIMUM

INPUT1 (CHANnel1) = MEM:VME:ADDR + (2 * ACQuire:POINts)
INPUT2 (CHANnel2) = MEM:VME:ADDR + 8192 + (2 * ACQuire:POINts)
INPUT3 (CHANnel3) = MEM:VME:ADDR + 16384 + (2 * ACQuire:POINts)
INPUT4 (CHANnel4) = MEM:VME:ADDR + 24576 + (2 * ACQuire:POINts)

If data is NOT digitized:

MINIMUM

INPUT1 (CHANnel1) = MEM:VME:ADDR
INPUT2 (CHANnel2) = MEM:VME:ADDR + 8192
INPUT3 (CHANnel3) = MEM:VME:ADDR + 16384
INPUT4 (CHANnel4) = MEM:VME:ADDR + 24576

MAXIMUM

INPUT1 (CHANnel1) = MEM:VME:ADDR + 1002
INPUT2 (CHANnel2) = MEM:VME:ADDR + 8192 + 1002
INPUT3 (CHANnel3) = MEM:VME:ADDR + 16384 + 1002
INPUT4 (CHANnel4) = MEM:VME:ADDR + 24576 + 1002

Data — 16 bits

```
                         1111110000000000
                         5432109876543210
                         xxxxxxxxxxxxxxxx

        BYTE                     WORD                  COMPRESSED
15                0  15                  0  15                    0
xxxxxxx--------->    xxxxxxxxxxxxxxxx>      xxxxxxxxxx-------->
   shift right 9 bits     shift right 1 bit      shift right 8 bits
```

*NOTE: x = bits to shift. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the WAVeform:PREamble? query.

:VME:ADDRess

MEMory:VME:ADDRess ***<address>*** sets the address of the external memory board in A24 memory address space where acquisition data will be available. *address* must be on an even boundary or a settings conflict will be generated.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
address	numeric	2097152-14647294| #H200000-#HDF7FFE|MIN|MAX	none

Example **Setting the VME memory address**

```
MEM:VME:ADDR  #H250000
```
Set memory address location

Comments

- **Entering Address:** Address location can be specified in:

  Decimal or hexadecimal (#H....)

  MIN – sets the address to 2097152 (#H200000)

  MAX – sets the address to 14647294 (#HDF7FFE).

- ***RST Condition:** MEM:VME:ADDR #H200000

:VME:ADDRess?

MEMory:VME:ADDRess? [MINimum|MAXimum] returns one of the following numbers to the output buffer:

- The present hexadecimal address selected if MIN or MAX are not specified.
- The lowest hexadecimal address available (H200000) if MIN is specified.
- The highest hexadecimal address available (HDF7FFE) if MAX is specified.

Example **Querying the VME memory address**

```
dimension  statement     Dimension computer string array
MEM:VME:ADDR  #H250000   Set memory address location
MEM:VME:ADDR?            Query instrument to return
                         memory address (in decimal)
enter  statement         Enter string into computer
```

:VME:MEASure:ADDRess

MEMory:VME:MEASure:ADDRess ***<address>*** sets the address of the external memory board in A24 memory address space where measurement data will be available. *address* must be on an even boundary or a settings conflict will be generated.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
address	numeric	2097152-14647294| #H200000-#HDF7FFE|MIN|MAX	none

Example **Setting the VME memory measure address**

```
MEM:VME:ADDR  #H250000
```
Set memory address location

Comments

- **Entering Address:** Address location can be specified in:

  Decimal or hexadecimal (#H....)

  MIN – sets the address to 2097152 (#H200000)

  MAX – sets the address to 14647294 (#HDF7FFE).

- ***RST Condition:** MEM:VME:ADDR #H200000

:VME:MEASure:ADDRess?

MEMory:VME:MEASure:ADDRess? [MINimum|MAXimum] returns one of the following numbers to the output buffer:

- The present hexadecimal address selected if MIN or MAX are not specified.
- The lowest hexadecimal address available (H200000) if MIN is specified.
- The highest hexadecimal address available (HDF7FFE) if MAX is specified.

Example **Querying the VME memory measure address**

`dimension statement`	*Dimension computer string array*
`MEM:VME:ADDR #H250000`	*Set memory address location*
`MEM:VME:ADDR?`	*Query instrument to return memory address (in decimal)*
`enter statement`	*Enter string into computer*

:VME:MEASure:STATe

MEMory:VME:MEASure:STATe ***<mode>*** enables or disables use of the external memory board in A24 memory where data can be used for making a measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	OFF|0|ON|1	none

Example

Enabling VME measure memory

`MEM:VME:MEAS:ADDR`	`#H250000`	*Set the measurement memory address location*
`MEM:VME:MEAS:STAT`	`ON`	*Enable use of external measurement data*

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Memory VME State:** When MEMory:VME:STATe is set to ON, MEMory:VME:MEASure:STATe will automatically be set to ON.
- ***RST Condition:** MEM:VME:MEAS:STAT defaults to OFF.

:VME:MEASure:STATe?

MEMory:VME:MEASure:STATe? returns the current state of the VME measurement memory. ON is returned when the external memory board in A24 memory is being used for measurements. OFF is returned if the internal buffers are being used. The value is sent to the output buffer.

Example

Querying the VME measure memory state

`MEM:VME:MEAS:STAT 1`	*Enables use of external memory card*
`MEM:VME:MEAS:STAT?`	*Query instrument to return external memory state*
`enter statement`	*Enter value into computer*

:VME:SIZE

MEMory:VME:SIZE *<bytes>* sets the size, in bytes, of the external VME memory card.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
bytes	numeric	#8000 through #HC00000	bytes

Example

Setting the VME memory size

```
MEM:VME:SIZE  64000
```
Set memory size to 64 kBytes

Comments

- **Entering Size:** Memory size can be specified in decimal or hexadecimal (#H....).
- **Minimum Memory Required:** A minimum of H8000 bytes of VME memory are required to use the external VME feature.
- ***RST Condition:** MEM:VME:SIZE defaults to H8000.

:VME:SIZE?

MEMory:VME:SIZE? returns the current external VME memory allocation (in hexadecimal) to the output buffer.

Example

Querying the VME memory size

`MEM:VME:SIZE  64000`	*Set memory size to 64 kBytes*
`MEM:VME:SIZE?`	*Query instrument to return memory size*
`enter statement`	*Enter string into computer*

:VME:STATe

MEMory:VME:STATe *<mode>* enables or disables use of an external VME memory card for acquisition data storage.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	OFF|0|ON|1	none

Example

Enabling VME memory

```
MEM:VME:ADDR  #H250000   Set memory address location
MEM:VME:SIZE  64000      Set memory size to 64 kBytes
MEM:VME:STAT  ON         Enable use of external memory
                         card
```

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Memory VME Measure State:** When MEMory:VME:STATe is set to ON, MEMory:VME:MEASure:STATe will automatically be set to ON.
- ***RST Condition:** MEM:VME:STAT defaults to OFF.

:VME:STATe?

MEMory:VME:STATe? returns whether the external VME memory feature is enabled (ON) or disabled (OFF). The value is sent to the output buffer.

Example

Querying the VME memory state

```
MEM:VME:STAT  ON        Enables use of external memory card
MEM:VME:STAT?           Query instrument to return external
                        memory state
enter  statement        Enter value into computer
```

OUTPut

The OUTPut command subsystem selects the source of the output trigger generated when the Oscilloscope generates an internal trigger event. The selected output can be enabled, disabled, and queried. The three available outputs are the ECL Trigger bus (lines 0 to 1), TTL Trigger bus (lines 0 to 7), or the "TTL Trigger Output" BNC port on the Oscilloscope front panel.

OUTPut[:STATe] acts like the master switch for the OUTPut subsystem. If the ECLTrg, TTLTrg, or EXTernal states are on, an output will **ONLY** occur when the OUTPut[:STATe] is set to ON.

Subsystem Syntax

```
OUTPut
    :ECLTrg<number>
        [:STATe] <mode>
        [:STATe]?
    :EXTernal
        [:STATe] <mode>
        [:STATe]?
    [:STATe]
    [:STATe]?
    :TTLTrg<number>
        [:STATe] <mode>
        [:STATe]?
```

:ECLTrg[:STATe]

OUTPut:ECLTrg<*number*>[:STATe] <*mode*> selects and enables which ECL Trigger bus line (0 or 1) will output a trigger when the Oscilloscope triggers. It is also used to disable a selected ECL Trigger bus line. *number* specifies the ECL Trigger bus line (0 or 1). *mode* enables (ON|1) or disables (OFF|0) the specified bus line.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	0 or 1	none
mode	boolean	ON|OFF|1|0	none

Example

Enabling ECL trigger bus line 0

```
OUTP:ECLT0:STAT  1      Enable ECL Trigger bus line 0
OUTP  1                 Enable output subsystem
```

Comments

- **Enabling ECL Trigger bus:** When enabled, a pulse is output to the selected ECL Trigger bus line (0 or 1) when the Oscilloscope triggers. If disabled, a pulse is not output. The output is a positive going pulse.
- **Numerous outputs selected at a time:** All outputs (ECLTrg 0 or 1; TTLTrg 0, 1, 2, 3, 4, 5, 6, or 7; and EXTernal) can be enabled at one time. Before any output is enabled, both the individual trigger state and the output state must be enabled. For example, to enable TTLTrg1 and ECLTrg2, send the following commands:

```
OUTP:TTLT1 ON
OUTP:ECLT2 ON
OUTP ON
```

- **Related Commands:** TRIGger subsystem.
- ***RST Condition:** Default is OFF.

:ECLTrg[:STATe]?

OUTPut:ECLTrg<*number*>[:STATe]? queries the present state of the specified ECL Trigger bus line. The query returns ON if the specified bus line is enabled or OFF if the specified bus line is disabled. The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	0 or 1	none

Example

Query ECL trigger bus line 0 state

```
dimension  statement    String for data
OUTP:ECLT0:STAT  1      Enable ECL Trigger bus line 0
OUTP:ECLT0?             Query instrument to return  ECL line
                        0 bus enable state
enter  statement        Enter result into computer
```

:EXTernal[:STATe]

OUTPut:EXTernal[:STATe] ***<mode>*** enables or disables the "TTL Trigger Output" BNC port on the Oscilloscope Module to output a trigger when the Oscilloscope triggers. *mode* enables (ON | 1) or disables (OFF | 0) the BNC port.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	ON|OFF|1|0	none

Example **Enabling TTL trigger output BNC port**

```
OUTP:EXT  1        Enable "TTL Trigger Output" BNC
                   port to output pulse
OUTP  1            Enable output subsystem
```

Comments

- **Enabling Trig Out Port:** When enabled, a pulse is output from the "TTL Trigger Output" BNC port on the Oscilloscope Module. The output is a negative going pulse.
- **Numerous outputs selected at a time:** All outputs (ECLTrg 0 or 1; TTLTrg 0, 1, 2, 3, 4, 5, 6, or 7; and EXTernal) can be enabled at one time. Before any output is enabled, both the individual trigger state and the output state must be enabled. For example, to enable TTLTrg1 and ECLTrg2, send the following commands:

```
OUTP:TTLT1 ON
OUTP:ECLT2 ON
OUTP ON
```

- **Related Commands:** TRIGger subsystem.
- ***RST Condition:** Defaults to OFF.

:EXTernal[:STATe]?

OUTPut:EXTernal:STATe? queries the present state of the "TTL Trigger Output" BNC port. The query returns ON if the port is enabled or OFF if the port is disabled. The value is sent to the output buffer.

Example **Query TTL Trigger Output BNC Port Enable State**

```
dimension  statement   String for data
OUTP:EXT  ON           Enable "TTL Trigger Output" BNC
                       port
OUTP:EXT?              Query instrument to return  port
                       enable state
enter  statement       Enter value into computer
```

[:STATe]

OUTPut[:STATe] <*mode*> enables or disables the OUTPut subsystem. *mode* enables (ON | 1) or disables (OFF | 0) all selected TTLTrg, ECLTrg, and EXTernal outputs.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	ON|OFF|1|0	none

Example

Enabling Trig Out BNC Port

`OUTP:EXT 1`	*Enable "TTL Trigger Output" BNC port to output pulse*
`OUTP 1`	*Enable output subsystem*

Comments

- **Selecting Outputs:** Use the TTLTrg, ECLTrg, or EXTernal commands to enable a specific output. Use the OUTPut[:STATe] command to enable the subsystem.
- ***RST Condition:** Defaults to OFF.

[:STATe]?

OUTPut[:STATe]? queries the present state of the OUTPut subsystem. The query returns ON if the output is enabled or OFF if the output is disabled. The value is sent to the output buffer. See [:STATe] command for more information.

Example

Query output subsystem

`dimension statement`	*String for data*
`OUTP 1`	*Enable "Trigger Output" BNC port*
`OUTP?`	*Query instrument to return port enable state*
`enter statement`	*Enter value into computer*

:TTLTrg[:STATe]

OUTPut:TTLTrg<*number*>[:STATe] <*mode*> selects and enables which TTL Trigger bus line (0 to 7) will output a trigger when the Oscilloscope triggers. It is also used to disable a selected TTL Trigger bus line. *mode* enables (ON | 1) or disables (OFF | 0) the specified bus line.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number *mode*	numeric boolean	0 to 7 ON|OFF|1|0	none none

Example

Enabling TTL Trigger bus line 7

```
OUTP:TTLT7:STAT  1       Enable TTL Trigger bus line 7 to
                         output pulse
OUTP  1                  Enable output subsystem
```

Comments

- **Enabling TTL Trigger bus:** When enabled, a pulse is output to the selected TTL Trigger bus line (0 to 7) after the Oscilloscope triggers. If disabled, a pulse is not output. The output is a negative going pulse.
- **Numerous outputs selected at a time:** All outputs (ECLTrg 0 or 1; TTLTrg 0, 1, 2, 3, 4, 5, 6, or 7; and EXTernal) can be enabled at one time. Before any output is enabled, both the individual trigger state and the output state must be enabled. For example, to enable TTLTrgl and ECLTrg2, send the following commands:

```
OUTP:TTLT1 ON
OUTP:ECLT2 ON
OUTP ON
```

- **Related Commands:** TRIGger subsystem.
- ***RST Condition:** Defaults to OFF.

:TTLTrg[:STATe]?

OUTPut:TTLTrg<*number*>[:STATe]? queries the present state of the specified TTL Trigger bus line. The query returns ON if the specified bus line is enabled or OFF if the specified bus line is disabled. The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	0 or 7	none

Example

Query TTL trigger bus line 7 state

```
dimension  statement     String for data
OUTP:TTLT7:STAT  1       Enable TTL Trigger bus line 7
OUTP:TTLT7?              Query instrument to return TTL bus
                         line 7 enable state
enter  statement         Enter value into computer
```

Root Level Commands

The Root Level command subsystem is a fictitious subsystem used to group all the single commands that do not belong to any other subsystem. These commands control many of the basic operations and special features of the Oscilloscope.

"**ROOT**" is **NOT** a command and **MUST NOT** precede the commands listed in this section, or an error will be generated.

Subsystem Syntax

AUToscale
BLANk *<source>*[,*<source>*[,*<source>*[,*<source>*]]]
BNC *<output>*
BNC?
DIGitize *<source>*[,*<source>*[,*<source>*[,*<source>*]]]
ERASe *<source>*
LTER?
MERGe *<location>*
RUN
RUN?
SERial *<string>*
STATus? *<source>*
STOP
STORe *<source>*,*<destination>*
TER?
VIEW *<source>*[,*<source>*[,*<source>*[,*<source>*]]]

AUToscale

The AUToscale command causes the instrument to evaluate all input signals, and then set the correct conditions to present the signals.

Subsystem Syntax

AUToscale

Example

Execute an autoscale

```
AUT                              Initiate an autoscale
```

Comments

- **Controls Affected:** The following controls are set to present the input signals:

  All markers to off
  All memories to OFF
  CHANnel:OFFSet as required
  CHANnel:RANGe as required
  DISPlay:PERSistence to SINGle
  FUNCtion to OFF
  MEASure to OFF
  TIMebase:RANGe as required
  TIMebase:WINDow to OFF
  TRIGger:HOLDoff as required
  TRIGger:LEVel as required
  TRIGger:MODe to EDGe
  TRIGger:SLOPe as required

- **More than One Input Signal:** If signals are present on more than one input, the sweep will be triggered on the signal closest to channel 1. If a signal is not present on channel 1, then the instrument will be triggered on channel 2. If a signal is not present on channel 2, then the instrument will be triggered on channel 3, etc.
- **No Input Signal:** If no signals are found on any input, the instrument will be returned to its former state.
- **Channel Coupling:** If a large offset is present on the input signal, coupling may change from DC to AC.

BLANk

The BLANk command is used to turn off or stop presenting the specified CHANnel<*n*>, FUNCtion<*n*>, PMEMory<*n*>, or WMEMory<*n*>.

Subsystem Syntax BLANk <*source*>[,<*source*>[,<*source*>[,<*source*>]]]

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	CHANnel*n* (*n*=1 to 4) FUNCtion*n* (*n*=1 to 2) WMEMory*n* (*n*=1 to 4) PMEMory*n* (*n*=0 to 2)	None

Example **Blank Channel 3**

```
BLAN  CHAN3          Stop presenting channel 3
```

Comments

- **Erasing Memory:** Executing the BLANk command does not erase the the contents of pixel or waveform memory. See ERASe command for information on erasing pixel memory contents.
- **Start Presenting Waveform:** Use the VIEW command to start presenting the specified channel, function, pixel memory, or waveform memory.
- **Related Commands:** VIEW, ERASe.

BNC

BNC *<output>* selects either PROBe or TRIGger as the output mode of the Probe Compensation AC Calibrator Output BNC connector. PROBe outputs a square wave signal at ≈1.5 kHz, and TRIGger outputs a rising edge when an internal trigger occurs.

Subsystem Syntax BNC *<output>*

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
output	discrete	PROBe|TRIGger	none

Example **Set BNC connector to trigger**

```
BNC  TRIG                  AC Calibrator output to trigger
```

Comments

- ***RST Condition:** Defaults to PROBe.

BNC?

BNC? returns the currently selected (PROBe or TRIGger) Probe Compensation AC Calibrator Output BNC connector output signal. Returns PROBe if output is a square wave signal, and TRIGger if output is a rising edge when an internal trigger occurs. The data is sent to the output buffer. See BNC command for more information on available signals.

Subsystem Syntax BNC?

Example **Querying BNC connector**

```
Dimension  statement       String for data
BNC  TRIG                  AC Calibrator output to trigger
BNC?                       Query instrument to return BNC
                           selection
enter  statement           Enter value into computer
```

DIGitize

The DIGitize command is used to acquire waveform data present on channels 1 to 4 to ensure that all measurements are performed using the same data, and that the data obtained is valid. Digitized data can also be moved into waveform memory or transferred over the bus. It causes an acquisition to take place on the specified channel(s) with the resulting digitized data being placed in the channel buffer, then the acquisition is stopped.

Subsystem Syntax

DIGitize *<source>*[,*<source>*[,*<source>*[,*<source>*]]]

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	numeric	CHANnel*n* (*n*=1 to 4)	none

Example

Digitize waveform data present on channel 2

The following example illustrates the use of the DIGitize command only. Chapter 3 contains an example on performing a complete digitizing operation.

```
DIG  CHAN2                 Channel 2 waveform data digitized
```

Comments

- **Selecting Channel:** Up to four channels can be digitized using a single command; however, because channels 1-2 and 3-4 are paired to share A/D Converters, the digitize sequence is dependent on the channel(s) selected.

  Channels 1-2 (2-1) or 3-4 (4-3): First channel digitized when trigger requirements are satisfied. The second channel is digitized only when another trigger is received.

  Any other Channel Combinations: Both channels digitized simultaneously when trigger requirements satisfied.

  See Appendix C, Optimizing Measurements, for additional information on digitizing operations.

- **Before Digitizing:** Before the waveform data can be digitized (using DIGitize command), set up conditions such as TYPe, number of POINts, and the COUNt must be selected. See the ACQuire subsystem for more information on selecting these commands.

- **After Digitizing:** After the waveform data has been digitized (using DIGitize command), the waveform DATa is placed in the channel buffer of the specified source where it can be read. See the WAVeform subsystem for more information on selecting these commands.

- **Loss of Digitized Data:** When the DIGitize command is complete the instrument is placed in the stopped mode. When restarted (RUN command), the digitized data stored in the channel buffers will be overwritten. Before executing the RUN command, verify all operations that require the digitized data are completed.

  Unused Channels: Executing the DIGitize command will turn off any unused channels.

- **Stopping a Digitize:** Send the device clear (e.g. CLEAR 707) command to stop a digitize in process.
- **Digitizing Speed:** The speed of the total digitize operations may be improved if two or more DIGitize commands are sent without changing other parameters.
- **Related Commands:** ACQuire:, WAVeform:, RUN.

ERASe

The ERASe command erases a specified pixel memory.

Subsystem Syntax

ERASe *<source>*

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
source	discrete	PMEMory*n* (*n*=0 to 2)	none

Example

Erase contents of pixel memory 2

```
ERASe  PMEM2            Erase pixel memory 2
```

Comments

- **Erasing Pixel MEMory0:** If the instrument is running and being triggered when ERASe PMEMory0 is executed, the instrument will momentarily stop acquiring data, clear the contents of pixel memory 0, and then continue with data acquisition.
- **Erasing Pixel MEMory1-2:** When ERASe PMEMory1 or 2 is executed, the instrument will clear the specified pixel memory. Once the command is executed, there is no way to retrieve the original data.
- **Related Commands:** BLANk, VIEW.

LTER?

The Limit Test Event Register (LTER?) query is used to return the Limit Test Event Register — Limit Test Fail bit. This bit is set (1) when the limit test has failed. After the Limit Test Event Register is queried, it is cleared (0). The value is sent to the output buffer.

Subsystem Syntax LTER?

Example **Query the limit test event register—limit test fail bit**

`LTER?`	*Query instrument to return LTER limit test fail bit*
`enter statement`	*Enter value into computer*

Comments

- **SRQ:** A Service Request (SRQ) can only be generated when the bit transitions from 0 to 1. The bit must be cleared (0) each time you would like a new Service Request to be generated.
- **Related Commands:** *STB?.

MERGe

The MERGe command is used to "merge" the current contents of pixel memory 0 (active waveform) with the current contents of pixel memory 1 or 2. Result is retained in pixel memory number specified.

Subsystem Syntax

MERGe <*location*>

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
location	discrete	PMEMory*n* (*n*=1 or 2)	none

Example

Merge pixel memory 0 and pixel memory 2, and retain results in pixel memory 2

```
VIEW  PMEM0          Enable pixel memory 0
MERG  PMEM2          Merge pixel memory 0 and pixel
                     memory 2
```

Comments

- **Merge Result:** The merge result is retained in the pixel memory number specified. The original contents of the specified pixel memory (prior to MERGe) are not retained.
- **Pixel Memory 0:** Before any data can be merged into pixel memory 1 or 2, pixel memory 0 must be set to on.
- **Related Commands:** BLANk, VIEW, ERASe.

RUN

RUN is used to start acquiring data for the active waveform.

Subsystem Syntax

RUN

Example

Start acquiring data

```
RUN                          Acquire data
```

Comments

- **RUN versus TIMebase:MODe:** The data acquisition is defined by the selected TIMebase:MODe.

  TIMebase:MODe SINGle - executing RUN enables the trigger once and saves the acquired data.

  TIMebase:MODe AUTO or TRIGGERED - executing RUN enables the trigger repeatedly and saves the acquired data continuously.

- **Stop Acquiring Data:** Use the STOP command to stop data acquisition in pixel memory 0.
- **Related Commands:** STOP, TIMebase:MODe.

RUN?

RUN? returns a number to show the current acquisition state. "1" = RUN, "0" = STOP. The value is sent to the output buffer.

Subsystem Syntax

RUN?

Example

Query acquisition state

```
RUN                          Acquire data
RUN?                         Query instrument to return
                             acquisition state
enter  statement             Enter results into computer
```

Comments

- **STOP Command:** This is also the query for the STOP command.

SERial

The SERial command is used to enter a serial number in the instrument. As the instrument serial number is entered at the factory, do not use this command unless there is a need to serialize the instrument for a different application.

Subsystem Syntax

SERial <*string*>

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
string	numeric	Alpha-numeric, no special	none

Example

Enter a different serial number

```
SER  "1234A56789"
```
Different serial number

Comments

- **Entering Serial Number *string*:** Serial number consists of 10 alpha-numeric digits enclosed in quotes (").
- **Protect Switch:** The calibration protection switch must be set to the unprotected position to write a new serial number to the protected non-volatile ram within the instrument.
- **Serial Number versus *IDN?:** The serial number is part of the string returned for the *IDN? query.

STATus

The STATus? query returns a number to indicate whether a CHANnel<*n*>, FUNCtion<*n*>, WMEMory<*n*>, or PMEMory<*n*> is ON or OFF. "1" = ON, "0" = OFF. The value is sent to the output buffer.

Subsystem Syntax

STATus? <*source*>

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	CHANnel*n* (*n*=1 to 4)| FUNCtion*n* (*n*=1 to 2)| WMEMory*n* (*n*=1 to 4)| PMEMory*n* (*n*=1 to 2)	none

Example

Query CHANnel 1 status

```
STAT?  CHAN1              Query instrument to return channel 1
                          status
enter  statement          Enter results into computer
```

Comments

- **Changing the Status:** The STATus? query does not change the present status of a channel, function, waveform memory, or pixel memory.

STOP

The STOP command is used to stop acquiring data for the active waveform.

Subsystem Syntax STOP

Example **Stop acquiring data**

```
STOP                          Stop acquiring data
```

Comments

- **Restart Acquiring Data:** Use the RUN command to start data acquisition.
- **Related Commands:** RUN, RUN?.
- **STOP Query:** Use the RUN? query to return acquisition status.

STORe

The STORe command is used to move a previously stored waveform, CHANnel<*n*>, or FUNCtion<*n*> to a WMEMory<*n*> location.

Subsystem Syntax

STORe *<source>,<destination>*

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
source	discrete	CHANnel*n* (*n*=1 to 4)| FUNCtion*n* (*n*=1 to 2)| WMEMory*n* (*n*=1 to 4)	none
destination		WMEMory*n* (*n*=1 to 4)	none

Example

Move the CHANnel 2 waveform to WMEMory 3

```
STOR   CHAN2,WMEM3
```
Channel 2 waveform stored in waveform memory 3

Comments

- **Selecting Source:** The source can be specified as any channel, function, or waveform memory. Pixel memories cannot be stored.

  Selecting Destination: The destination of the waveform can only be waveform memory 1 through 4. When executed, the current contents of the specified WMEMory<*n*> will be overwritten.

- **Related Commands:** BLANk, VIEW.

TER?

The Trigger Event Register (TER?) query is used to return the Trigger Event Register. Bit 0 is set (1) when a trigger has occurred. Bit is not set (0) if a trigger has not occurred, or if a trigger event is not found and the sweep is auto-triggering. After the Trigger Event Register is queried, it is cleared (0). The value is sent to the output buffer.

Subsystem Syntax

TER?

Example

Query the trigger event register

```
TER?                       Query instrument to return trigger
                           event register
enter  statement           Enter value into computer
```

Comments

- SRQ: A Service Request (SRQ) can only be generated when the bit transitions from 0 to 1, therefore the bit must be cleared each time you would like a new Service Request to be generated.
- **Related Commands:** *STB?.

VIEW

The View command causes the instrument to turn on an active CHANnel<*n*>, FUNCtion<*n*>, PMEMory<*n*>, or WMEMory<*n*>.

Subsystem Syntax VIEW <*source*>[,<*source*>[,<*source*>[,<*source*>]]]

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	CHANnel*n* (*n*=1 to 4)| FUNCtion*n* (*n*=1 to 2)| WMEMory*n* (*n*=1 to 4)| PMEMory*n* (*n*=0 to 2)	None

Example **View Channel 3**

```
VIEW  CHAN3                Start presenting channel 3
```

Comments

- **Stop Presenting Waveform:** Use the BLANk command to turn off the specified channel, function, pixel memory, or waveform memory.
- **Unused Channels:** BLANk all channels not in use. See Appendix C, Optimizing Measurements, for additional information.
- **Related Commands:** BLANk, ERASe.

SUMMary

The SUMMary command subsystem enables you to examine the status of the Oscilloscope calibration and self test results by monitoring (reading the bit value) the various register groups. Figure 4-1 shows the six SUMMary Registers in the Oscilloscope.

Standard Event Status Register (*ESE). Operates under IEEE 488.2 control. Refer to the appropriate mainframe or command module users manual for more information on this register.

Status Byte Register (*STB?). Operates under IEEE 488.2 control. Refer to the appropriate mainframe or command module users manual for more information on this register.

Trigger Register (TER?). Operates under Oscilloscope control. Refer to the TER? query in this chapter for more information on this register.

Limit Test Register (LTER?). Operates under Oscilloscope control. Refer to the LTER? query in this chapter for more information on this register.

Questionable Data/Signal Register. Operates under Oscilloscope control. The Questionable Data/Signal Register is discussed in this section. The illustration shown in figure 4-2 illustrates the Questionable Data/Signal Register.

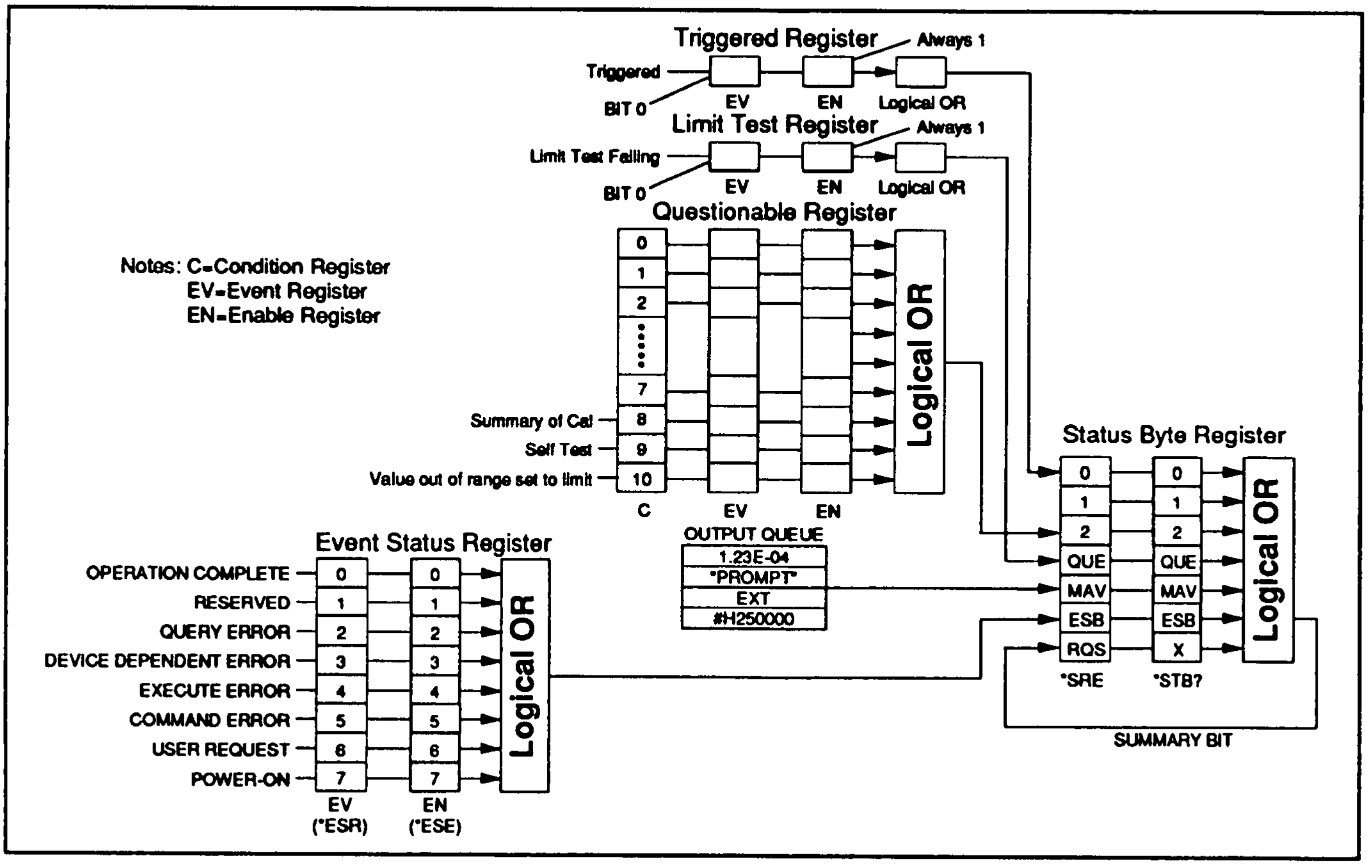


Figure 4-1. Oscilloscope Summary Registers

Subsystem Syntax

```
SUMMary
    :PRESet
    :QUEStionable
        :CONDition?
        :ENABle
        [:EVENt]?
        :CALibration
            :CONDition?
            :ENABle
            [:EVENt]?
            :CHANnel<number>
                :CONDition?
                :ENABle
                [:EVENt]?
                :AD
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :DELay
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :GAIN
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :HYSTeresis
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :LTRigger
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :OFFSet
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :TNULl
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :TRIGger
                    :CONDition?
                    :ENABle
                    [:EVENt]?
            :DCALibration
                :CONDition?
                :ENABle
                [:EVENt]?
            :PROBe
                :CONDition?
                :ENABle
                [:EVENt]?
```

Subsystem Syntax

```
SUMMary — Continued
  :QUEStionable — Continued
    :TEST
      :CONDition?
      :ENABle
      [:EVENt]?
      :ACQuisition
        :CONDition?
        :ENABle
        [:EVENt]?
        :AD
          :CONDition?
          :ENABle
          [:EVENt]?
        :ATRigger
          :CONDition?
          :ENABle
          [:EVENt]?
        :DA
          :CONDition?
          :ENABle
          [:EVENt]?
        :LTRigger
          :CONDition?
          :ENABle
          [:EVENt]?
        :TIMebase
          :CONDition?
          :ENABle
          [:EVENt]?
          :INTerpolator
            :CONDition?
            :ENABle
            [:EVENt]?
      :RAM
        :CONDition?
        :ENABle
        [:EVENt]?
        :ACQuisition
          :CONDition?
          :ENABle
          [:EVENt]?
        :DISPlay
          :CONDition?
          :ENABle
          [:EVENt]?
        :NVOLatile
          :CONDition?
          :ENABle
          [:EVENt]?
        :SYSTem
          :CONDition?
          :ENABle
          [:EVENt]?
```

Subsystem Syntax

```
SUMMary — Continued
    :QUEStionable — Continued
        :TEST — Continued
            :ROM
                :CONDition?
                :ENABle
                [:EVENt]?
                :NPRotect
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :SYSTem
                    :CONDition?
                    :ENABle
                    [:EVENt]?
```

:PRESet

SUMMary:PRESet sets the contents of the Oscilloscope enable registers to a known state. When executed, the PRESet command affects all 51 QUEStionable ENABle registers, and sets all bits true (1).

Example **Preset the oscilloscope enable register**

```
SUMM:PRES                    All Enable register bits to true
```

Comments

- **Other Registers:** PRESet does not affect the Status Byte or Event Status registers. The Triggered and Limit Test ENABle registers are always set at 1.
- **Questionable Enable Register:** PRESet sets the questionable enable register to 0.
- **Event Registers:** PRESet does not affect any of the QUEStionable EVENt registers. Use the *CLS command is used to clear all event registers.

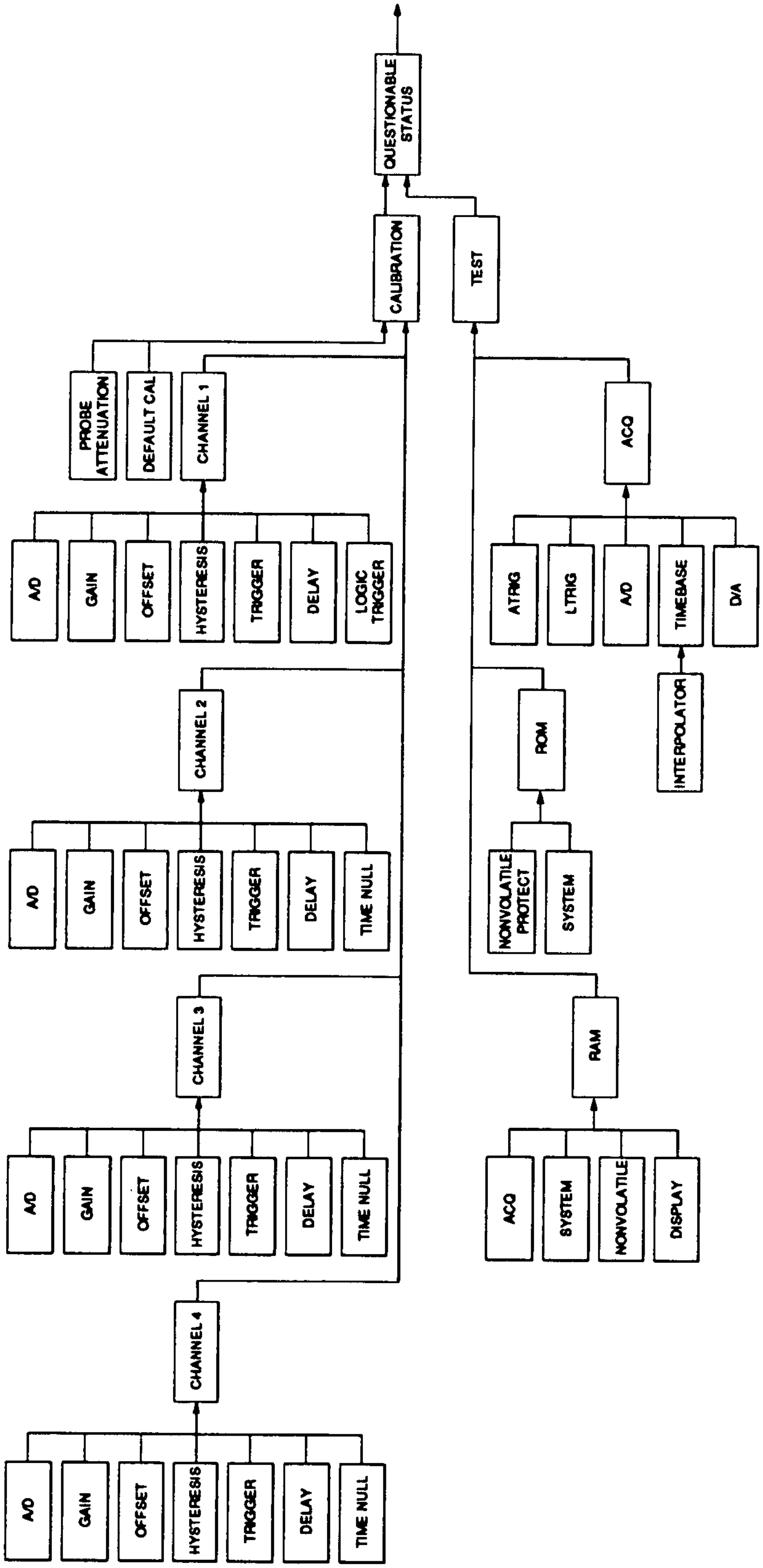
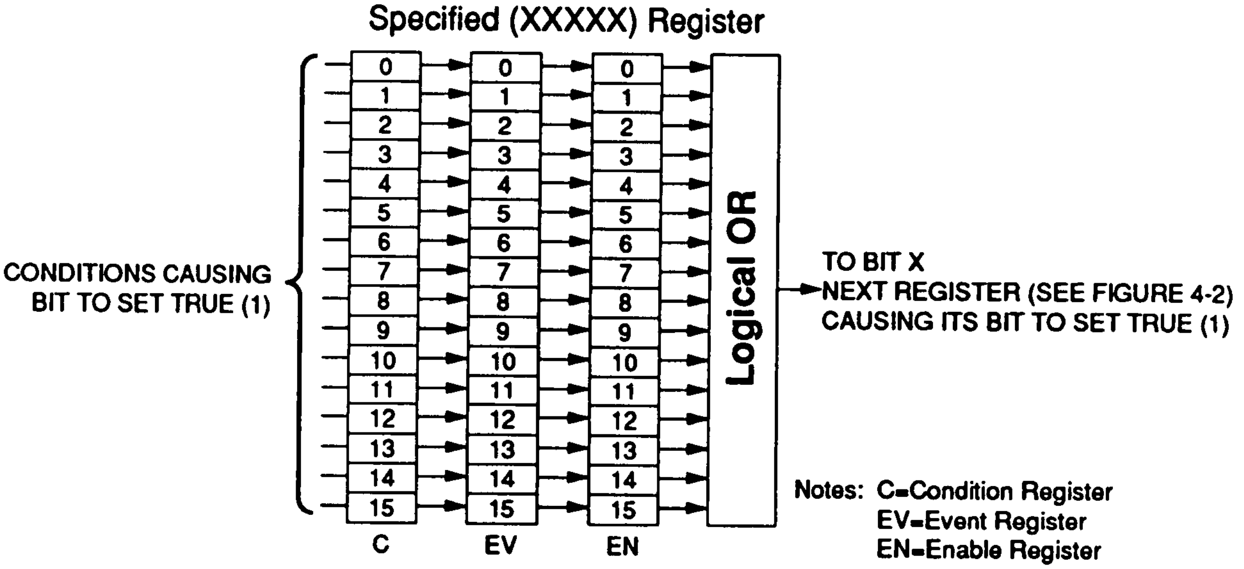


Figure 4-2. Oscilloscope Summary Questionable Data/Signal Register Subsystem

:QUEStionable

SUMMary:QUEStlonable subsystem contains 52 separate registers that, through summing registers, eventually report to the QUEStionable Data/Signal register. See figure 4-2 and the Subsystem Syntax at the beginning of this section for a list of all the registers that set the QUEStionable Data/Signal Register.

A diagram is provided for each register in the QUEStionable Data/Signal Register system as shown in figure 4-3. The following description for using the CONDition?, [:EVENt]?, and ENABle commands applies to all registers within the Oscilloscope.


Specified (XXXXX) Register
CONDITIONS CAUSING BIT TO SET TRUE (1)
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
TO BIT X NEXT REGISTER (SEE FIGURE 4-2) CAUSING ITS BIT TO SET TRUE (1)
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

These registers are set and queried using decimal weighted bit values. The decimal equivalent for bits 0 to 15 is shown below. As an example, sending a decimal value of 4608 will set bits 9 and 12 true (1).

Bit Number to Decimal Value

Bit Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Decimal Value	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	reserved

Figure 4-3. Register Diagram

Each individual register (XXXXX) in the Oscilloscope is made up of three separate registers:

:CONDition register

:EVENt register

:ENABle register

:CONDition? **SUMMary:QUEStionable:XXXXX:CONDition?** queries the current contents of the specified (XXXXX):CONDition Register. The contents of all the CONDition Registers are always set to "0".

Example **Read the condition register**

```
SUMM:QUES:XXXXX:COND?
```

Queries the specified (XXXXX) Condition Register.

:ENABle **SUMMary:QUEStionable:XXXXX:ENABle** ***<number>*** sets the enable mask, which allows true conditions (transitions) in the specified (XXXXX):EVENt Register to be reported.

Example **Set enable register bits 9 through 12 to true**

```
SUMM:QUES:XXXXX:ENAB 7680
```

Sets bits 9 to 12 true

:ENABle? **SUMMary:QUEStionable:XXXXX:ENABle?** returns the bit value of the specified (XXXXX):ENABle Register. Returns a decimal weighted value from 0 to 65535 indicating which bits are set true. Reading the ENABLe register does not clear its contents.

Example **Query the enable register**

```
SUMM:QUES:XXXXX:ENAB?
```

Queries the specified (XXXXX) Enable register, without clearing the contents

[:EVENt]? **SUMMary:QUEStionable:XXXXX[:EVENt]?** Queries the status of the specified (XXXXX):EVENt Register. The EVENt Register latches only low to high events from the specified (XXXXX):CONDition Register. Returns a decimal weighted value from 0 to 65535 indicating which bits are set true. Reading the specified EVENt Register by a query will clear its contents.

Example **Read the event register**

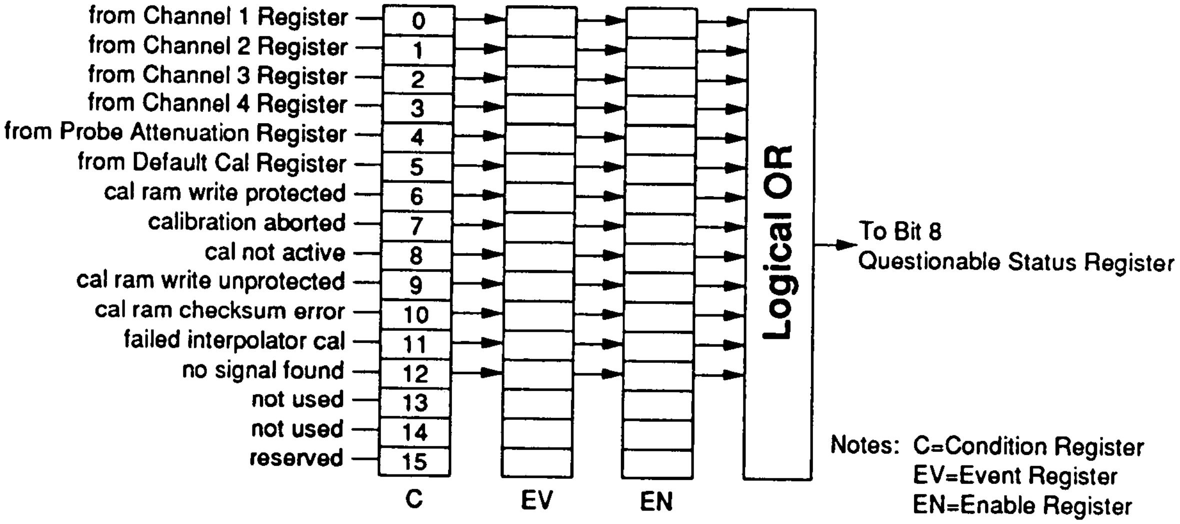
```
SUMM:QUES:XXXXX?
```

Queries the specified (XXXXX) Event Register and clears the contents

:QUEStionable:CALibration

SUMMary:QUEStionable:CALibration register reports a summary of calibration results and status for all channels to the Questionable Data/Signal Register. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Summary of Calibration Register



Example **Query calibration event register**

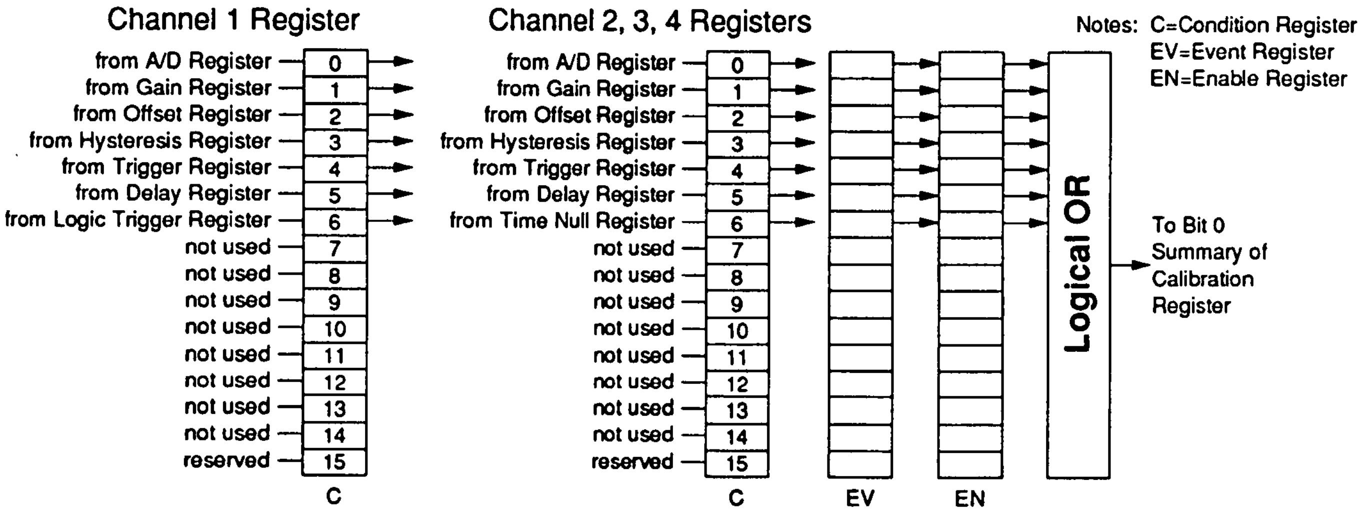
`SUMM:QUES:CAL?` *Query instrument to return register contents*

:QUEStionable:CALibration:CHANnel

SUMMary:QUEStionable:CALibration:CHANnel<*number*> register reports the status of calibration data for the channel specified. *number* (1 to 4) specified the desired channel. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none



Example **Query channel 1 event register**

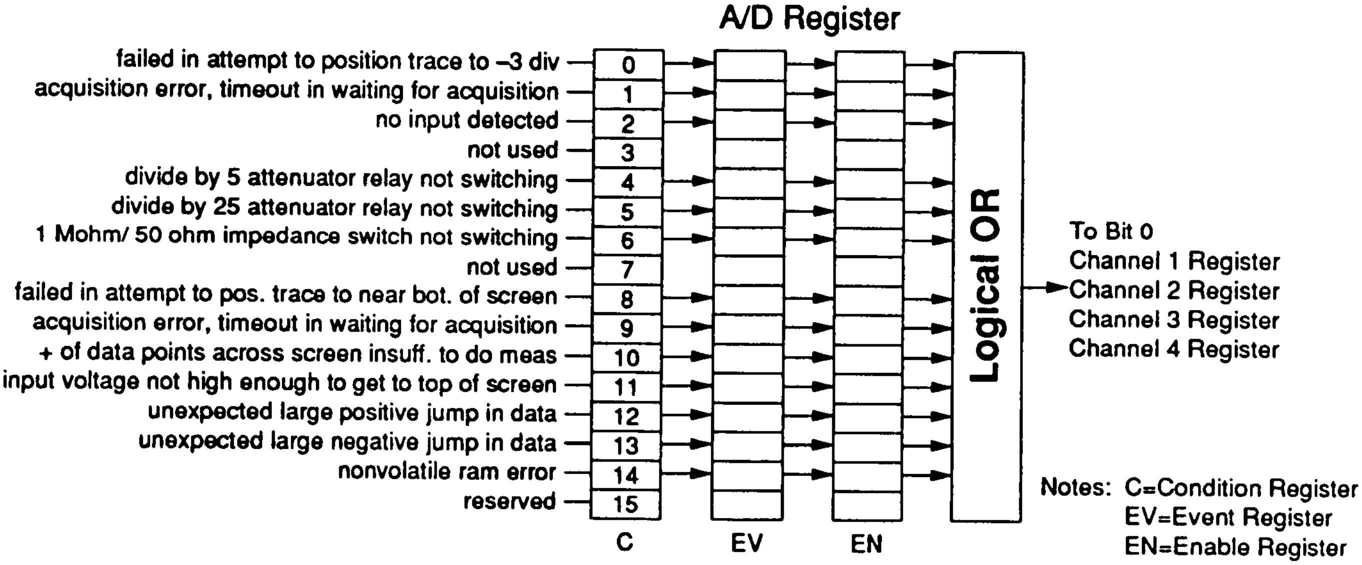
`SUMM:QUES:CAL:CHAN1?` *Query instrument to return register contents*

:QUEStionable:CALibration:CHANnel:AD

SUMMary:QUEStionable:CALibration:CHANnel<*number*>:AD register reports the status of the A/D calibration data for the channel specified. *number* (1 to 4) specifies the desired channel. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none


A/D Register
failed in attempt to position trace to -3 div
acquisition error, timeout in waiting for acquisition
no input detected
not used
divide by 5 attenuator relay not switching
divide by 25 attenuator relay not switching
1 Mohm/ 50 ohm impedance switch not switching
not used
failed in attempt to pos. trace to near bot. of screen
acquisition error, timeout in waiting for acquisition
+ of data points across screen insuff. to do meas
input voltage not high enough to get to top of screen
unexpected large positive jump in data
unexpected large negative jump in data
nonvolatile ram error
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 0
Channel 1 Register
Channel 2 Register
Channel 3 Register
Channel 4 Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query channel 2 A/D event register**

```
SUMM:QUES:CAL:CHAN2:AD?
```

Query instrument to return register contents

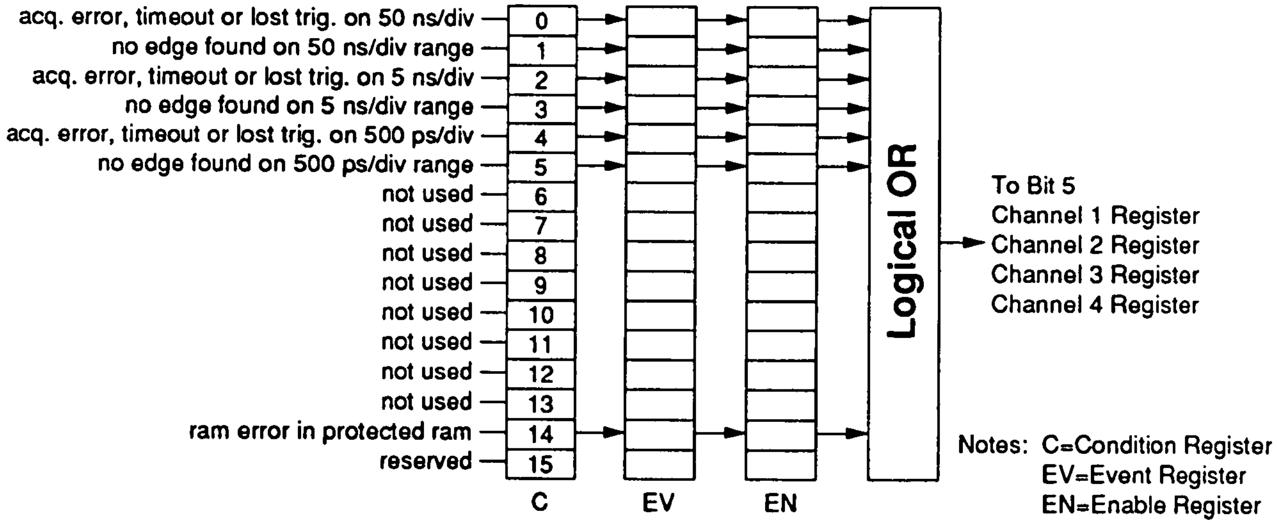
:QUEStionable:CALibration:CHANnel:DELay

SUMMary:QUEStionable:CALibration:CHANnel<*number*>:DELay register reports the status of delay calibration data for the channel specified. *number* (1 to 4) specifies the desired channel. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Delay Register



acq. error, timeout or lost trig. on 50 ns/div
no edge found on 50 ns/div range
acq. error, timeout or lost trig. on 5 ns/div
no edge found on 5 ns/div range
acq. error, timeout or lost trig. on 500 ps/div
no edge found on 500 ps/div range
not used
not used
not used
not used
not used
not used
not used
not used
ram error in protected ram
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 5
Channel 1 Register
Channel 2 Register
Channel 3 Register
Channel 4 Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example

Query channel 3 delay event register

```
SUMM:QUES:CAL:CHAN3:DEL?
```

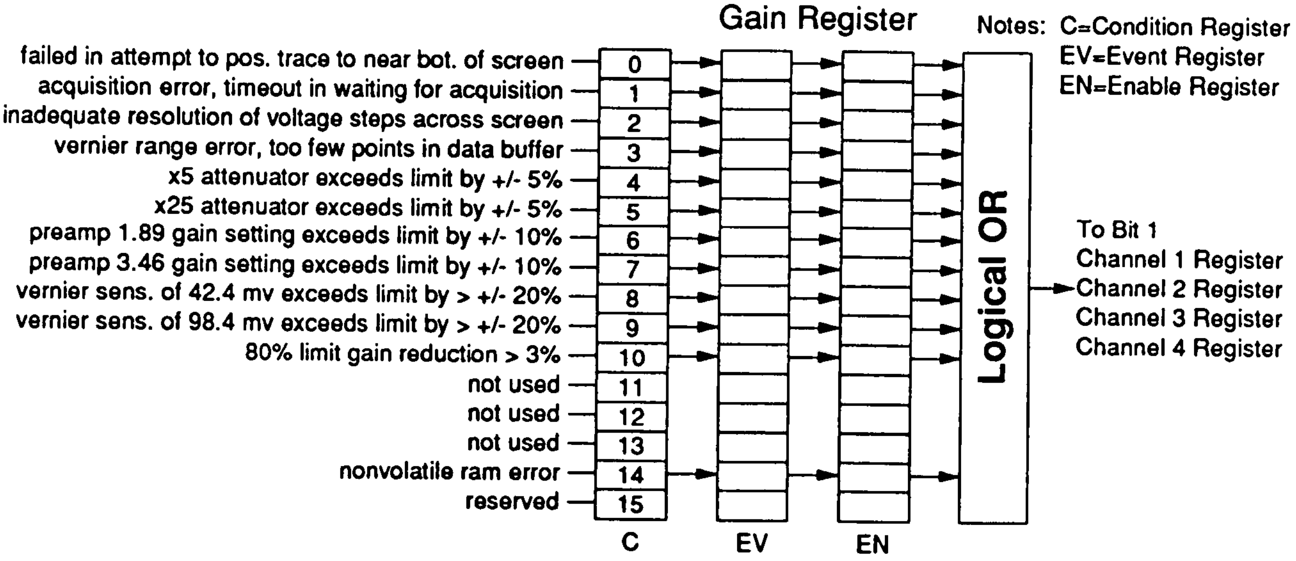
Query instrument to return register contents

:QUEStionable:CALibration:CHANnel:GAIN

SUMMary:QUEStionable:CALibration:CHANnel<*number*>:GAIN register reports the status of gain calibration data for the channel specified. *number* (1 to 4) specifies the desired channel. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none


Gain Register
failed in attempt to pos. trace to near bot. of screen
acquisition error, timeout in waiting for acquisition
inadequate resolution of voltage steps across screen
vernier range error, too few points in data buffer
x5 attenuator exceeds limit by +/- 5%
x25 attenuator exceeds limit by +/- 5%
preamp 1.89 gain setting exceeds limit by +/- 10%
preamp 3.46 gain setting exceeds limit by +/- 10%
vernier sens. of 42.4 mv exceeds limit by > +/- 20%
vernier sens. of 98.4 mv exceeds limit by > +/- 20%
80% limit gain reduction > 3%
not used
not used
not used
nonvolatile ram error
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
Notes: C=Condition Register
EV=Event Register
EN=Enable Register
To Bit 1
Channel 1 Register
Channel 2 Register
Channel 3 Register
Channel 4 Register

Example **Query channel 4 gain event register**

```
SUMM:QUES:CAL:CHAN4:GAIN?
```
Query instrument to return register contents

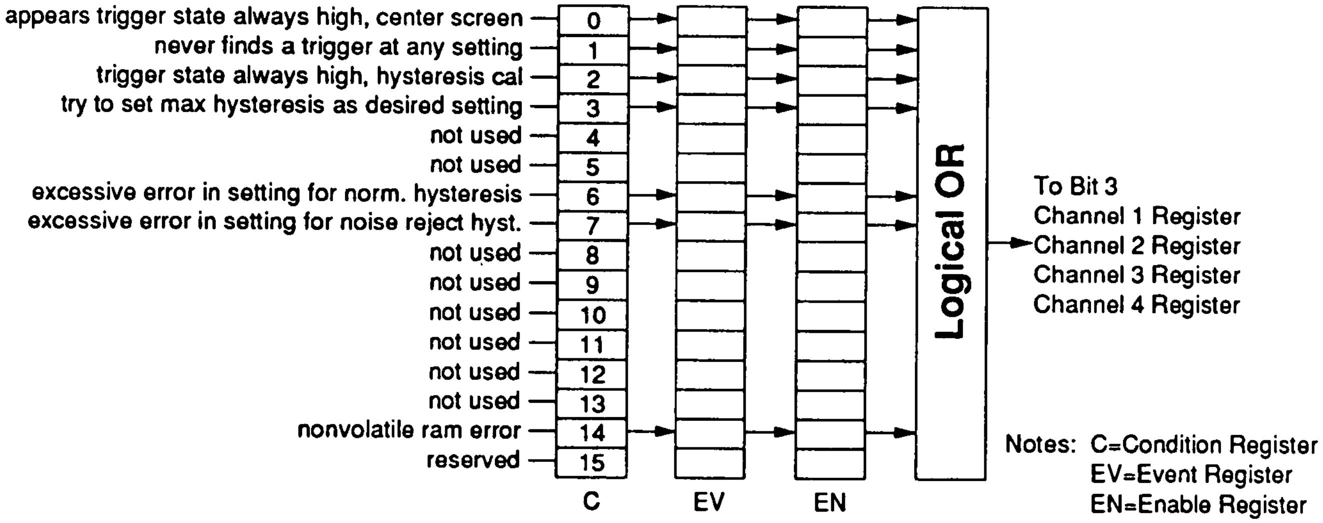
:QUEStionable:CALibration:CHANnel:HYSTeresis

SUMMary:QUEStionable:CALibration:CHANnel<*number*>:HYSTeresis register reports the status of hysteresis calibration data for the channel specified. *number* (1 to 4) specifies the desired channel. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Hysteresis Register


appears trigger state always high, center screen
never finds a trigger at any setting
trigger state always high, hysteresis cal
try to set max hysteresis as desired setting
not used
not used
excessive error in setting for norm. hysteresis
excessive error in setting for noise reject hyst.
not used
not used
not used
not used
not used
not used
nonvolatile ram error
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 3
Channel 1 Register
Channel 2 Register
Channel 3 Register
Channel 4 Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example

Query channel 1 hysteresis event register

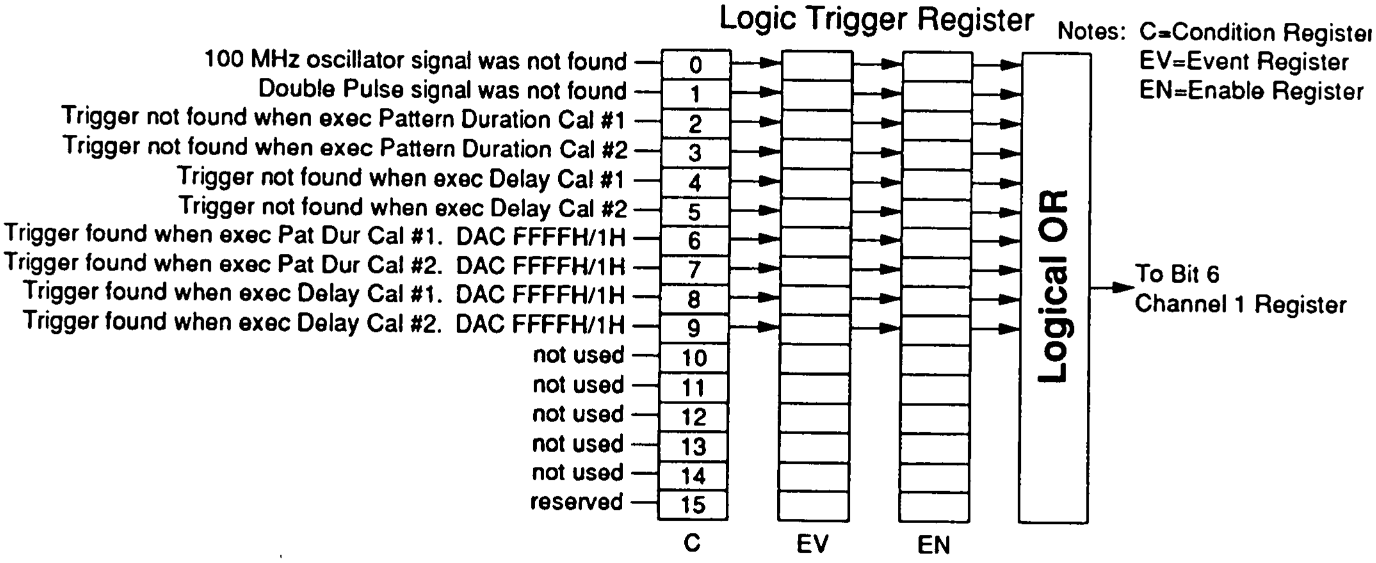
`SUMM:QUES:CAL:CHAN1:HYST?` *Query instrument to return register contents*

:QUEStionable:CALibration:CHANnel:LTRigger

SUMMary:QUEStionable:CALibration:CHANnel*<number>*:LTRigger register reports the status of logic trigger calibration data for channel 1. Only channel 1 contains the LTRigger register. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1	none


Logic Trigger Register
Notes: C=Condition Registei
EV=Event Register
EN=Enable Register
100 MHz oscillator signal was not found
Double Pulse signal was not found
Trigger not found when exec Pattern Duration Cal #1
Trigger not found when exec Pattern Duration Cal #2
Trigger not found when exec Delay Cal #1
Trigger not found when exec Delay Cal #2
Trigger found when exec Pat Dur Cal #1. DAC FFFFH/1H
Trigger found when exec Pat Dur Cal #2. DAC FFFFH/1H
Trigger found when exec Delay Cal #1. DAC FFFFH/1H
Trigger found when exec Delay Cal #2. DAC FFFFH/1H
not used
not used
not used
not used
not used
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Logical OR
To Bit 6
Channel 1 Register
C
EV
EN

Example **Query channel 1 logic trigger event register**

`SUMM:QUES:CAL:CHAN1:LTR?` *Query instrument to return register contents*

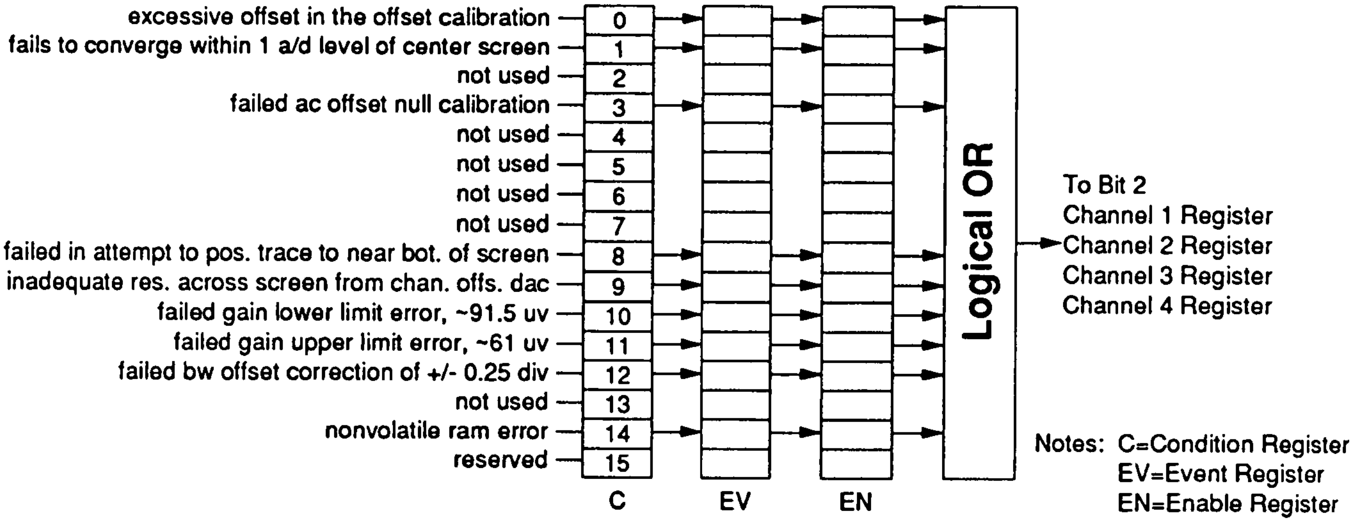
:QUEStionable:CALibration:CHANnel:OFFSet

SUMMary:QUEStionable:CALibration:CHANnel<*number*>:OFFSet register reports the status of offset calibration data for the channel specified. *number* (1 to 4) specifies the desired channel. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Offset Register


excessive offset in the offset calibration — 0
fails to converge within 1 a/d level of center screen — 1
not used — 2
failed ac offset null calibration — 3
not used — 4
not used — 5
not used — 6
not used — 7
failed in attempt to pos. trace to near bot. of screen — 8
inadequate res. across screen from chan. offs. dac — 9
failed gain lower limit error, ~91.5 uv — 10
failed gain upper limit error, ~61 uv — 11
failed bw offset correction of +/- 0.25 div — 12
not used — 13
nonvolatile ram error — 14
reserved — 15
C
EV
EN
Logical OR
To Bit 2
Channel 1 Register
Channel 2 Register
Channel 3 Register
Channel 4 Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query channel 2 offset event register**

`SUMM:QUES:CAL:CHAN2:OFFS?`	*Query instrument to return register contents*

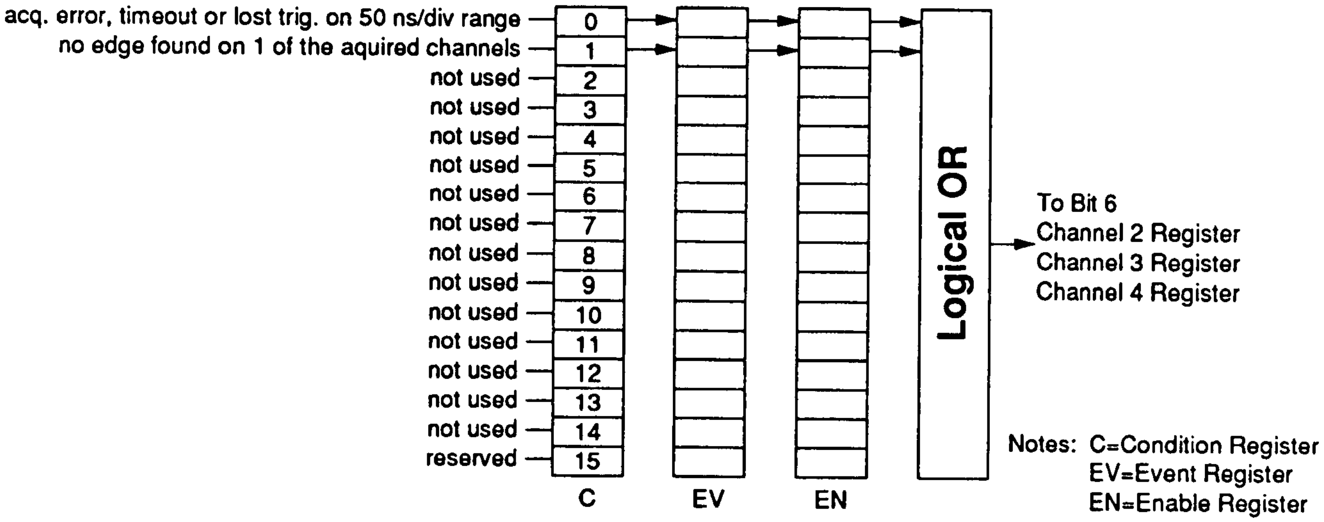
:QUEStionable:CALibration:CHANnel:TNULI

SUMMary:QUEStionable:CALibration:CHANnel<*number*>TNULl register reports the status of time null calibration data for the channel specified. *number* (2 to 4) specifies the desired channel. Channel 1 does not contain a time null register. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	2 to 4	none

Time Null Register



Example **Query channel 3 time null event register**

SUMM:QUES:CAL:CHAN3:TNUL? *Query instrument to return register contents*

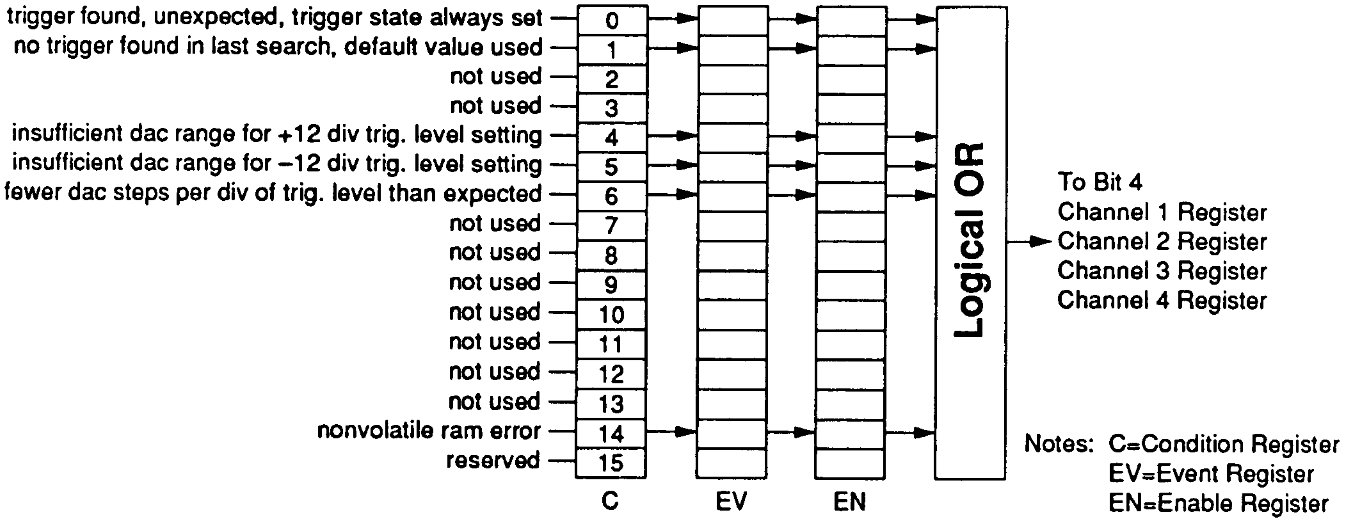
:QUEStionable:CALibration:CHANnel:TRIGger

SUMMary:QUEStionable:CALibration:CHANnel<*number*>:TRIGger register reports the status of trigger calibration data for the channel specified. *number* (1 to 4) specifies the desired channel. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Trigger Register


trigger found, unexpected, trigger state always set
no trigger found in last search, default value used
not used
not used
insufficient dac range for +12 div trig. level setting
insufficient dac range for –12 div trig. level setting
fewer dac steps per div of trig. level than expected
not used
not used
not used
not used
not used
not used
not used
nonvolatile ram error
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 4
Channel 1 Register
Channel 2 Register
Channel 3 Register
Channel 4 Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

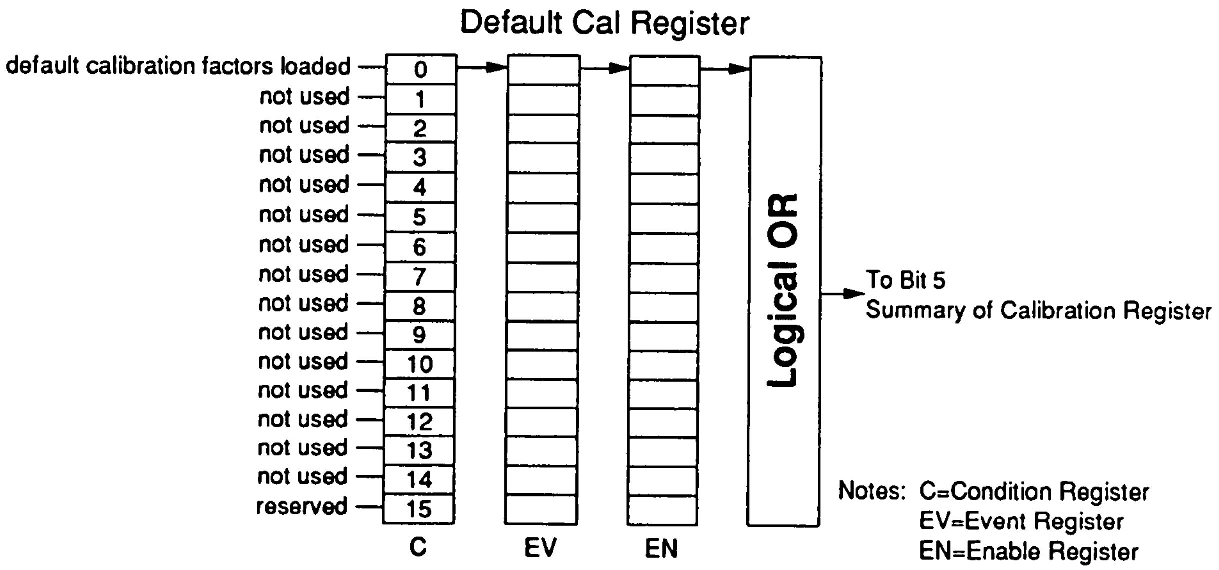
Example **Query channel 4 trigger event register**

```
SUMM:QUES:CAL:CHAN4:TRIG?
```

Query instrument to return register contents

:QUEStionable:CALibration:DCALibration

SUMMary:QUEStionable:CALibration:DCALibration register reports default calibration factor status load. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

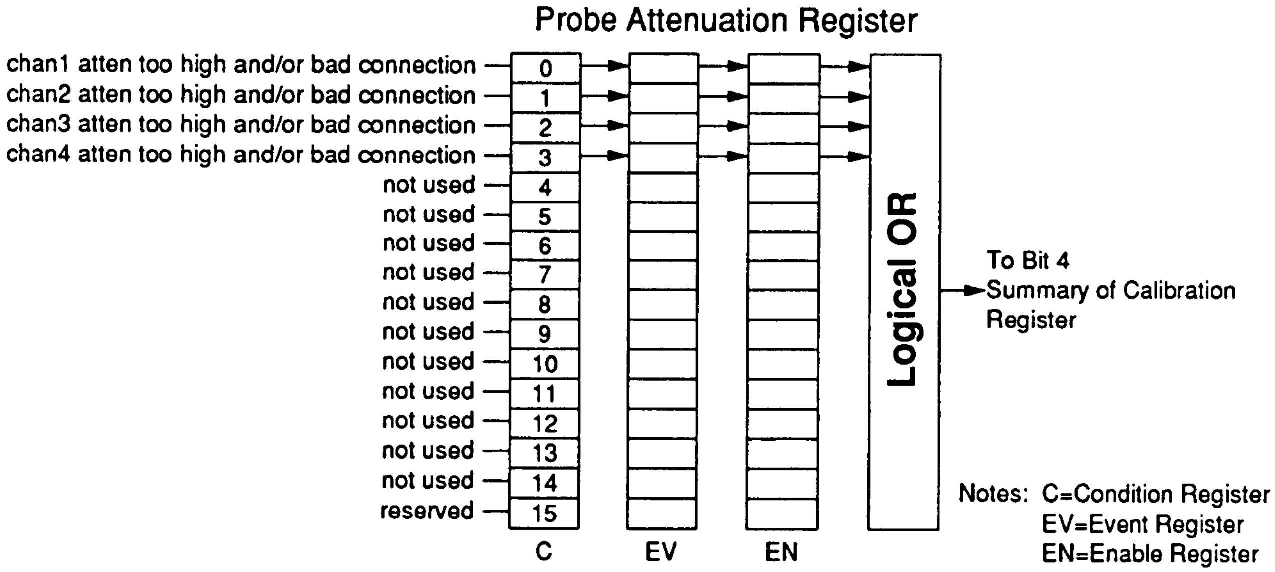


Example **Query default calibration event register**

`SUMM:QUES:CAL:DCAL?` *Query instrument to return register contents*

:QUEStionable:CALIbration:PROBe

SUMMary:QUEStionable:CALibration:PROBe register reports probe calibration attenuation results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.


Probe Attenuation Register
chan1 atten too high and/or bad connection
chan2 atten too high and/or bad connection
chan3 atten too high and/or bad connection
chan4 atten too high and/or bad connection
not used
not used
not used
not used
not used
not used
not used
not used
not used
not used
not used
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 4
Summary of Calibration
Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

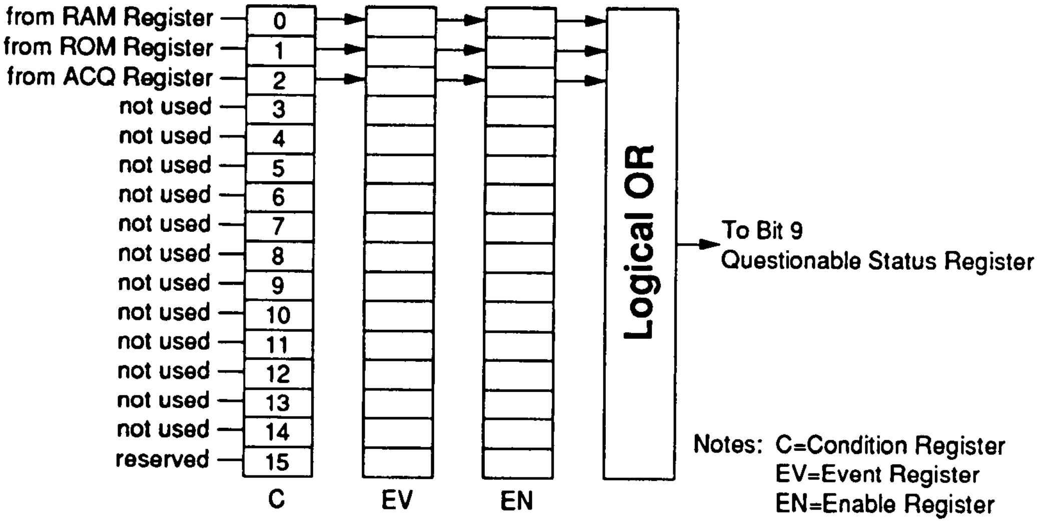
Example **Query probe calibration attenuation event register**

`SUMM:QUES:CAL:PROB?` *Query instrument to return register contents*

:QUEStionable:TEST

SUMMary:QUEStionable:TEST register reports diagnostic test results or self test status. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Self Test Status Register



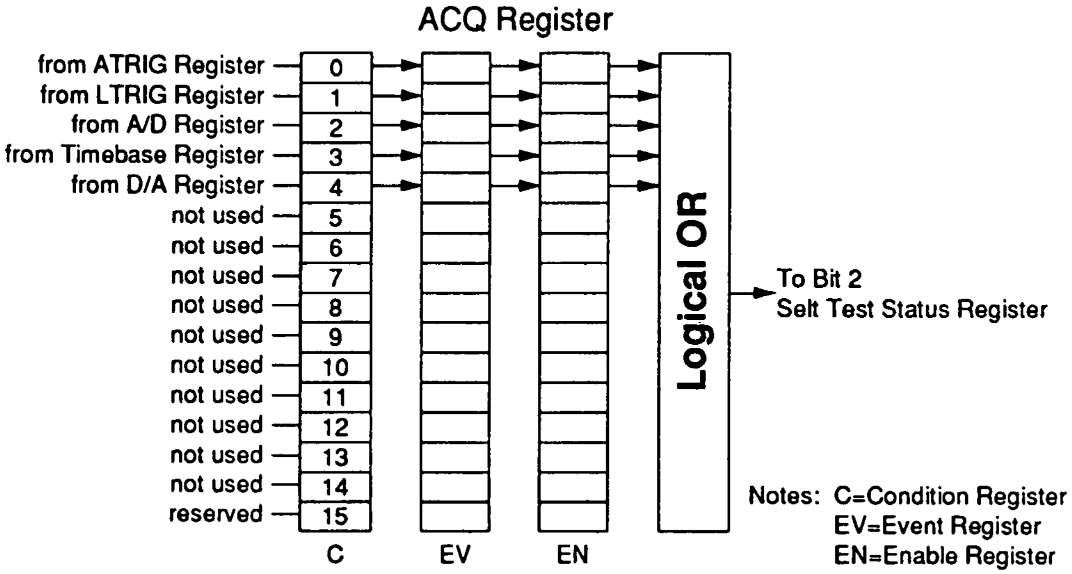
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query test event register**

`SUMM:QUES:TEST?` *Query instrument to return register contents*

:QUEStionable:TEST:ACQuisition

SUMMary:QUEStionable:TEST:ACQuisition register reports acquisition diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.



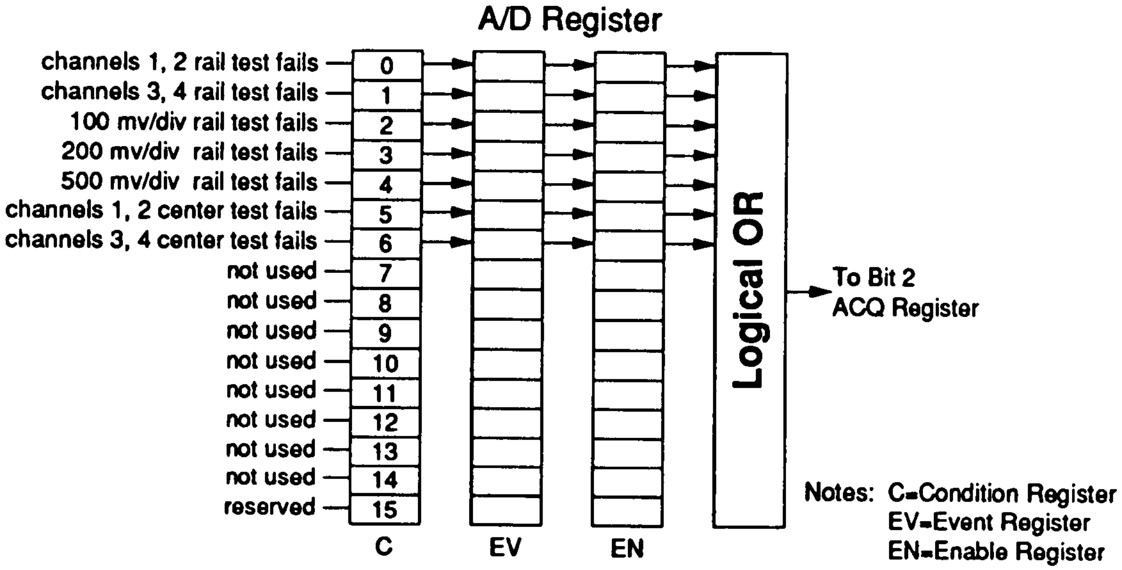
Example **Query acquisition event register**

```
SUMM:QUES:TEST:ACQ?
```

Query instrument to return register contents

:QUEStionable:TEST:ACQuisition:AD

SUMMary:QUEStionable:TEST:ACQuisition:AD register reports acquisition A/D diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.


A/D Register
channels 1, 2 rail test fails
channels 3, 4 rail test fails
100 mv/div rail test fails
200 mv/div rail test fails
500 mv/div rail test fails
channels 1, 2 center test fails
channels 3, 4 center test fails
not used
not used
not used
not used
not used
not used
not used
not used
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 2
ACQ Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

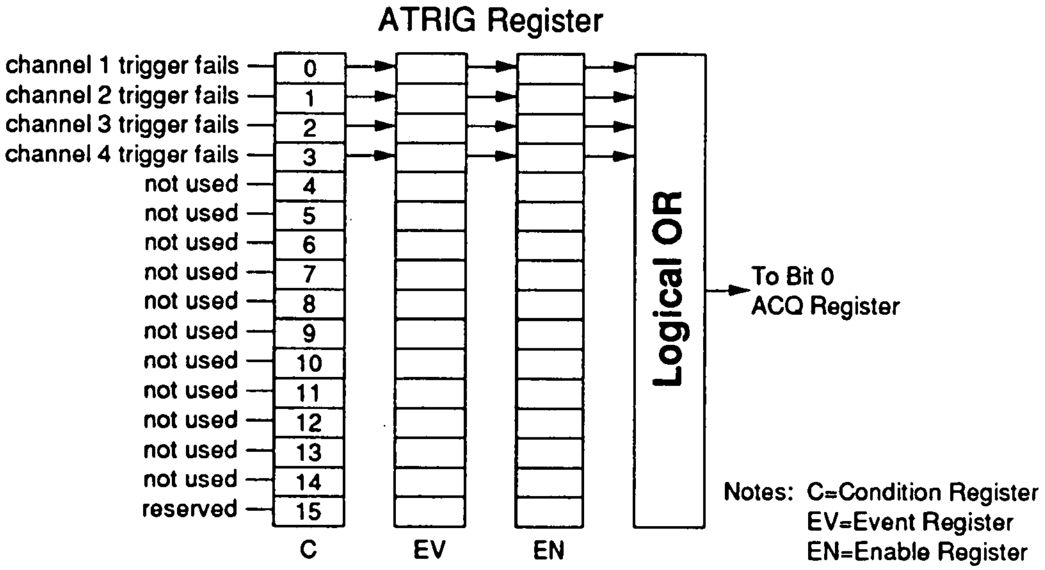
Example **Query acquisition A/D event register**

```
SUMM:QUES:TEST:ACQ:AD?
```

Query instrument to return register contents

:QUEStionable:TEST:ACQuisition:ATRigger

SUMMary:QUEStionable:TEST:ACQuisition:ATRigger register reports acquisition analog trigger diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

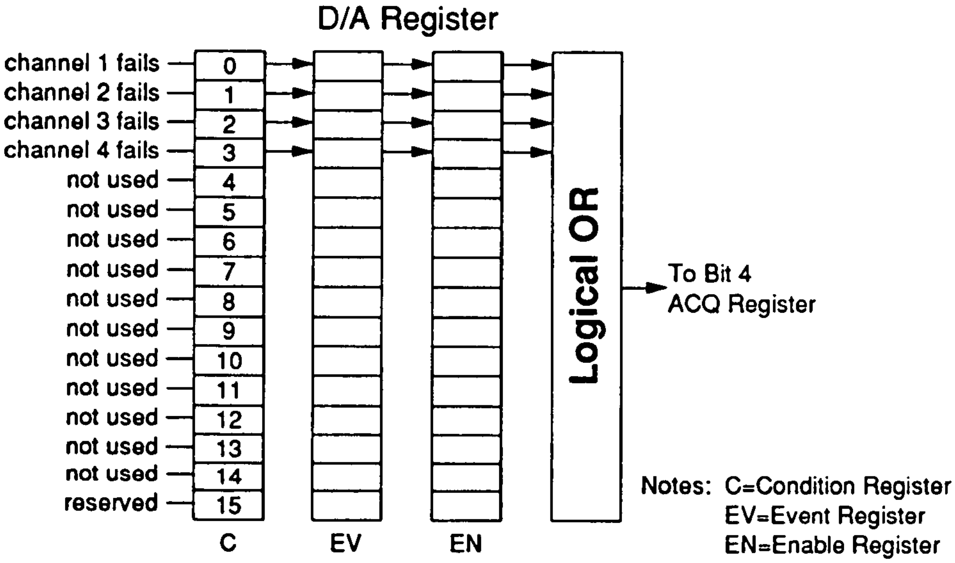


Example **Query acquisition analog trigger event register**

`SUMM:QUES:TEST:ACQ:ATR?` *Query instrument to return register contents*

:QUEStionable:TEST:ACQuisition:DA

SUMMary:QUEStionable:TEST:ACQuisition:DA register reports acquisition D/A diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.



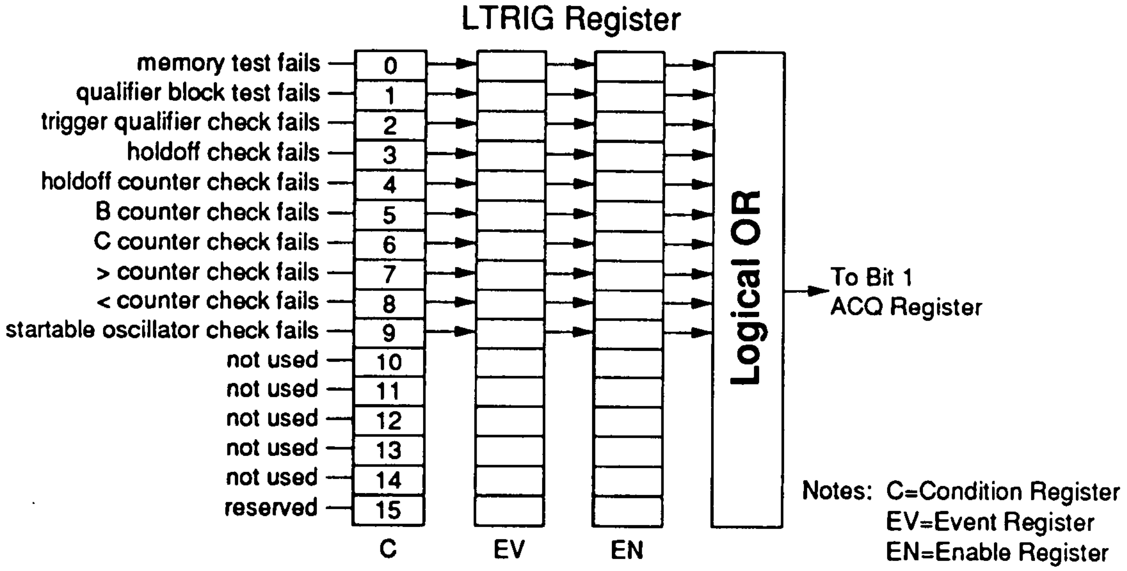
Example **Query acquisition D/A event register**

```
SUMM:QUES:TEST:ACQ:DA?
```

Query instrument to return register contents

:QUEStionable:TEST:ACQuisition:LTRigger

SUMMary:QUEStionable:TEST:ACQuisition:LTRigger register reports acquisition logic trigger diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.



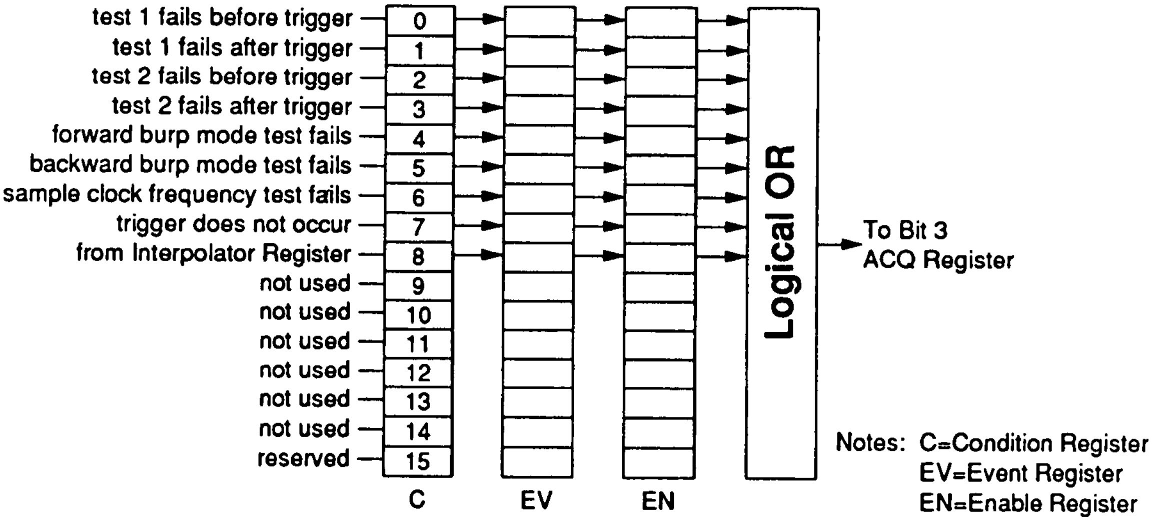
Example **Query acquisition logic trigger event register**

`SUMM:QUES:TEST:ACQ:LTR?` *Query instrument to return register contents*

:QUEStionable:TEST:ACQuisition:TIMebase

SUMMary:QUEStionable:TEST:ACQuisition:TIMebase register reports acquisition time base diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

Timebase Register

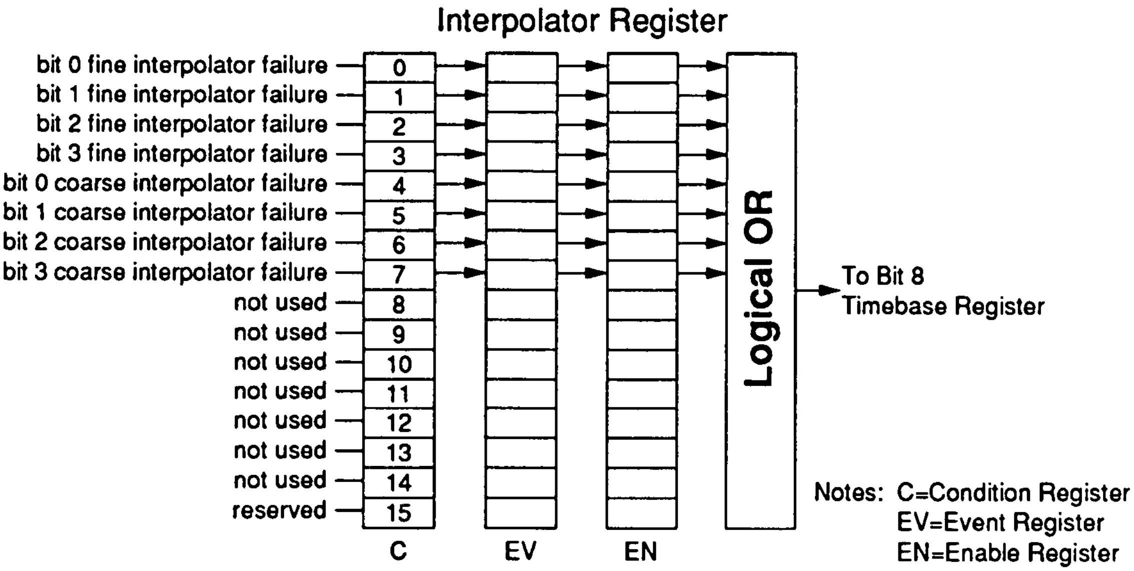

test 1 fails before trigger
test 1 fails after trigger
test 2 fails before trigger
test 2 fails after trigger
forward burp mode test fails
backward burp mode test fails
sample clock frequency test fails
trigger does not occur
from Interpolator Register
not used
not used
not used
not used
not used
not used
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 3 ACQ Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query acquisition time base event register**

`SUMM:QUES:TEST:ACQ:TIM?` *Query instrument to return register contents*

:QUEStionable:TEST:ACQuisition:TIMebase:INTerpolator

SUMMary:QUEStionable:TEST:ACQuisition:TIMebase:INTerpolator register reports acquisition time base interpolator diagnostics. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

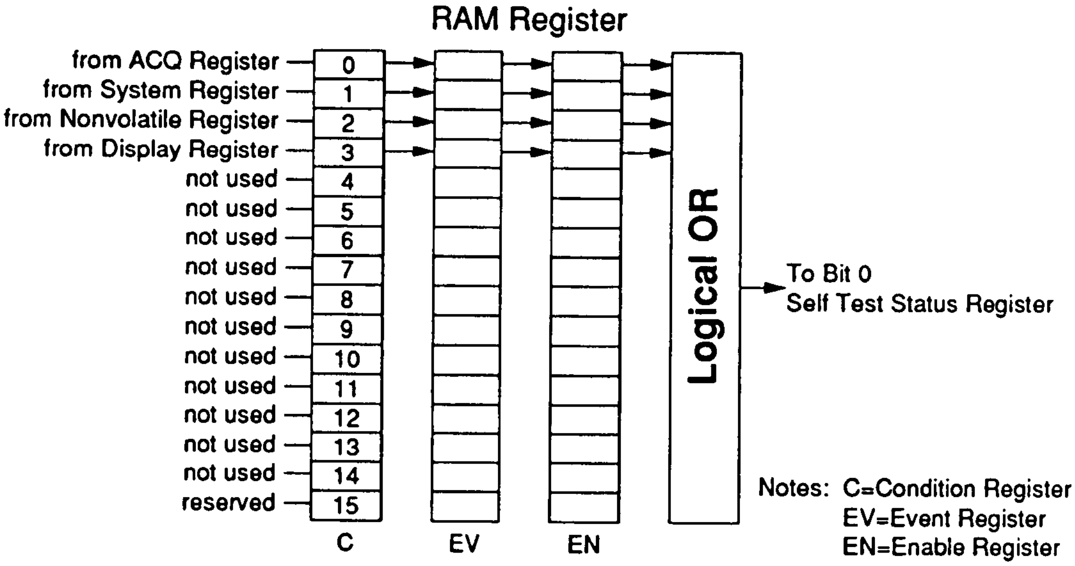

Interpolator Register
bit 0 fine interpolator failure
bit 1 fine interpolator failure
bit 2 fine interpolator failure
bit 3 fine interpolator failure
bit 0 coarse interpolator failure
bit 1 coarse interpolator failure
bit 2 coarse interpolator failure
bit 3 coarse interpolator failure
not used
not used
not used
not used
not used
not used
not used
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 8
Timebase Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query acquisition time base interpolator event register**

`SUMM:QUES:TEST:ACQ:TIM:INT?`	*Query instrument to return register contents*

:QUEStionable:TEST:RAM

SUMMary:QUEStionable:TEST:RAM register reports random access memory diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

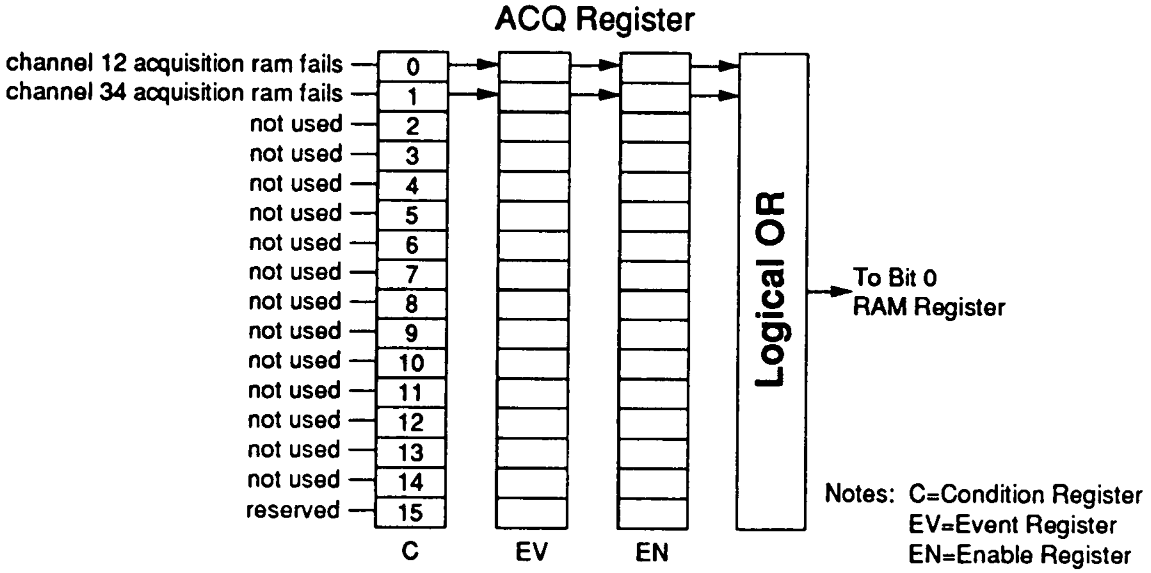


Example **Query RAM event register**

`SUMM:QUES:TEST:RAM?` *Query instrument to return register contents*

:QUEStionable:TEST:RAM:ACQuisition

SUMMary:QUEStionable:TEST:RAM:ACQuisition register reports acquisition random access memory diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

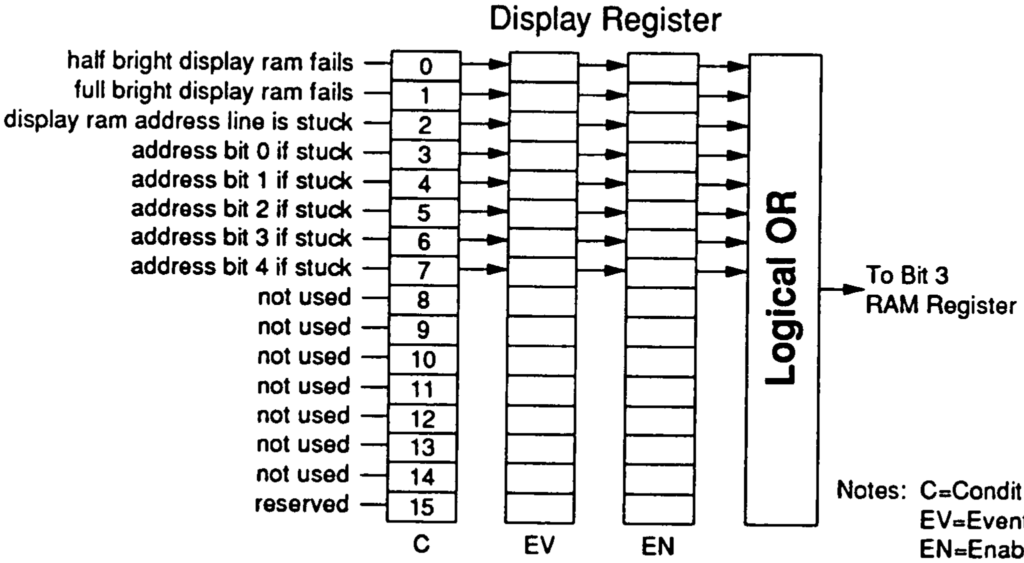


Example **Query acquisition RAM event register**

`SUMM:QUES:TEST:RAM:ACQ?` *Query instrument to return register contents*

:QUEStionable:TEST:RAM:DISPlay

SUMMary:QUEStionable:TEST:RAM:DISPlay register reports display random access memory test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.



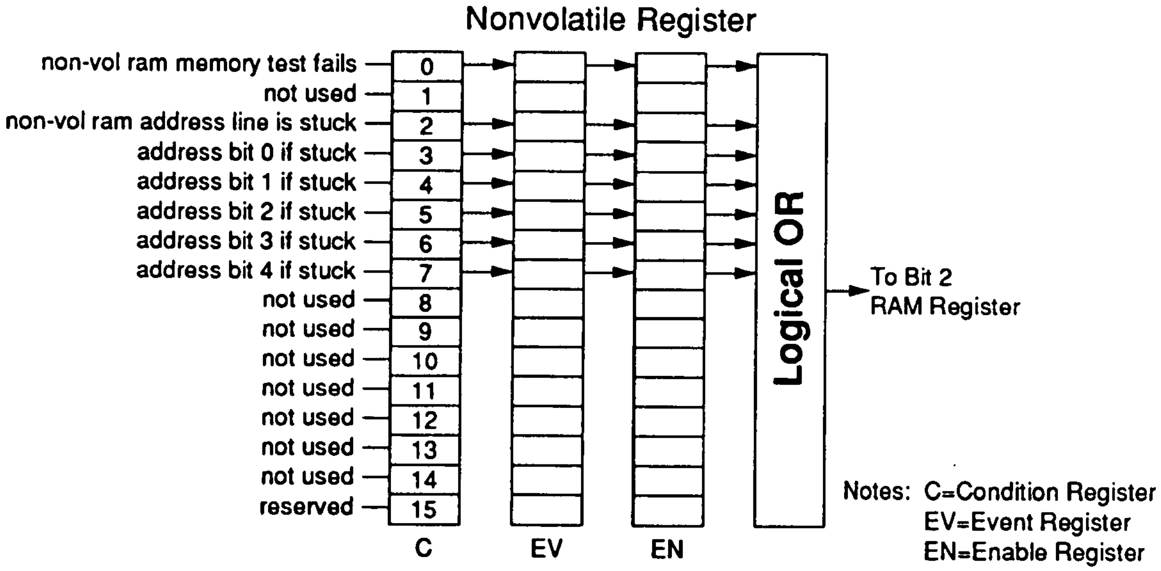
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query display RAM event register**

`SUMM:QUES:TEST:RAM:DISP?` *Query instrument to return register contents*

:QUEStionable:TEST:RAM:NVOLatile

SUMMary:QUEStionable:TEST:RAM:NVOLatile register reports nonvolatile random access memory diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.


Nonvolatile Register
non-vol ram memory test fails — 0
not used — 1
non-vol ram address line is stuck — 2
address bit 0 if stuck — 3
address bit 1 if stuck — 4
address bit 2 if stuck — 5
address bit 3 if stuck — 6
address bit 4 if stuck — 7
not used — 8
not used — 9
not used — 10
not used — 11
not used — 12
not used — 13
not used — 14
reserved — 15
C
EV
EN
Logical OR
To Bit 2
RAM Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

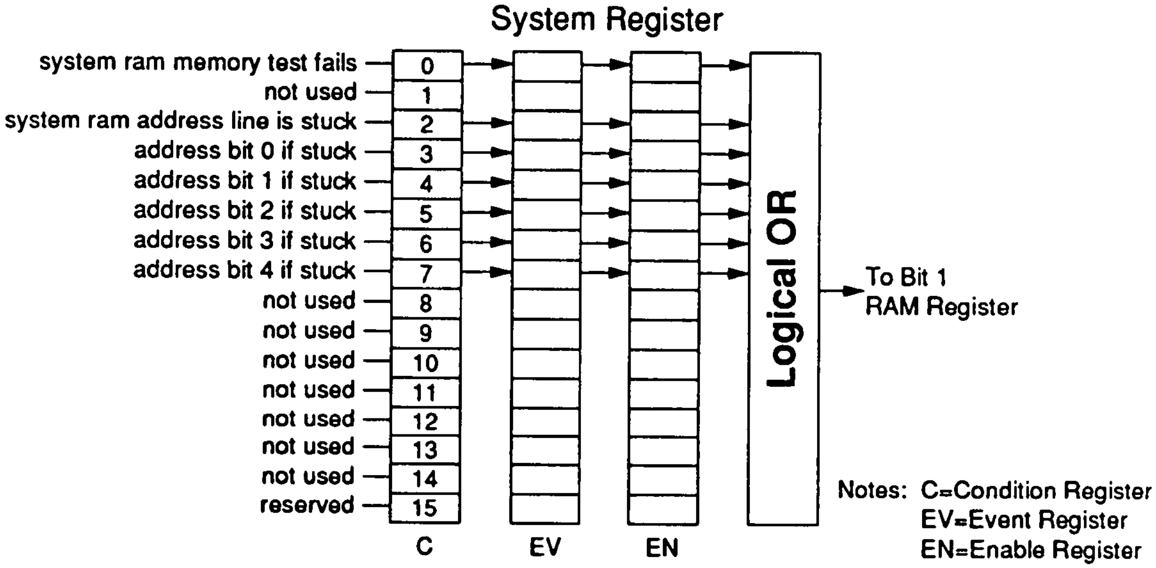
Example **Query nonvolatile RAM event register**

```
SUMM:QUES:TEST:RAM:NVOL?
```

Query instrument to return register contents

:QUEStionable:TEST:RAM:SYSTem

SUMMary:QUEStionable:TEST:RAM:SYSTem register reports system random access memory diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

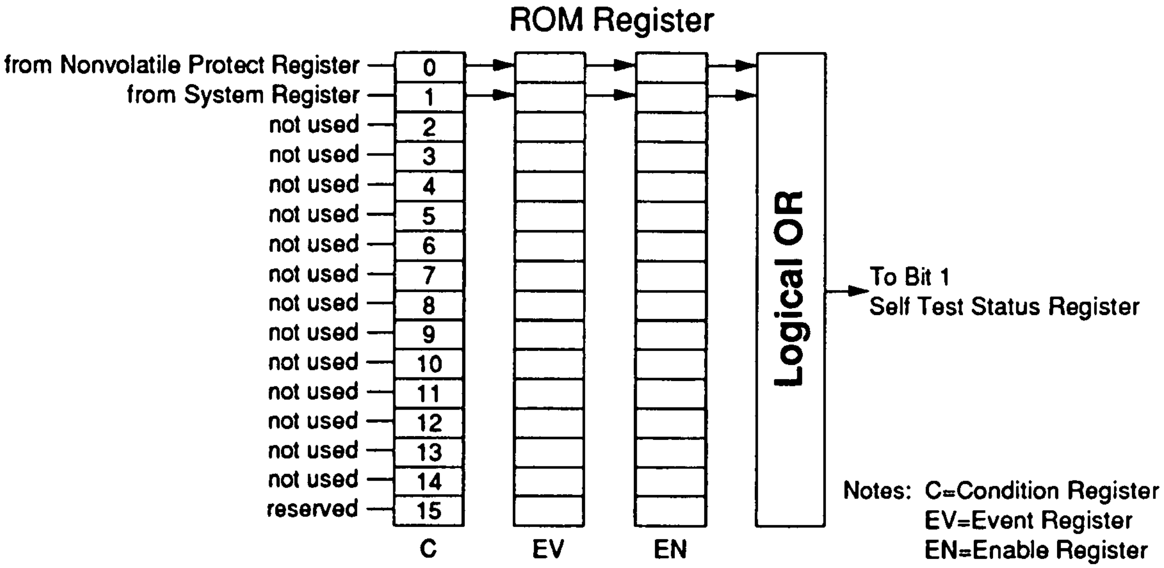


Example **Query system RAM event register**

`SUMM:QUES:TEST:RAM:SYST?` *Query instrument to return register contents*

:QUEStionable:TEST:ROM

SUMMary:QUEStionable:TEST:ROM register reports read only memory diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

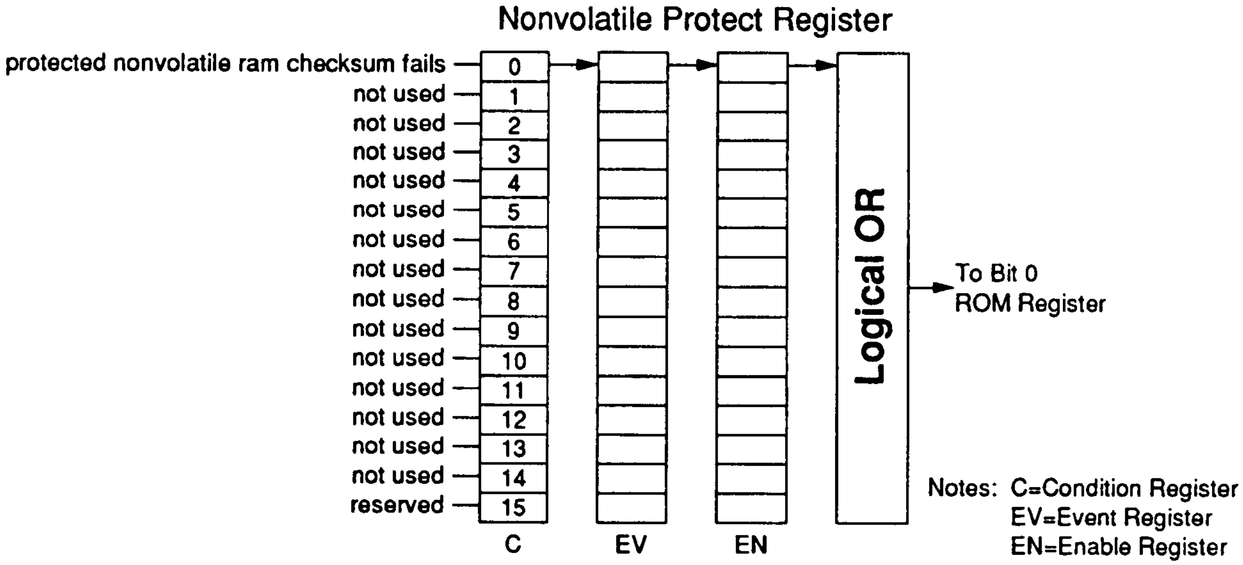


Example **Query ROM event register**

`SUMM:QUES:TEST:ROM?` *Query instrument to return register contents*

:QUEStionable:TEST:ROM:NPRotect

SUMMary:QUEStionable:TEST:ROM:NPRotect register reports non-volatile protected random access memory diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.

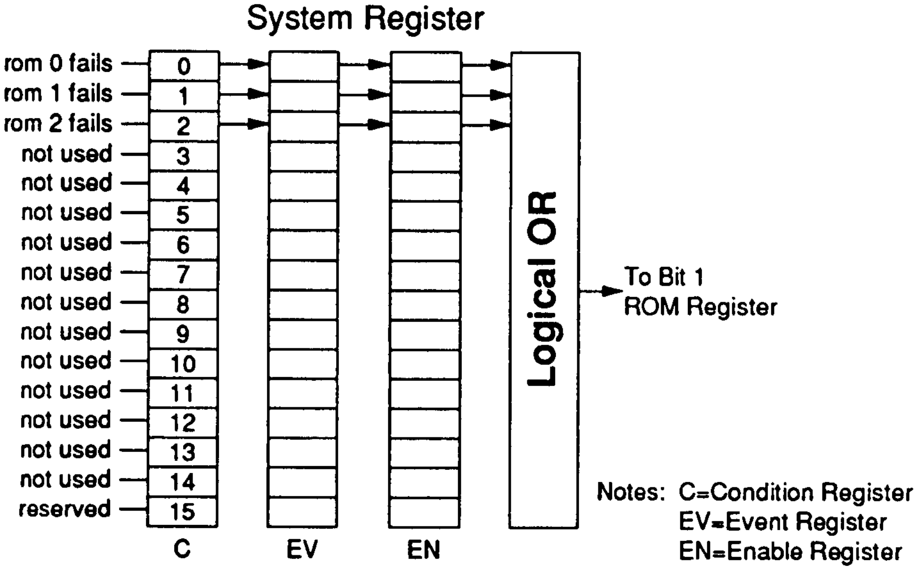


Example **Query non-volatile protected ROM event register**

`SUMM:QUES:TEST:ROM:NPR?`	*Query instrument to return register contents*

:QUEStionable:TEST:ROM:SYSTem

SUMMary:QUEStionable:TEST:ROM:SYSTem register reports system read only memory diagnostic test results. Use the diagram to interpret returned results. See figure 4-2 and 4-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands.



Example **Query system ROM event register**

`SUMM:QUES:TEST:ROM:SYST?` *Query instrument to return register contents*

SYSTem

The SYSTem command subsystem is used to control the way in which query responses are formatted, and define the programming language used.

Subsystem Syntax

```
SYSTem
    :ERRor? [<mode>]
    :HEADer <mode>
    :HEADer?
    :LANGuage <command>
    :LANGuage?
    :LONGform <mode>
    :LONGform?
    :SETup <setup>
    :SETup?
```

:ERRor?

SYSTem:ERRor? [<*message*>] returns the next error number and (if specified) corresponding error message in the error queue. See Appendix B for a listing of error numbers and messages.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	NUMBer|STRing	none

Example

Read the next error number and quoted string in the error queue

```
SYST:ERR? STR
```
Query instrument to return the next error number and message

Comments

- **Entering Mode:** When the NUMBer is specified, only the numeric error code is output. When the STRing is specified the error number is output followed by a comma and a quoted string. If no parameter is specified then only the numeric error code is output (same as NUMBer).
- **Error Numbers/Messages in the Error Queue.** Each error generated by the instrument stores an error number and corresponding error message in the error queue. The error message can be up to 255 characters long.
- **Clearing the Error Queue:** An error number/message is removed from the queue each time the SYSTem:ERRor? query is sent. The errors are cleared first-in, first-out. When the queue is empty, each following SYSTem:ERRor? query returns 0, "No error". To clear all error numbers/messages in the queue, execute the *CLS command.
- **Maximum Error Numbers/Messages in the Error Queue:** The queue holds a maximum of 30 error numbers/messages. If the queue overflows, the last error number/message in the queue is replaced by –350, "Too many errors". The least recent error numbers/messages remain in the queue and the most recent are discarded.
- ***RST Condition:** *RST does not clear the error queue.

:HEADer

SYSTem:HEADer <mode> is used to enable or disable the output header returned with query responses. When selected, all query responses will include a command header. *mode* enables (ON | 1) or disables (OFF | 0) the command header.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	OFF|0|ON|1	none

Example

Enable command headers

```
SYST:HEAD ON
```
Command headers enabled

Comments

- **Length of Command Headers:** If command headers are enabled, use SYSTem:LONGform command to specify the header length (long form or short form). For example, response to CHANnel1:RANGe? query is as follows:

```
Headers OFF:                  6.40000E-01
Headers ON, Longform OFF:     :CHAN1:RANG 6.40000E-01
Headers ON, Longform ON:      :CHANNEL1:RANGE 6.40000E-01
```

- **Numeric Strings:** Headers should be turned OFF when returning values to numeric variables.
- **Related Commands:** SYSTem:LONGform.
- ***RST Conditions:** Defaults to OFF.

:HEADer?

SYSTem:HEADer? returns a number to show whether the command headers are enabled or disabled: "1" = enabled, "0" = disabled. The value is sent to the output buffer.

Example

Querying the command header state

`SYST:HEAD ON`	*Command headers enabled*
`SYST:HEAD?`	*Query instrument to return command header state*
`enter statement`	*Enter value into computer*

:LANGuage

SYSTem:LANGuage *<command>* used to select the programming language. COMPatible selects the HP54503A Compatible Language, and SCPI selects the Standard Commands for Programmable Instruments programming Language.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
command	discrete	COMPatible|SCPI	none

Example

Select HP 54503A compatible language to instruct the Oscilloscope

```
SYST:LANG  COMP
```
Select compatible language

Comments

- **Selecting Command:** When the HP 54503A Compatible Language (COMPatible) is selected, use the instructions found in Chapters 3 and 4 to program the Oscilloscope. When Standard Commands for Programmable Instruments (SCPI) is selected, use the instructions found in Chapters 5 and 6 to program the Oscilloscope.
- **Switching Languages:** Switching languages while programming is permissible, however:

  Allow 1 second after changing for the Oscilloscope to configure to the new language.

  After switching languages, a *RST is automatically performed to place the instrument in a known state.
- **Programming the Wrong Language:** If the Oscilloscope is configured to operate using one language, and a command from the other language is executed (with different syntax), an error will be generated.
- ***RST Condition:** *RST does not change SYSTem:LANGuage selected.

:LANGuage?

SYSTem:LANGuage? returns the current programming language selected. Returns COMPatible if the HP 54503A Compatible Language is selected, and SCPI if the Standard Commands for Programmable Instruments programming Language is selected. The data is sent to the output buffer.

Example

Querying the current programming language selected

`dimension statement`	*Dimension a string*
`SYST:LANG?`	*Query instrument to return current programming language*
`enter statement`	*Enter value into computer*

:LONGform

SYSTem:LONGform *<mode>* is used to select the format of the command header (when on) and alpha arguments sent **FROM** the Oscilloscope **TO** the controller. *mode* (ON | 1) is used to select the long form, and (OFF | 0) selects the short form.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	OFF|0|ON|1	none

Example

Select long form command headers

```
SYST:LONG  ON
```

Command headers to long form

Comments

- **Input Data Messages:** The SYSTem:LONGform command does not affect the input data messages sent **TO** the instrument. Headers and arguments may be sent to the Oscilloscope in either the longform or shortform regardless of how the LONGform command is set.
- **Selecting Command Headers:** See the SYSTem:HEADer command for more information on selecting command headers.
- **Related Commands:** SYSTem:HEADer.
- ***RST Conditions:** Defaults to OFF.

:LONGform?

SYSTem:LONGform? returns a number to show the current longform state: "1" = long form, "0" = short form. The value is sent to the output buffer.

Example

Querying the longform state

```
SYST:LONG  ON          Command headers to long form
SYST:LONG  ?           Query instrument to return long form
                       state
enter  statement       Enter value into computer
```

:SETup

SYSTem:SETup *<setup>* is used to set the Oscilloscope to a condition defined by a previously returned learn string. The learn string contains all the commands and parameters necessary to setup the instrument in one 1024 byte string.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
setup	block	binary block data in # format	none

Example

See SYSTem:SETup? query for example

Comments

- **Using SETup:** The logical order for using this instruction is to send the query first to retrieve setup data, store the data until needed, then send the learn string to the instrument using this command.
- **SETup versus *SAV/*RCL:** The SYSTem:SETup command performs the same function as the save and recall commands, except:

  Data can be saved at any location (external to the Oscilloscope) the user desires.

  No limit to the number of setups that can be saved/recalled.

:SETup?

SYSTem:SETup? returns the current learn string from the Oscilloscope. The learn string contains all the commands and parameters the Oscilloscope is currently setup to in one 1024 byte string and a header.

Example

Query setup learn string

For this example, the learn string is "#41024...."

`Dimension statement`	*Dimension string for data*
`Setup statements`	*Setup oscilloscope as desired using commands described in this chapter*
`SYST:HEAD OFF`	*Set headers to off*
`SYST:SET?`	*Query instrument to return setup learn string*
`enter statement`	*Enter value into computer*
`store statement`	*Save data*
`change instrument`	*Set the oscilloscope to perform a different function.*
`recall statement`	*Recall data*
`SYST:SET #41024....`	*Send data to the oscilloscope (recalls previous setup)*

Comments

- **Related Commands:** *LRN?, *SAV, *RCL.

TEST

The TEST command subsystem is used to perform internal diagnostics. These diagnostics are provided to give a high confidence level of instrument functionality. Before performing any of the diagnostics, execute a *RST to set critical parameters to a known state, and a SUMMary:PRESet to enable the SUMMary QUEStionable registers.

Subsystem Syntax

```
TEST
    :ACQ [<test>]
    :RAM [<test>]
    :ROM [<test>]
    :TALL
```

:ACQ

TEST:ACQ [<*test*>] is used to perform up to five acquisition tests. When selected, the Oscilloscope performs an Analog Trigger test, Logic Trigger test, an A/D test, a Time base test, and/or a D/A test. If the *test* parameter is not sent, all five tests are performed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
test	discrete	ATRigger|LTRigger| AD|TIMebase|DA	none

Example **Perform the acquisition time base test**

`TEST:ACQ TIM` *Perform Acquisition Time base Test*

Comments

- **Test Results:** Found by querying the SUMMary:QUEStionable:TEST:ACQ register.
- **Test Failure:** If any of the five acquisition tests fail, perform Firmware Calibration procedures provided in Chapter 3, and repeat the test. If the test fails again, refer to instructions found in the Service Manual.
- **Related Commands:** SUMMary:QUEStionable:TEST.

:RAM

TEST:RAM [<*test*>] is used to perform up to four random access memory tests. When selected, the Oscilloscope performs a Display RAM test, System RAM test, a Non-volatile RAM test, and/or an Acquisition RAM test. If the *test* parameter is not sent, all four tests are performed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
test	discrete	DISPlay|ACQuisition| SYSTem|NVOLatile	none

Example **Perform all four RAM tests**

```
TEST:RAM                    Perform RAM Test
```

Comments

- **Test Results:** Found by querying the SUMMary:QUEStionable:TEST:RAM register.
- **Test Failure:** If any of the four RAM tests fail, perform Firmware Calibration procedures provided in Chapter 3, and repeat the test. If the test fails again, refer to instructions found in the Service Manual.
- **Related Commands:** SUMMary:QUEStionable:TEST.

:ROM

TEST:ROM [<*test*>] is used to perform one read only memory test and one nonvolatile protected random access memory test. When selected, the Oscilloscope performs a System ROM test, and/or a Protected Non-volatile RAM test. If the *test* parameter is not sent, both tests are performed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
test	discrete	SYSTem|NVPRotect	none

Example **Perform the system ROM test**

```
TEST:ROM  SYST              Perform system ROM Test
```

Comments

- **Test Results:** Found by querying the SUMMary:QUEStionable:TEST:ROM register.
- **Test Failure:** If any of the two ROM tests fail, perform Firmware Calibration procedures provided in Chapter 3, and repeat the test. If the test fails again, refer to instructions found in the Service Manual.
- **Related Commands:** SUMMary:QUEStionable:TEST.

:TALL

TEST:TALL is used to perform the RAM, ROM, and ACQ tests. When selected, the Oscilloscope performs all the individual tests.

Example **Perform the RAM, ROM, and ACQ tests**

```
TEST:TALL                 Perform all tests
```

Comments

- **User Connection:** Disconnect all inputs prior to performing self tests.
- **Test Results:** Found by querying the SUMMary:QUEStionable:TEST register.
- **Test Failure:** If any of the tests fail, perform Firmware Calibration procedures provided in Chapter 3, and repeat the test. If the test fails again, refer to instructions found in the Service Manual.
- **Related Commands:** SUMMary:QUEStionable:TEST.

TIMebase

The TIMebase command subsystem is used to control the horizontal axis, or "X-axis," functions. A second expanded time base (Window mode) can be selected, and has separate control over position and width.

Subsystem Syntax

```
TIMebase
    :DELay <time>
    :DELay?
    :MODe <mode>
    :MODe?
    :RANGe <range>
    :RANGe?
    :REFerence <position>
    :REFerence?
    :WINDow <mode>
    :WINDow?
        :DELay <time>
        :DELay?
        :RANGe <range>
        :RANGe?
```

:DELay

TIMebase:DELay ***<time>*** is used to set the time interval between the trigger event and the active waveform delay reference point. The delay reference point is set to the left, center, or right of the active waveform using the TIMebase:REFerence command.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
time	numeric	Dependent on TIMebase:RANGe	S

Example

Set the time interval between the trigger event and the delay reference point to 2 msec

```
TIM:DEL  2E-3           Set delay to 2 msec
```

Comments

- **Entering Time:** When 0 is entered, the trigger event occurs at the delay reference point. Positive values set the trigger event to occur before the delay reference point (to capture post-trigger events). Negative values set the trigger event to occur after the delay reference point (to capture pre-trigger events). The range of acceptable DELay values is dependent on the current TIMebase:RANGe setting. If DELay is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.
- **Related Commands:** TIMebase:REFerence, RANGe.
- ***RST Condition:** Defaults to 0 seconds.

:DELay?

TIMebase:DELay? returns a number representing the current time interval between the trigger event and the delay reference point. The value (in ± seconds) is sent to the output buffer.

Example

Querying current delay value

```
TIM:DEL  2E-3           Set delay to 2 msec
TIM:DEL?                Query instrument to return delay
                        value in seconds
enter  statement        Enter value into computer
```

:MODe

TIMebase:MODe <*mode*> is used to select the time base mode. Defines when data will be acquired with respect to triggering.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	AUTo|TRIGgered|SINGle	none

Example **Set mode to only acquire data only when triggered**

```
TIM:MOD  TRIG          Set mode to triggered
```

Comments

- **Selecting Mode:** The time base *mode* is selected as follows:

  AUTo - will acquire data regardless of trigger requirements. If selected and no trigger is present, available data is acquired. Provides a baseline in the absence of a signal. If a signal is present but the instrument is not triggered, the waveform will be unsynchronized (not a baseline).

  TRIGgered - will not acquire data until all selected trigger requirements are satisfied (set using TRIGger commands). If selected and no trigger is present, the data acquired on the previous trigger will remain.

  SINGle - will clear the present waveform and stop acquiring data. When the RUN command is received, one data acquisition will occur on the next trigger.

- **Related Commands:** RUN, TRIGger subsystem.
- ***RST Condition:** Defaults to AUTo.

:MODe?

TIMebase:MODe? returns the currently selected mode under which the time base will operate. The data is sent to the output buffer. Returns AUTo, TRIGgered, or SINGle depending on the current mode selected. See TIMebase:MODe command for more information.

Example **Query current time base mode**

```
Dimension  statement   String for data
TIM:MOD  TRIG          Set mode to triggered
TIM:MOD?               Query instrument to return mode
enter  statement       Enter data into computer
```

:RANGe

TIMebase:RANGe ***<range>*** **is used to define the full scale horizontal axis, or "X-axis" of the main sweep. Controls sweep speed.**

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
range	numeric	2 NS to 50S	S

Example

Set time base range to 10 µsec (full scale)

```
TIM:RANG  10E-6         Range to 10µsec
```

Comments

- **Entering Range:** Range values are entered in a 1,2,5 sequence. If a value is entered that is not in a 1,2,5 sequence, it is automatically rounded to the closest allowable value without generating an error.
- **Effects on Other TIMebase Selections:** Changes in the range parameter may effect the current settings specified for TIMebase:DELay, and TIMebase:WINDow:RANGe.
- **Related Commands:** TIMebase:DELay, WINDow:RANGe.
- ***RST Condition:** Defaults to 1 msec.

:RANGe?

TIMebase:RANGe? returns a numeric value representing the current range setting for the horizontal axis. The value (in seconds) is sent to the output buffer.

Example

Querying full scale horizontal range setting

```
TIM:RANG  10E-6         Range to 10µsec
TIM:RANG?               Query instrument to return time base
                        range setting
enter  statement        Enter value into computer
```

:REFerence

TIMebase:REFerence *<position>* sets the delay reference to the left, right, or to the center of the active waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
position	discrete	LEFT|CENTer|RIGHt	none

Example **Set the reference to the left of the active waveform**

```
TIM:REF  LEFT            Reference to left
```

Comments

- **Selecting Position:** Position entered is used with the TIMebase:DELay command to set the time interval between the trigger event and the delay reference point. For example, if DELay is 0 seconds, and REFerence is CENTer, pre-trigger data is on the left and post-trigger data is on the right of the active waveform.
- **Related Commands:** TIMebase:DELay.

:REFerence?

TIMebase:REFerence? returns the currently selected delay reference point. The data is sent to the output buffer. Returns LEFT, CENTer, or RIGHt depending on the current position selected. See TIMebase:REFerence command for more information.

Example **Query current reference point selection**

```
Dimension  statement   String for data
TIM:REF  LEFT          Reference to left
TIM:REF  ?             Query instrument to return position
enter  statement       Enter data into computer
```

:WINDow

TIMebase:WINDow ***<mode>*** is used to enable or disable the expanded time base. *mode* enables (ON | 1) or disables (OFF | 0) the expanded time base.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	ON|OFF|1|0	none

Example

Enabling the expanded time base

```
TIM:WIND  1                    Enable the expanded time base
```

Comments

- **Enabling the Expanded Time Base:** When ON, the expanded time base is part of the active waveform, and ALL measurements are taken on the data present in the expanded time base.
- **Controlling the Expanded Time Base:** Two commands are used to set the expanded time base as follows:

  TIMebase:WINDow:DELay - controls the position of the expanded time base window on the main sweep.

  TIMebase:WINDow:RANGe - sets the full scale horizontal time for the expanded time base.
- **Retrieving Second Time Base Data:** Data is acquired from the expanded time base using the WAVeform:DATa? query.
- **Expanding Stopped Memories and Channels:** Stopped Waveforms (stored in WMEM1-4) or channels (CHAN1-4) cannot be expanded into the window.
- **Related Commands:** WAVeform:DATa?, TIMebase:WINDow:DELAY, RANGe.
- ***RST Condition:** Defaults to OFF.

:WINDow?

TIMebase:WINDow? queries the present state of the expanded time base. The query returns 1 if the expanded time base is enabled or 0 if the expanded time base is disabled. The value is sent to the output buffer.

Example

Query expanded time base state

```
TIM:WIND  1                    Enable the expanded time base
TIM:WIND?                      Query instrument  to return expanded
                               time base state
enter  statement               Enter value into computer
```

:WINDow:DELay

TIMebase:WINDow:DELay *<time>* is used to set the expanded time base delay relative to the main sweep DELay and REFerence point. The window delay actually sets the position of the expanded time base window on the main sweep.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
time	numeric	Dependent on TIMebase:DELay and TIMebase:RANGe	none

Example

Set expanded time base delay 20 nsec before the main sweep delay/reference point

```
TIM:WIND:DEL  20E-9    Delay to 20 nsec
```

Comments

- **Entering Time:** The range for time is determined by the main sweep RANGe and DELay values. The value selected MUST keep the expanded window within the boundaries of the main sweep.
- **Related Commands:** TIMebase:RANGe, DELay
- ***RST Condition:** Defaults to 0 seconds.

:WINDow:DELay?

TIMebase:WINDow:DELay? returns a number representing the current expanded time base position. The value (in ± seconds) is sent to the output buffer.

Example

Querying current expanded time base delay value

```
TIM:WIND:DEL  -20E-9   Delay to -20 nsec
TIM:WIND:DEL?          Query instrument to return delay
                       value in seconds
enter  statement       Enter value into computer
```

:WINDow:RANGe

TIMebase:WINDow:RANGe *<range>* is used to define the full scale horizontal axis, or "X-axis" of the expanded time base.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
range	numeric	2 NS to 50S	S

Example

Set expanded time base range to 10 nsec (full scale)

```
TIM:RANG  10E-9
```
Range to 10nsec

Comments

- **Entering Range:** Range values entered can be equal to the TIMebase:RANGe selection, or as low as 1/20 the current setting. Expanded time base setting cannot exceed the current TIMebase:RANGe setting. Any other values entered outside the acceptable range are changed to acceptable limits without generating an error.
- **Related Commands:** TIMebase:RANGe.
- ***RST Condition:** Defaults to 1 msec.

:WINDow:RANGe?

TIMebase:WINDow:RANGe? returns a numeric value representing the current range setting for the expanded time base horizontal axis. The value (in seconds) is sent to the output buffer.

Example

Querying expanded time base horizontal range setting

`TIM:RANG  10E-6`	*Range to 10µsec*
`TIM:RANG?`	*Query instrument to return time base range setting*
`enter  statement`	*Enter value into computer*

TRIGger

The TRIGger command subsystem is used to define the conditions for a trigger. Many of the commands in the TRIGger subsystem are used in more than one of the TRIGger MODes. If the command is a valid command for a trigger mode, that setting will be accepted. If the command is not valid for a trigger mode, an error will be generated.

See the TRIGger:MODe command for a description of all the available trigger modes, and a sequential list of all parameters that can be entered using each mode.

Auto or triggered mode is selected with the TIMebase:MODe command.

Subsystem Syntax

```
TRIGger
    :CENTered
    :CONDition <argument>
    :CONDition?
    :DELay <mode>
    :DELay?
        :SLOPe <polarity>
        :SLOPe?
        :SOURce <source>
        :SOURce?
    :FIELd <number>
    :FIELd?
    :HOLDoff <holdoff>
    :HOLDoff?
    :LEVel <level>
    :LEVel?
    :LINe <number>
    :LINe?
    :LOGic <level>
    :LOGic?
    :MODe <mode>
    :OCCurrence <number>
    :OCCurrence?
        :SLOPe <polarity>
        :SLOPe?
        :SOURce <channel>
        :SOURce?
    :PATH <channel>
    :PATH?
    :POLarity <polarity>
    :POLarity?
    :QUALify <mode>
    :QUALify?
    :SENSitivity <mode>
    :SENSitivity?
    :SLOPe <polarity>
    :SLOPe?
    :SOURce <source>
    :SOURce?
    :STANdard <standard>
    :STANdard?
```

:CENTered

TRIGger:CENTered used to automatically set the trigger level to the current vertical offset value for the channel selected. Can be used in all TRIGger:MODes.

Example **Set trigger to 50% level**

```
TRIG:CENT
```

Trigger level set to center

Comments

- **Query Trigger Level:** Use the TRIGger:LEVel? query to return currently selected trigger level.
- **Related Commands:** TRIGger:LEVel.

:CONDItion

TRIGger:CONDition ***<argument>*** is used to specify a set of conditions that must be satisfied to generate a trigger event. Can be used in PATTern, STATe, DELay, and TV TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
argument	discrete	ENTer|EXIT|GT,<*value*>|LT,<*value*>| RANGe,<*range_gt*>,<*range_lt*>| TRUE | FALSe	none
value	numeric	20NS to 160MS	S
range_gt	numeric	20NS to 159.999MS	S
range_lt	numeric	30NS to 160MS	S

Example **Set time range from 22 msec to 33 msec (valid for pattern, delay, and TV trigger modes)**

```
TRIG:COND  RANG,22E-6,33E-6
```

Set range from 22msec to 33msec

Comments

- **Entering Argument:** Purpose, selection and value that can be entered for *argument* are dependent on the TRIGger:MODe selected.

  TRIGger:MODe PATTern: Valid *arguments* are ENTer | EXIT | GT,<value> | LT,<value> | RANGE,*<range_gt>*, *<range_lt>*.

  ENTer: When specified, a trigger is generated on the first transition that makes a specified logic pattern true. The pattern must be false and then go true to generate the trigger.

  EXIT: When specified, a trigger is generated on the first transition that makes a specified logic pattern false. The pattern must be true and then go false to generate the trigger.

GT,<*value*>: When specified, a trigger is generated when the logic pattern is true for longer than the *value* specified. Time *values* entered are rounded to the nearest 10 nsec.

LT,<*value*>: When specified, a trigger is generated when the logic pattern is true for less than the *value* specified. Time *values* entered are rounded to the nearest 10 nsec.

RANGe,<*range_gt*>,<*range_lt*>: When specified, a trigger is generated when the logic pattern is true within the time range specified. Time *range* entered is rounded to the nearest 10 nsec. *range_gt* must not exceed *range_lt*.

TRIGger:MODe STATe: Valid *arguments* are TRUE and FALSe.

TRUE: When specified, a trigger is generated when the logic pattern is true.

FALSe: When specified, a trigger is generated when the logic pattern is false.

TRIGger:MODe DELay: CONDition command can only be used when PATTern or STATe is selected as the qualifier using the TRIGger:QUALify command. See PATTern or STATe *arguments* above for information on selecting conditions.

TRIGger:MODe TV: CONDition command can only be used when TRIGger:STANdard USER is selected. *argument* is RANGe,<*range_gt*>,<*range_lt*>.

RANGe,<*range_gt*>,<*range_lt*>: When specified, a time range is set for the trigger to occur. Time *range* entered is rounded to the nearest 10 nsec. *range_gt* must not exceed *range_lt*.

- **Related Commands:** TRIGger:MODe, STANdard, QUALify.

:CONDition?

TRIGger:CONDition? is used to return the condition currently selected. The data is sent to the output buffer. Returns ENTer, EXIT, GT,<*value*>, LT,<*value*>, RANGe,<*range_gt*>,<*range_lt*> dependent on current TRIGger:MODe selected. *value* is time in seconds from 20nsec to 160msec. *range_gt* is time in seconds from 20nsec to 159.999msec. *range_lt* is time in seconds from 30nsec to 160msec. See TRIGger:CONDition command for more information.

Example **Query the current condition selection**

```
dimension statement   String to hold data
TRIG:COND?            Query instrument to return condition
enter statement       Enter data into computer
```

Comments

- **Related Commands:** TRIGger:MODe.

:DELay

TRIGger:DELay ***<delay>*** is used to set a delay value in time or number of events. Disables the trigger circuit for a specified period of time, or number of events after the trigger has been qualified. Can be used only in the DELay TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
delay	discrete	TIMe,*<time>*|EVENt,*<event>*	none
time	numeric	30NS to 160MS	S
event	numeric	1 to 16,000,000	none

Example

Disable trigger circuit for 5 events after trigger is qualified

```
TRIG:DEL  EVEN,5         Delay to 5 events
```

Comments

- **Selecting Delay:** *delay* is specified as a period of time or number of events using the following guidelines:

  TIMe,*<time>*: Disables the trigger circuit for from 30nsec to 160msec after the trigger has been qualified. Time delay is not available with time qualified pattern settings of GT, LT, or RANGe.

  EVENt,*<event>*: Disables the trigger circuit for from 1 to 16,00,000 counts after the trigger has been qualified. After the selected count is reached, the instrument will look for the user specified edge.

- **Qualifying the Trigger:** The mode used to qualify the trigger before a delay is selected using the TRIGger:QUALify command.
- **User Specified Edge:** The user specified edge is selected using the TRIGger:DELay:SOURce and SLOPe commands:
- **Related Commands:** TRIGger:DELay:SOURce, SLOPe, TRIGger:QUALify.

:DELay?

TRIGger:DELay? is used to return the currently selected delay time or number. The data is sent to the output buffer. Returns TIMe,*<time>* if the current delay is set to time, where *time* is from 30nsec to 160 msec seconds. Returns EVENt,*<event>* if the current delay is set to number of events, where *event* is from 1 to 16,000,000. See TRIGger:DELay command for more information.

Example

Query the current delay selection

```
dimension  statement    String to hold data
TRIG:DEL  EVEN,5        Delay to 5 events
TRIG:DEL?               Query instrument to return delay
                        setting
enter  statement        Enter data into computer
```

:DELay:SLOPe

TRIGger:DELay:SLOPe <*polarity*> is used to select the edge that will be counted by the DELay EVENt command. Can be used only in in the DELay TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
polarity	discrete	POSitive|NEGative	none

Example **Set delay edge to rising**

```
TRIG:DEL:SLOP  POS        Delay slope to positive
```

Comments

- **Selecting Polarity:** Enter POSitive to select the rising edge, and NEGative to select the falling edge.
- **Related Commands:** TRIGger:DELay, DELay:SOURCE.

:DELay:SLOPe?

TRIGger:DELay:SLOPe? returns the currently selected delay edge polarity that will be counted by the DELay EVENt command. The data is sent to the output buffer. Returns POSitive if the rising edge is selected, or NEGative if falling edge is selected.

Example **Query the current delay slope selection**

```
dimension  statement  String to hold data
TRIG:DEL:SLOP  POS    Delay slope to positive
TRIG:DEL:SLOP?        Query instrument to return delay
                      slope setting
enter  statement      Enter data into computer
```

:DELay:SOURce

TRIGger:DELay:SOURce ***<source>*** is used to select the source that will be counted by the DELay EVENt command. Can be used only in the DELay TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	CHANneln (*n*=1 to 4)	none

Example

Set delay source to channel 3

```
TRIG:DEL:SOUR  CHAN3
```
Delay source to channel 3

Comments

- **Related Commands:** TRIGger:DELay, DELay:SLOPe.

:DELay:SOURce?

TRIGger:DELay:SOURce? returns the currently selected delay source (CHANnel1-4). The data is sent to the output buffer.

Example

Query the current delay source selection

`dimension statement`	*String to hold data*
`TRIG:DEL:SOUR CHAN3`	*Delay source to channel 3*
`TRIG:DEL:SOUR?`	*Query instrument to return delay source setting*
`enter statement`	*Enter data into computer*

:FIELd

TRIGger:FIELd *<number>* is used to select the field of the TV signal. Can be used only in the TV TRIGger:MODe, and when TRIGger:STANdard is 525 or 625.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 or 2	none

Example **Set field of the TV signal to 1**

```
TRIG:FIEL  1            Set field to 1
```

Comments

- **Selecting Field:** The field *number* entered will determine the TRIGger:LINe selections available.
- **Related Commands:** TRIGger:STANdard, LINe.

:FIELd?

TRIGger:FIELd? returns the currently selected field (1 or 2). The value is sent to the output buffer.

Example **Query the current field selection**

```
TRIG:FIEL  1            Set field to 1
TRIG:FIEL?              Query instrument to return field
                        setting
enter  statement        Enter data into computer
```

:HOLDoff

TRIGger:HOLDoff *<holdoff>* is used to set a holdoff value in time or number of events. Disables the trigger circuit for a specified period of time, or number of events after the trigger event. Can be used only in the EDGE, PATTern, STATe, or TV TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
holdoff	discrete	TIMe,<*time*>|EVENt,<*event*>	none
time	numeric	40NS to 320MS	S
event	numeric	2 to 16,000,000	none

Example

Disables the trigger circuit for 50nsec after trigger event

```
TRIG:HOLD  TIM,50E-9  Holdoff to 50 nsec
```

Comments

- **Selecting Holdoff:** *holdoff* is specified as a period of time or number of events using the following guidelines:

  TIMe,<*time*>: Disables the trigger circuit for from 40nsec to 320msec after the trigger event.

  EVENt,<*event*>: Disables the trigger circuit for from 2 to 16,00,000 counts after the trigger event.

- **Related Commands:** TRIGger:MODe.
- ***RST Conditions:** Defaults to TIMe, 40 nsec.

:HOLDoff?

TRIGger:HOLDoff? is used to return the currently selected holdoff time or number. The data is sent to the output buffer. Returns TIMe,<*time*> if the current holdoff is set to time, where *time* is from 40nsec to 320 msec. Returns EVENt,<*event*> if the current holdoff is set to number of events, where *event* is from 2 to 16,000,000. See TRIGger:HOLDoff command for more information.

Example

Query the current holdoff selection

```
dimension  statement  String to hold data
TRIG:HOLD  TIM,50E-9  Holdoff to 50 nsec
TRIG:HOLD?            Query instrument to return holdoff
                      setting
enter  statement      Enter data into computer
```

:LEVel

TRIGger:LEVel *<level>* is used to set the trigger level voltage of the active trigger. Used for all TRIGger:MODes.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
level	numeric	See below	V

Example

Set trigger level to 1 volt

```
TRIG:LEV  1                 Trigger level to 1 volt
```

Comments

- **Selecting Level:** *level* can be entered to a value that is ±0.75 of the current CHANNel<n>:RANGe setting **from** the current CHANNel<n>:OFFSet setting.
- **Trigger Level versus Mode:** The trigger *level* can be sent in any mode, however only two separate levels are stored. One value is kept for the TV mode and another value is kept for all other modes. If you are in the PATTern mode and set a trigger level value, that level will also be used for the EDGe, STATe, and DELay modes.
- **Trigger Level Source:** The trigger *level* source is selected using the TRIGger:SOURce command.
- **Related Commands:** TRIGger:MODe.
- ***RST Conditions:** Defaults to 0 volts.

:LEVel?

TRIGger:LEVel? returns the currently selected trigger level (in volts) of the current trigger mode. The value is sent to the output buffer.

Example

Query the current trigger level

```
TRIG:LEV  1                 Trigger level to 1 volt
TRIG:LEV?                   Query instrument to return trigger
                            level
enter  statement            Enter data into computer
```

:LINe

TRIGger:LINe <*number*> is used to set which line in the selected FIELd the trigger will be generated on. Can be used only in the TV TRIGger: MODe when TRIGger:STANdard is 525 or 625.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 625	none

Example

Set 35th line of field 1 to generate the trigger

`TRIG:STAN 525`	*Set standard to 525*
`TRIG:FIEL 1`	*Set field to 1*
`TRIG:LIN 35`	*Set line to 35*

Comments

- **Selecting Line:** The line *number* parameter is dependent on the current TRIGger:STANdard and FIELd selection.

Standard to 60Hz/525 and Field to 1	1 to 263
Standard to 60Hz/525 and Field to 2	1 to 262
Standard to 50Hz/625 and Field to 1	1 to 313
Standard to 50Hz/625 and Field to 2	314 to 625

- **Related Commands:** TRIGger:STANdard, FIELd.

:LINe?

TRIGger:LINe? returns the currently selected line number. The number (1 to 625) is sent to the output buffer. See TRIGger:LINe command for more information.

Example

Query the current line selection

`TRIG:LIN 35`	*Set line to 35*
`TRIG:FIEL?`	*Query instrument to return line setting*
`enter statement`	*Enter data into computer*

:LOGic

TRIGger:LOGic ***<level>*** is used to set the "trigger on" logic level of the currently enabled path. Specifies the relation between the signal (TRIGger:PATH) and the defined voltage level (TRIGger:LEVel) that must exist before that bit of the pattern is considered valid. When all bits (up to 4) are valid, a trigger event is generated. Can be used in the PATTern, STATe, or DELay (when qualifying with PATTern or STATe) TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
level	discrete	HIGH|LOW|DONTcare	none

Example **Set logic level of the currently enabled path to high**

```
TRIG:LOG  HIGH
```
Logic level to high

Comments

- **Selecting Level:** Use the following guidelines to select *level*:

  HIGH: If the signal on a selected path must be greater than the trigger level.

  LOW: If the signal on a selected path must be lower than the trigger level.

  DONTcare: Signal on a selected path disregarded.

- **Defining Pattern:** Both the TRIGger:PATH and LOGic commands are used to define trigger pattern. The PATH command must be executed prior to the LOGic command. For example, to set path 2 at logic low, send "TRIGger:PATH CHAN2;LOGic LOW".
- **Related Commands:** TRIGger:PATH, LOGic, LEVel, MODe.

:LOGic?

TRIGger:LOGic? returns the selected logic level (HIGH, LOW, or DONTcare) of the currently enabled path. The data is sent to the output buffer. See TRIGger:LOGic and TRIGger:PATH commands for more information.

Example **Query the current logic level selection**

`dimension statement`	*String to hold data*
`TRIG:LOG HIGH`	*Logic level to high*
`TRIG:LOG?`	*Query instrument to return logic setting*
`enter statement`	*Enter data into computer*

:MODe

TRIGger:MODe *<mode>* is used to select the edge, pattern, state, delay, or TV trigger modes.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	EDGe|PATTern|STATe| DELay|TV	none

Example

Set trigger mode to pattern

```
TRIG:MOD  PATT
```

Pattern trigger mode enabled

Comments

- **Selecting Mode —** Five trigger modes provide many distinctive techniques to trigger and capture data. Select the desired trigger mode using the following guidelines:

  :EDGe — Provides simple edge triggering. Easiest mode to understand and use. Use the following TRIGger commands in the order presented to setup EDGe triggering.

  :SOURce — Use to select the channel that the instrument will trigger on. See TRIGger:SOURce for more information.

  :LEVel — Use to select the trigger level that the instrument will trigger on. Can be set for each trigger source. See TRIGger:LEVel and CENTer for more information.

  :SLOPe — Use to select the actual edge that will create the trigger. Can be set for each trigger source. See TRIGger:SLOPe for more information.

  :SENSitivity — Use to select noise reject on or off. Can be set for each trigger source. See TRIGger:SENSitivity for more information.

  :HOLDoff — Use to select the time or number of events to holdoff after the trigger event. See TRIGger:HOLDoff for more information.

  :PATTern — Defines up to four patterns for the instrument to recognize, and then generate a trigger event. Use the following TRIGger commands in the order presented to setup PATTern triggering.

  Up to four logic patterns are defined using the PATH and LOGic commands.

  :PATH — Use to select which of the four inputs are used for the logic pattern. See TRIGger:PATH for more information.

  :LOGic — Use to select the "logic" conditions that must be satisfied. Level is set for each active path. See TRIGger:LOGic for more information.

:LEVel — Use to select the trigger level that the instrument will use to determine logic level. Level is set for each active path. See TRIGger:LEVel and CENTer for more information.

:SENSitivity — Use to select noise reject on or off. Can be set for each trigger source. See TRIGger:SENSitivity for more information.

:CONDition — Use to select the "when" conditions that must be satisfied before a trigger event is generated. See TRIGger:CONDition for more information.

:HOLDoff — Use to select the time or number of events to holdoff after the trigger event. See TRIGger:HOLDoff for more information.

:STATe — Similar to PATTern mode, except one input is selected as a clock edge and the other three inputs define a pattern. When the pattern becomes true, the instrument triggers on the next clock edge. Use the following TRIGger commands in the order presented to setup STATe triggering.

:SOURce — Use to select the channel for the clock source. See TRIGger:SOURce for more information.

:SLOPe — Use to select the edge for the clock source. See TRIGger:SLOPe for more information.

Up to three logic patterns are defined using the PATH and LOGic commands.

:PATH — Use to select which of the three inputs are used for the logic pattern. See TRIGger:PATH for more information.

:LOGic — Use to select the "logic" conditions that must be satisfied. Level is set for each active path. See TRIGger:LOGic for more information.

:LEVel — Use to select the trigger level that the instrument will use to determine logic level. Level is set for each active path. See TRIGger:LEVel and CENTer for more information.

:SENSitivity — Use to select noise reject on or off. Can be set for each trigger source. See TRIGger:SENSitivity for more information.

:CONDition — Use to select the true/false condition that must be satisfied before a trigger event is generated. See TRIGger:CONDition for more information.

:HOLDoff — Use to select the time or number of events to holdoff after the trigger event. See TRIGger:HOLDoff for more information.

:DELay — Qualifies on a signal (edge, pattern, or state), delays for a period of time or occurrence of events, and then enable a trigger event on a selected edge from any source. Use the following TRIGger commands in the order presented to setup DELay triggering.

> **:QUALify** — Use to select which mode (EDGe, PATTern, or STATe) to qualify the trigger before a delay is defined. Selection of these modes is described above. See TRIGger:QUALify for more information.
>
> **:DELay** — Use to select the type (time or event) and amount of delay. If events are selected, the source and slope must also be specified. See TRIGger:DELay for more information.
>
> **:OCCurrence** — Use to select the source, slope, and number of trigger events that occur before the sweep is triggered. See TRIGger:OCCurrence for more information.

:TV — Used for triggering on clamped television signals. This mode allows selection of one TV signal frame and one of the lines within that frame. Use the following TRIGger commands in the order presented to setup TV triggering.

> **:STANdard** — Use to select the TV standard signal. See TRIGger:STANdard for more information.
>
> **:SOURce** — Use to select the channel that the instrument will trigger on. See TRIGger:SOURce for more information.
>
> **:LEVel** — Use to select the trigger level that the instrument will trigger on. See TRIGger:LEVel and CENTer for more information.
>
> **:SENSitivity** — Use to select noise reject on or off. Can be set for each trigger source. See TRIGger:SENSitivity for more information.
>
> **:POLarity** — Use to select the edge that will create the trigger. See TRIGger:POLarity for more information.
>
> **:FIELd** — Use to select the field that will create the trigger. See TRIGger:FIELd for more information.
>
> **:LINe** — Use to select the line in the field that will create the trigger. See TRIGger:LINe for more information.
>
> **:HOLDoff** — Use to select the time or number of events to holdoff after the trigger event. See TRIGger:HOLDoff for more information.

:QUALify — Use to select the qualify on field. See TRIGger:QUALify for more information.

:CONDition — Use to select a range that the qualify on field must occur in before a trigger event is generated. See TRIGger:CONDition for more information.

:OCCurrence — Use to select the source, slope, and number of trigger events that occur before the sweep is triggered. See TRIGger:OCCurrence for more information.

- ***RST Conditions:** Defaults to EDGe.

:MODe?

TRIGger:MODe? is used to return the currently selected trigger mode. Returns EDGe, PATTern, STATe, DELay, or TV. The value is sent to the output buffer. See TRIGger:MODe command for more information.

Example **Return trigger mode currently selected**

`dimension statement`	*String for data*
`TRIG:MOD PATT`	*Pattern trigger mode enabled*
`TRIG:MOD?`	*Query instrument to return trigger mode setting*
`enter statement`	*Enter data into computer*

:OCCurrence

TRIGger:OCCurrence *<number>* is used to set the number of trigger events that must occur before the sweep is actually triggered. Can be used only in the DELay or TV (with STANdard USER selected) TRIGger: MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 16,000,000	none

Example

Trigger sweep after 100 trigger events

```
TRIG:OCC  100
```
Occurrence to 100 events

Comments

- **Selecting Source:** The source to trigger on is selected using the TRIGger:OCCurrence:SOURce command.
- **Selecting Edge:** The specific edge to trigger on is selected using the TRIGger:OCCurrence:SLOPe command.
- **Related Commands:** TRIGger:OCCurrence:SOURce, SLOPe.

:OCCurrence?

TRIGger:OCCurrence? is used to return the currently selected occurrence number from 1 to 16,000,000. The value is sent to the output buffer. See TRIGger:OCCurrence command for more information.

Example

Query the current occurrence selection

`TRIG:OCC  100`	*Occurrence to 100 events*
`TRIG:OCC?`	*Query instrument to return occurrence setting*
`enter  statement`	*Enter data into computer*

:OCCurrence:SLOPe

TRIGger:OCCurrence:SLOPe <*polarity*> is used to select the edge that will be counted by the OCCurrence command. Can be used only in the DELay or TV TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
polarity	discrete	POSitive|NEGative	none

Example

Set occurrence edge to falling

```
TRIG:OCC:SLOP  NEG      Occurrence slope to negative
```

Comments

- **Selecting Polarity:** Enter POSitive to select the rising edge, and NEGative to select the falling edge.
- **Related Commands:** TRIGger:OCCurrence, OCCurrence:SOURce.

:OCCurrence:SLOPe?

TRIGger:OCCurrence:SLOPe? returns the currently selected occurrence edge. The data is sent to the output buffer. Returns POSitive if the rising edge is selected, or NEGative if falling edge is selected.

Example

Query the current occurrence slope selection

```
dimension  statement  String to hold data
TRIG:OCC:SLOP  NEG    Occurrence slope to negative
TRIG:OCC:SLOP?        Query instrument to return
                      occurrence slope setting
enter  statement      Enter data into computer
```

:OCCurrence:SOURce

TRIGger:OCCurrence:SOURce *<source>* is used to select the source that will be counted by the OCCurrence command. Can be used only in the DELay TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	CHANNel*n* (*n*=1 to 4)	none

Example

Set occurrence source to channel 2

```
TRIG:OCC:SOUR   CHAN2  Occurrence source to channel 2
```

Comments

- **Related Commands:** TRIGger:OCCurrence, OCCurrence:SLOPe.

:OCCurrence:SOURce?

TRIGger:OCCurrence:SOURce? returns the currently selected occurrence source (CHANnel1-4). The data is sent to the output buffer.

Example

Query the current occurrence source selection

```
dimension  statement  String to hold data
TRIG:OCC:SOUR   CHAN2  Occurrence source to channel 2
TRIG:OCC:SOUR?         Query instrument to return
                       occurrence source setting
enter   statement      Enter data into computer
```

:PATH

TRIGger:PATH *<channel>* used to select a pattern bit as the source for LOGic commands. Can be used in the PATTern, STATe, or DELay (when qualifying with PATTern or STATe) TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel	discrete	CHANnel*n* (*n*=1 to 4)	none

Example

Set path to channel 2

```
TRIG:PATH   CHAN2          Path to channel 2
```

Comments

- **Defining Pattern:** Both the TRIGger:PATH and LOGic commands are used to define trigger pattern. PATH must be executed prior to the LOGic command. For example, to set path 2 at logic low, send "TRIGger:PATH CHAN2;LOGic LOW".
- **Related Commands:** TRIGger:LOGic, LEVel, MODe.

:PATH?

TRIGger:PATH? returns the currently selected trigger source (CHANnel1-4) for the present mode. The data is sent to the output buffer. See TRIGger:LOGic and TRIGger:PATH commands for more information.

Example

Query the current trigger source

```
dimension  statement   String to hold data
TRIG:PATH  CHAN2       Path to channel 2
TRIG:PATH?             Query instrument to return source
enter  statement       Enter data into computer
```

:POLarity

TRIGger:POLarity *<polarity>* is used to select the edge for the trigger. Can be used only in the TV TRIGger:MODe when STANdard 525 or 625 is selected.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
polarity	discrete	POSitive|NEGative	none

Example

Set TV trigger polarity to rising

```
TRIG:POL  POS               Polarity to positive
```

Comments

- **Selecting Polarity:** Enter POSitive to select the rising edge, and NEGative to select the falling sync pulse to trigger on.
- **Related Commands:** TRIGger:MODe.

:POLarity?

TRIGger:POLarity? returns the currently selected polarity setting. The data is sent to the output buffer. Returns POSitive if the rising edge is selected, or NEGative if falling sync pulse is selected.

Example

Query the current polarity selection

```
dimension  statement    String to hold data
TRIG:POL  POS           Polarity to positive
TRIG:POL?               Query instrument to return polarity
                        setting
enter  statement        Enter data into computer
```

:QUALify

TRIGger:QUALify *<mode>* is used to select a mode to qualify the trigger before a delay is defined in the DELay TRIGger:MODe. Used to set the qualify on field for the TV TRIGger:MODe when STANdard USER selected.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	EDGe|PATTern|STATe|LOW|HIGH	none

Example

Qualify delay trigger on pattern

```
TRIG:QUAL  PATT          Qualify on pattern
```

Comments

- **Selecting Mode:** *mode* is specified depending on the current TRIGger:MODe selected. See TRIGger:MODe command for more information on each of the available modes.

  EDGe: Available only in DELay mode. When selected, all TRIGger:MODe EDGe parameters and commands can be used to set the source and slope.

  PATTern: Available only in DELay mode. When selected, all TRIGger:MODe PATTern commands can be used to set the pattern mode parameters.

  STATe: Available only in DELay mode. When selected, all TRIGger:MODe STATe commands can be used to set the state mode parameters.

  HIGH: Available only in TV mode with STANdard set to USER. Uses the current TRIGger:LEVel for the specified channel as the reference.

  LOW: Available only in TV mode with STANdard set to USER. Uses the current TRIGger:LEVel for the specified channel as the reference.

- **Related Commands:** TRIGger:MODe, STANdard, CONDition.

:QUALify?

TRIGger:QUALify? returns the selected qualify setting in the currently selected mode. The data is sent to the output buffer. Returns EDGe, PATTern, or STATe to specify delay trigger qualify mode, and HIGH or LOW to specify edge in the TV trigger mode. See TRIGger:QUALify command for more information.

Example

Query the current qualify selection

```
dimension  statement   String to hold data
TRIG:QUAL  PATT        Qualify on pattern
TRIG:QUAL?             Query instrument to return setting
enter  statement       Enter data into computer
```

:SENSitivity

TRIGger:SENSitivity *<mode>* sets the trigger sensitivity for the selected source. Used to avoid false triggering on noisy signals. NORMal corresponds to noise reject off and LOW corresponds to noise reject on.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	NORMal|LOW	none

Example **Set trigger sensitivity to low (noise reject to on)**

```
TRIG:SENS  LOW          Low trigger sensitivity
```

Comments

- **Specifying Source:** Select the source using the TRIGger:SOURce command.
- **Related Commands:** TRIGger:LEVel, SOURce.
- ***RST Conditions:** Defaults to NORMal.

:SENSitivity?

TRIGger:SENSitivity? returns the selected sensitivity setting for the currently selected source. The data is sent to the output buffer. Returns NORMal with noise reject off and LOW with noise reject on.

Example **Query the current sensitivity selection**

```
dimension  statement    String to hold data
TRIG:SENS  LOW          Low trigger sensitivity
TRIG:SENS?              Query instrument to return sensitivity
                        setting
enter  statement        Enter data into computer
```

:SLOPe

TRIGger:SLOPe *<polarity>* is used to select the edge for the trigger. Can be used only in the EDGe, STATe, and DELay (when qualified on EDGe or STATe) TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
polarity	discrete	POSitive|NEGative	none

Example

Set trigger edge to rising

```
TRIG:SLOP  POS          Slope to positive
```

Comments

- **Selecting Slope:** Enter POSitive to select the rising edge, and NEGative to select the falling edge.
- **Related Commands:** TRIGger:SOURce.
- ***RST Condition:** Defaults to POSitive.

SLOPe?

TRIGger:SLOPe? returns the selected trigger edge for the currently selected trigger mode. The data is sent to the output buffer. Returns POSitive if the rising edge is selected, or NEGative if falling edge is selected.

Example

Query the current slope selection

```
dimension  statement   String to hold data
TRIG:SLOP  POS         Slope to positive
TRIG:SLOP?             Query instrument to return slope
                       setting
enter  statement       Enter data into computer
```

:SOURce

TRIGger:SOURce *<source>* is used to select the source that will actually produce the trigger. Can be used only in the EDGe, STATe, TV, and DELay (when qualified on EDGe or STATe) TRIGger: MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	CHANnel*n* (*n*=1 to 4)| TTLTrg*n* (*n*=0 to 7)| ECLTrg*n* (*n*=0 to 1)	none

Example

Set trigger source to TTL trigger bus line 5

```
TRIG:HOLD  TIM,40NS     Holdoff to 40 nsec
TRIG:MOD  EDG           Trigger mode to edge
TRIG:SOUR  TTLT5        Source to TTL trigger bus line 5
```

Comments

- **Selecting Source:** *source* is specified depending on the TRIGger:MODe currently selected.

  CHANnel1-4: Available on all modes except PATTern. Source is input connectors 1-4 on the instrument panel.

  TTLTrg0-7: Available only in EDGe mode and when TRIGger:HOLDoff is set to TIMe (EVENt generates an error). Other trigger commands have no effect on the TTL trigger source. Source is backplane TTL trigger bus lines 0-7.

  ECLTrg0-1: Available only in EDGe mode and when TRIGger:HOLDoff is set to TIMe (EVENt generates an error). Other trigger commands have no effect on the ECL trigger source. Source is backplane ECL trigger bus lines 0-1.

- **AUToscale:** Autoscale selects the trigger source from CHANnel1-4 only. TTLTrg and ECLTrg lines cannot be used for an autoscale.

- ***RST Condition:** Defaults to CHANnel1.

:SOURce?

TRIGger:SOURce? returns the selected source (CHANnel1-4, TTLTrg0-7, or ECLTrg0-1) for the currently selected trigger mode. The data is sent to the output buffer.

Example

Query the current trigger source selection

```
dimension  statement    String to hold data
TRIG:SOUR  TTLT5        Source to TTL trigger bus line 5
TRIG:SOUR?              Query instrument to return trigger
                        source setting
enter  statement        Enter data into computer
```

:STANdard

TRIGger:STANdard *<standard>* is used to select the television signal standard to be used. Can be used only in the TV TRIGger:MODe.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
standard	discrete	525|625|USER	none

Example

Set mode so user can define measurement standards

```
TRIG:STAN  USER          Set standard to USER
```

Comments

- **Selecting Standard:** Standard is selected using the following guidelines:

  525 - Standard TV signal used in the United States. 60Hz and 525 lines per frame.

  625 - Standard TV signal used in Europe. 50Hz and 625 lines per frame.

  USER - Allows the user to define ranges of the TV signal. When selected, user inputs parameters with the TRIGger:POLarity, TRIGger:QUALify, and TRIGger:CONDition commands.

- **Related Commands:** TRIGger:CONDition, POLarity, QUALify, LINe, and FIELD.

:STANdard?

TRIGger:STANdard? returns the currently selected standard (525, 625, or USER). The data is sent to the output buffer. See TRIGger:STANdard command for more information.

Example

Query the current standard selection

```
dimension  statement     String to hold data
TRIG:STAN  USER          Set standard to USER
TRIG:STAN?               Query instrument to return standard
                         setting
enter  statement         Enter data into computer
```

WAVeform

The WAVeform command subsystem is used to transfer waveform data between the bus and the instrument's waveform memories. The waveform record is actually contained in two portions, the waveform data and the preamble.

> The waveform data is the actual data acquired for each point in the specified source.
>
> The preamble contains the information for interpreting the waveform data. This includes the number of points acquired, format of acquired data, and type of acquired data. The preamble also contains the X and Y increments, origins, and references for the acquired data, so that the returned data can be translated to time and voltage values.

The waveform data and preamble must be read (by the controller) or sent (to the instrument) with two separate commands, DATa and PREamble.

Note

The term "Time Buckets" is defined as — the time range divided into a specific number of horizontal time points (as defined by ACQuire:POINts), each with an equal and fixed time associated with it.

Subsystem Syntax

```
WAVeform
    :COUNt?
    :DATa <data>
    :DATa?
    :FORMat <format>
    :FORMat?
    :POINts?
    :PREamble <data>
    :PREamble?
    :SOURce <source>
    :SOURce?
    :TYPe?
    :XINCrement?
    :XORigin?
    :XREFerence?
    :YINCrement?
    :YORigin?
    :YREFerence?
```

:COUNt?

WAVeform:COUNt? always returns 1. This query has no effect on instrument operations, and is only included for compatibility with other instruments.

:DATa

WAVeform:DATa <*data*> is used to send the instrument a waveform data record over the bus and store it in the previously specified waveform memory.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Unit
data	block	binary block data in # format	none

Example

Send the Oscilloscope waveform data to waveform memory 1

For the example, waveform data is "#41024...."

```
WAV:SOUR  WMEM1        Select waveform memory 1
WAV:DAT   #41024....   Send waveform data to Oscilloscope
```

Comments

- **Storing Waveform Data:** Only Waveform MEMories (WMEM1-4) may have waveform data sent to them. Select the desired location using the WAVeform:SOURce command.
- **Waveform Data Format:** The format of the waveform data being sent must match the format previously specified by the waveforms preamble for the destination memory. See WAVeform:FORMat command for more information.
- **Related Commands:** WAVeform:PREamble.

:DATa?

WAVeform:DATa? is used to output the waveform data record stored in the instruments previously specified WMEMory<n>, CHANnel<n>, or FUNCtion<n> over the bus.

Example

Send waveform data from channel 1 over the bus

The following example illustrates the use of the WAVeform:DATa? query only. Chapter 3 contains an example of performing a complete digitizing operation, and also the procedure used to read the block length then re-define a string to hold the data.

`Dimension statement`	*Dimension string to hold data (1024 bytes)*
`WAV:SOUR CHAN1`	*Select channel 1*
`WAV:DAT?`	*Enter value into computer (see Chapter 3, Digitize example, for the procedure to read block length)*
`enter statement`	*Enter data into computer*

Comments

- **Waveform Data:** Waveform MEMories (WMEM1-4), channel buffers (CHAN1-4), or FUNCtions (FUNC1-2) may have waveform data sent from them. Select the desired location using the WAVeform:SOURce command.
- **Waveform Data Format:** The format of the waveform data being sent is specified using the WAVeform:FORMat command.
- **Interpreting Waveform Data:** In order to obtain useful information from the waveform data, the returned data must be scaled. The information necessary for scaling (X/Y) the waveform is contained in the preamble data.

  Conversion from Data Value to Voltage: The formula to convert a data value from the specified source to a voltage value is:

  voltage = [(data value - yreference)•yincrement] + yorigin

  Conversion from Data Value to Time: The time value of a data point can be determined by the position of the data point. The formula to convert a data point from the specified source to a time value is:

  time = [(data point number-xreference)•xincrement]+xorigin

  As an example, the third data point sent with XORigin = 16 ns, XREFerence = 0, and XINCrement = 2 ns, would result in the following calculation:

  time = [(3 – 0) • 2 ns] + 16 ns = 22 ns
- **Waveform Preamble:** The preamble should be read before the waveform data.
- **Related Commands:** WAVeform:PREamble, FORMat.

:FORMat

WAVeform:FORMat *<format>* is used to specify how the data is formatted on the bus when sent from the instrument, and set the data transmission mode for waveform data output.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
format	discrete	WORD|BYTE|COMPressed	none

Example

Format waveform data (from oscilloscope over the bus) in word

The following example illustrates the use of the WAVeform:FORMat command only. Chapter 3 contains an example on performing a complete digitizing operation.

`WAV:FORM WORD`	*Waveform data sent over the bus will be in word format*

Comments

- **Block Data:** Formatted waveform records are transmitted using the definite block program data format specified in IEEE 488.2. When using this format, the ASCII character string "#6<DD..D>" is sent before the actual data. The 6 indicates how many <D> 's will follow. The < D > 's are ASCII numbers, which indicate how many data bytes will follow.

  For example, if 512 points were acquired the Block Header "#3512" would be sent. The 3 indicates that three length bytes follow, 512 indicates that 512 data bytes (binary) follow.

- **Selecting Format:** Format is selected using the following guidelines:

  WORD: Useful in applications where the information is read directly into an integer array in a controller. This format also returns the most accurate data values and greatest resolution. Formatted data transfers as 16-bit binary integers in two bytes. The number of data bytes is twice the number of words (data points). The most significant byte of each word is sent first. If there is a hole in the data, it will be represented by the 16-bit value of –1. The range of data in the WORD format is from 0 to 32640.

  BYTE: Will transfer over the bus faster than WORD formatted data, but has less resolution. Only seven bits are used to represent the voltage values. If there is a hole in the data, it is represented by a value of –1.

  COMPressed: Gives greater vertical precision than BYTE formatted data, with faster transfer times than WORD formatted data. The number of data bytes is equal to the number of data points. Eight bits of resolution are retained. So that a hole in the data may be represented, a data value of 255 is mapped to 254, and 255 is used to represent a hole.

- **Related Commands:** WAVeform:DATa?,POINts?.
- ***RST Conditions:** Defaults to BYTE.

:FORMat?

WAVeform:FORMat? returns the currently selected output format (BYTE, WORD, COMPressed) for transfer of waveform data. Data is sent to the output buffer.

Example **Query current data format**

```
dimension  statement    String to hold data
WAV:FORM  WORD          Waveform data sent over the bus will
                        be in word format
WAV:FORM?               Query instrument to return selected
                        format
enter  statement        Enter data into computer
```

:POINts?

WAVeform:POINts? returns the points value in the currently selected waveform preamble. The points value is the number of time buckets contained in the waveform selected with the WAVeform:SOURce command. The value is sent to the output buffer.

Example **Query current points value**

```
WAV:POIN?               Query instrument to return points
                        value
enter  statement        Enter data into computer
```

Comments

- **Returned Results:** In most cases the number of time buckets actually acquired will be the number of points set with the ACQuire:POINts command. There are some TIMebase:RANGe settings where the actual number of points will be less than requested, as shown below.

  TIMebase:RANGe to 2 nsec - the number of points actually acquired will be 32, 64, or 100.

  TIMebase:RANGe to 20 nsec - the number of points actually acquired will be 32, 64, 128, or 200.

  TIMebase:RANGe to 50 nsec - the number of points actually acquired will be 32, 64, 128, 256, 500, 512, or 1000.

- **Related Commands:** TIMebase:RANGe, ACQuire:POINts.

:PREamble

WAVeform:PREamble *<data>* is used to send the instrument waveform preamble over the bus and store it in the previously specified waveform memory.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Unit
data	numeric	See below	none

Example

Send the Oscilloscope waveform preamble

For the example, the waveform preamble is "xxxx...."

```
WAV:PRE  xxxx....        Send waveform preamble to
                         Oscilloscope
```

Comments

- **Waveform Preamble Format:** The format of the preamble data is as follows.

  <data> = <format NR1 >,<type NR1>,<points NR1>,<count NR1>, <xincrement NR3>, <xorigin NR3>, <xreference NR3>, <yincrement NR3>, <yorigin NR3>, <yreference NR3>
  Where:

<format> =	1 for BYTE format 2 for WORD format 4 for COMPRESSED format
<type> =	1 for NORMAL type 2 for AVERAGE type 3 for ENVELOPE type
<points > =	See WAVeform:POINts? query.
<count>=	See ACQuire:COUNt? query.
<xincrement>=	See WAVeform:XINCrement? query.
<xorigin>=	See WAVeform:XORigin ? query.
<xreference>=	See WAVeform:XREFerence ? query.
<yincrement>=	See WAVeform:YINCrement ? query.
<yorigin>=	See WAVeform:YORigin ? query.
<yreference> =	See WAVeform:YREFerence ? query.

- **Storing Waveform Preamble:** Only Waveform MEMories (WMEM1-4) may have waveform preamble sent to them. The desired location is selected using the WAVeform:SOURce command.

- **Related Commands:** WAVeform:SOURce.

:PREamble?

WAVeform:PREamble? sends a waveform preamble stored in the instruments previously specified Waveform Memory<*n*>, channel buffer, or Function over the bus.

Example

Send waveform preamble from the oscilloscope over the bus

The following example illustrates the use of the WAVeform:PREamble? query only. Chapter 3 contains an example on performing a complete digitizing operation.

`Dimension statement`	*Dimension string or array*
`WAV:PRE?`	*Send waveform preamble over the bus*
`enter statement`	*Enter data into computer*

Comments

- **Waveform Preamble:** Waveform MEMories (WMEM1-4), channel buffers (CHAN1-4), or FUNCtions (FUNC1-2) may have waveform preamble sent from them. The desired source is selected using the WAVeform:SOURce command.

  Preamble Data: The values set in the preamble are determined when the DIGitize command is executed. The Preamble values are based on the settings of variables in the ACQuire subsystem. Although the preamble values can be changed with a controller, the way the data was acquired cannot be changed. Changing the preamble values cannot change the type of data that was actually acquired, the number of points actually acquired, etc. Therefore, extreme caution must be used when changing any waveform preamble values to ensure the data will still be useful. For example, setting POINts in the preamble to a value different from the actual number of points in the waveform will result in inaccurate data.

- **Waveform Preamble Returned Format:** The returned information can be read into a numeric string or an array. The format of the waveform preamble is explained in the WAVeform:PREamble command.

- **Waveform Data:** The preamble should be read before the waveform data.

- **Related Commands:** WAVeform:SOURce.

:SOURce

WAVeform:SOURce *<source>* is used to select the source used for all the WAVeform subsystem commands.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	CHANnel*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| FUNCtion*n* (*n*=1 to 2)	none

Example **Set waveform source to waveform memory 3**

The following example illustrates the use of the WAVeform:SOURce command only. Chapter 3 contains an example on performing a complete digitizing operation.

```
WAV:SOUR  WMEM3
```
Source to waveform memory 3

Comments

- **Selecting Source:** When the instrument receives information (data or preamble commands) from the bus, source specifies the location where the data will be stored. When the instrument sends information (data or preamble queries) to the bus, source specifies the location where the data currently resides.

  When specifying the source, use the following guidelines:

  ENVelope mode requires Waveform Memory pairs (WMEM1 and 3, or WMEM 2 and 4) to transfer data. Specify WMEMory1 for WMEM1 and 3, or WMEMory2 for WMEM2 and 4. The data is transferred as two arrays. For example, if WMEM1 is specified as the source, the first array is transferred into WMEMory 1 and the second array is transferred into WMEMory 3. The data type is then changed to normal for each of the waveform memories.

  NORMal and AVERage modes are transferred to the selected Waveform Memory (WMEM1, 2, 3, or 4).

- ***RST Condition:** Defaults to CHANnel1.

:SOURce?

WAVeform:SOURce? returns the currently selected source (CHANnel1-4, WMEMory1-4, or FUNCtion1-2) for the waveform subsystem. The data is sent to the output buffer.

Example **Query the current waveform source selection**

```
dimension  statement   String to hold data
WAV:SOUR  WMEM3        Source to waveform memory 3
WAV:SOUR?              Query instrument to return trigger
                       source setting
enter  statement       Enter data into computer
```

:TYPe? WAVeform:TYPe? returns the data type (AVERage, ENVelope, or NORMal) for the previously specified waveform source. The data is sent to the output buffer.

Example **Query acquisition type**

```
dimension statement    String to hold data
ACQ:TYP AVER           Acquisition type to average
WAV:TYP?               Query instrument to return
                       acquisition type setting
enter statement        Enter data into computer
```

Comments

- **Selecting Type:** The type of waveform acquisition is selected by the ACQuire:TYPe command.

  Normal: Normal data consists of the last data point (hit) in each time bucket. This data is transmitted over the bus in a sequential fashion starting with time bucket 0 and going through time bucket *n*-1, where *n* is the number returned by the WAVeform:POINts? query. Time buckets that don't have data in them return –1. Only the magnitude values of each data point are transmitted, the time values correspond to the position in the data array. The first voltage value corresponds to the first time bucket on the left of the active waveform and the last value corresponds to the next to last time bucket on the right of the active waveform.

  Average: Average data consists of the average of the first *n* hits in a time bucket, where *n* is the value returned by the ACQuire:COUNt? query. Time buckets that have fewer than *n* hits return the average of what data they do have. If the ACQuire:COMPlete parameter is set to 100%, then each time bucket must contain the number of data hits specified with the ACQuire:COUNt command. Again, if a time bucket doesn't have any data in it, it will return –1. This data is transmitted over the bus in linear fashion starting with time bucket 0 and proceeding through time bucket *n*-1, where *n* is the number returned by the WAVeform:POINts? query. The first value corresponds to a point at the left of the active waveform and the last value is one point away from the right of the active waveform.

  Envelope: Envelope data consists of two arrays of data, one containing the minimum of the first *n* hits in each time bucket and the other containing the maximum of the first *n* hits in each time bucket, where *n* is the value returned by the ACQuire:COUNt? query. If a time bucket does not have any hits in it, then –1 is returned for both the minimum and maximum values. The two arrays are transmitted one at a time over the bus linearly, starting with time bucket 0 (on the left of the active waveform) and proceeding through time bucket *n*–1, where *n* is the value returned by the WAVeform:POINts? query. The array with the minimum values is sent first. The first value of each array corresponds to the data point on the left of the active waveform. The last value is one data point away from the right of the active waveform.

- **Related Commands:** ACQuire:TYPe, WAVeform:SOURce.

:XINCrement?

WAVeform:XINCrement? returns the time difference between consecutive data points. Query returns x-increment value currently in the preamble (WAVeform:PREamble?) for the current specified source set using the WAVeform:SOURce command. The value is sent to the output buffer.

Example **Query preamble for data point time increments**

`dimension statement`	*String to hold data*
`WAV:XINC?`	*Query instrument to return x-increment value*
`enter statement`	*Enter data into computer*

Comments

- **Data Points:** Current data points are returned using the WAVeform:POINts? query.

  Data Type: Current data type returned using the WAVeform:TYPe? query.

- **Related Commands:** WAVeform:POINts?, TYPe?, PREamble?, SOURce.

:XORigin?

WAVeform:XORigin? returns the time of the first data point in the memory with respect to the trigger point. Query returns x-origin data currently in the preamble (WAVeform:PREamble?) for the current specified source set using the WAVeform:SOURce command. The value is sent to the output buffer.

Example **Query preamble for first data point time**

`dimension statement`	*String to hold data*
`WAV:XOR?`	*Query instrument to return x-origin value*
`enter statement`	*Enter data into computer*

Comments

- **Related Commands:** WAVeform:POINts, PREamble?, SOURce.

:XREFerence?

WAVeform:XREFerence? always returns 0. Represents the specific data point associated with the XORigin data value. Query returns x-reference data currently in the preamble (WAVeform:PREamble?) for the current specified source set using the WAVeform:SOURce command. The value is sent to the output buffer.

Example **Query preamble for reference data point**

`dimension statement`	*String to hold data*
`WAV:XREF?`	*Query instrument to return x-reference value*
`enter statement`	*Enter data into computer*

Comments

- **Related Commands:** WAVeform:XORigin?, PREamble?, SOURce.

:YINCrement?

WAVeform:YINCrement? returns the voltage difference between consecutive data points. Query returns y-increment value currently in the preamble (WAVeform:PREamble?) for the current specified source set using the WAVeform:SOURce command. The value is sent to the output buffer.

Example **Query preamble for data point voltage increments**

`dimension statement`	*String to hold data*
`WAV:YINC?`	*Query instrument to return y-increment value*
`enter statement`	*Enter data into computer*

Comments

- **Data Points:** Current data points are returned using the WAVeform:POINts? query.
- **Related Commands:** WAVeform:POINts?, TYPe?, PREamble?, SOURce.

:YORigin?

WAVeform:YORigin? returns the voltage at center range. Query returns y-origin data currently in the preamble (WAVeform:PREamble?) for the current specified source set using the WAVeform:SOURce command. The value is sent to the output buffer.

Example **Query preamble for center range voltage**

`dimension statement`	*String to hold data*
`WAV:YOR?`	*Query instrument to return y-origin value*
`enter statement`	*Enter data into computer*

Comments

- **Related Commands:** WAVeform:POINts?, PREamble?, SOURce.

:YREFerence?

WAVeform:YREFerence? returns the specific data point associated with the YORigin data value. Query returns y-reference data currently in the preamble (WAVeform:PREamble?) for the current specified source set using the WAVeform:SOURce command. The value is sent to the output buffer.

Example **Query preamble for reference data point**

`dimension statement`	*String to hold data*
`WAV:YREF?`	*Query instrument to return y-reference value*
`enter statement`	*Enter data into computer*

Comments

- **Related Commands:** WAVeform:YORigin?, PREamble?, SOURce.

Command Cross Reference to SCPI Commands

The following table is provided as a quick cross reference of all applicable Hewlett-Packard 54503A Compatible Language (COMP) commands to the similar Standard Commands for Programmable Instruments (SCPI) commands.

COMP Command	SCPI Title	Description
ACQuire		
:COMPlete	[SENSe:]SWEep:POINts:COMPlete	Command operates the same in both languages.
:COMPlete?	[SENSe:]SWEep:POINts:COMPlete?	Command operates the same in both languages.
:COUNt	[SENSe:]AVERage:COUNt	Command operates the same in both languages.
:COUNt?	[SENSe:]AVERage:COUNt?	Command operates the same in both languages.
:POINts	[SENSe:]SWEep:POINts	Command operates the same in both languages.
:POINts?	[SENSe:]SWEep:POINts	Command operates the same in both languages.
:TYPe	[SENSe:]AVERage[:STATe] and [SENSe:]AVERage:TYPE	In SCPI, average is set by state, scalar (normal) and envelope set by turning average state off, then selecting type.
:TYPe?	[SENSe:]AVERage[:STATe]? and [SENSe:]AVERage:TYPE?	See TYPe for explanation.
CALibrate	CALibration	The Compatible CALibrate and the SCPI CALibration subsystems operate the same in both languages.
CHANnel<n>		
:COUPling	[SENSe:]INPut<n>:COUPling and [SENSe:]INPut<n>:IMPedance	In SCPI, coupling and impedance must be selected separately.
:COUPling?	[SENSe:]INPut<n>:COUPling? and [SENSe:]INPut<n>:IMPedance?	In SCPI, coupling and impedance must be queried separately.
:HFReject	[SENSe:]INPut<n>:FILTer[:LPASs][:STATe]	Command operates the same in both languages.
:HFReject?	[SENSe:]INPut<n>:FILTer[:LPASs][:STATe]?	Command operates the same in both languages.
:LFReject	[SENSe:]INPut<n>:FILTer:HPASs[:STATe]	Command operates the same in both languages.
:LFReject?	[SENSe:]INPut<n>:FILTer:HPASs[:STATe]?	Command operates the same in both languages.
:OFFSet	[SENSe:]VOLTage<n>:RANGe:OFFSet	Command operates the same in both languages.
:OFFSet?	[SENSe:]VOLTage<n>:RANGe:OFFSet?	Command operates the same in both languages.
:PROBe	[SENSe:]CORRection<n>:AFACtor	Command operates the same in both languages.
:PROBe?	[SENSe:]CORRection<n>:AFACtor?	Command operates the same in both languages.
:RANGe	[SENSe:]VOLTage<n>:RANGe[:PTPeak]	Command operates the same in both languages.
:RANGe?	[SENSe:]VOLTage<n>:RANGe[:PTPeak]	Command operates the same in both languages.
DISPlay		
:DATa	TRACe[:DATA]	In SCPI, specify pixel memory 1-2 as the destination, and the display data is the source.
:DATa?	TRACe[:DATA]?	In SCPI, specify where the data is located (pixel memory 0-2) as the source.
FUNCtion<n>		
:ADD	CALCulate:MATH<n>[:EXPRession]	Command operates the same in both languages.
:MULTiply	CALCulate:MATH<n>[:EXPRession]	Command operates the same in both languages.
:SUBTract	CALCulate:MATH<n>[:EXPRession]	Command operates the same in both languages.

Command Cross Reference to SCPI Commands —Continued

COMP Command	SCPI Title	Description
MEASure		
:DUTycycle?	MEASure[:SCALar]:VOLTage:DCYCle [reference] [(@)]	Command operates the same in both languages, except source and the reference are specified with the command in SCPI In COMP, the SOURce, UPPer, and LOWer commands are used..
:FALLtime?	MEASure[:SCALar]:VOLTage: FALL:TIME [limits] [(@)] and FTIMe [limits] [(@)]	Commands operate the same in both languages, except source and the upper and lower thresholds are specified with the command in SCPI. In COMP, the SOURce, UPPer, and LOWer commands are used.
:FREQuency?	MEASure[:SCALar]:VOLTage: FREQuency [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:NWIDth?	MEASure[:SCALar]:VOLTage:NWIDth [reference] [(@)]	Command operates the same in both languages, except source and the reference are specified with the command in SCPI In COMP, the SOURce, UPPer, and LOWer commands are used..
:OVERshoot?	MEASure[:SCALar]:VOLTage: FALL:OVERshoot [(@)]and RISE:OVERshoot [(@)]	Commands operate the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:PERiod?	MEASure[:SCALar]:VOLTage:PERiod [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:PREshoot?	MEASure[:SCALar]:VOLTage: FALL:PREShoot [(@)]and RISE:PREShoot [(@)]	Commands operate the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:PWIDth?	MEASure[:SCALar]:VOLTage:PWIDth [reference] [(@)]	Command operates the same in both languages, except source and the reference are specified with the command in SCPI In COMP, the SOURce, UPPer, and LOWer commands are used..
:RISEtime?	MEASure[:SCALar]:VOLTage: RISE:TIME [limits] [(@)] and RTIMe [limits] [(@)]	Commands operate the same in both languages, except source and the upper and lower thresholds are specified with the command in SCPI. In COMP, the SOURce, UPPer, and LOWer commands are used.
:TMAX?	MEASure[:SCALar]:VOLTage: TMAXimum [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:TMIN?	MEASure[:SCALar]:VOLTage: TMINimum [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:VACrms?	MEASure[:SCALar]:VOLTage:AC [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.

Command Cross Reference to SCPI Commands —Continued

COMP Command	SCPI Title	Description
MEASure		
:VAMPlitude?	MEASure[:SCALar]:VOLTage: AMPLitude [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:VAVerage?	MEASure[:SCALar]:VOLTage:[DC] [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:VBASe?	MEASure[:SCALar]:VOLTage:LOW [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:VMAX?	MEASure[:SCALar]:VOLTage:MAXimum [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:VMIN?	MEASure[:SCALar]:VOLTage:MINimum [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:VRMS?	MEASure[:SCALar]:VOLTage:AC [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
:VTOP?	MEASure[:SCALar]:VOLTage:HIGH [(@)]	Command operates the same in both languages, except source is specified with the command in SCPI, and using the SOURce command in COMP.
OUTPut	OUTPut	The Compatible SCPI OUTPut subsystems operate the same in both languages.
Root Commands		
AUToscale	SYSTem:AUToscale	Command operates the same in both languages.
BLANk	[SENSe:]INPut<n>[:STATe] OFF	Compatible command allows channels, functions, and memories to be turned off. SCPI only allows inputs to be turned off.
DIGitize	INITiate[:IMMediate]	Command operates the same in both languages.
ERASe	TRACe[:DATA]	In SCPI, specify pixel memory being erased as the destination, and "0" as the source.
MERGe	TRACE[:DATA]	In SCPI, specify pixel memory being merged as the destination, and "PMEM0" as the source.
RUN	INITiate:CONTinuous ON	Command operates the same in both languages.
SERial	SYSTem:SERial	Command operates the same in both languages.
STATus?	[SENSe:]INPut<n>[:STATe]?	Compatible command allows channels, functions, and memories to be turned queried. SCPI only allows inputs to be queried.
STOP	ABORt	Command operates the same in both languages.
STORe	TRACe[:DATA]	Command operates the same in both languages.
TER?	STATus:OPERation[:EVENt]?	In SCPI, bit 8 is triggered bit.
VIEW	[SENSe:]INPut<n>[:STATe] ON	Compatible command allows channels, functions, and memories to be turned on. SCPI only allows inputs to be turned on.

Command Cross Reference to SCPI Commands —Continued

COMP Command	SCPI Title	Description
SUMMary	STATus	The Compatible SUMMary and the SCPI STATus subsystems operate the same in both languages.
SYSTem		
:ERRor?	SYSTem:ERRor?	Compatible command allows message to be blanked. SCPI always returns number and message.
:LANGuage	SYSTem:LANGuage	Command operates the same in both languages.
:LANGuage?	SYSTem:LANGuage?	Command operates the same in both languages.
:SETup	SYSTem:SET	Command operates the same in both languages.
:SETup?	SYSTem:SET?	Command operates the same in both languages.
TEST	TEST	The Compatible SCPI TEST subsystems operate the same in both languages.
TIMebase		
:DELay	[SENSe:]SWEep:TIME:DELay	Command operates the same in both languages.
:DELay?	[SENSe:]SWEep:TIME:DELay?	Command operates the same in both languages.
:RANGe	[SENSe:]SWEep:TIME:RANGe	Command operates the same in both languages.
:RANGe?	[SENSe:]SWEep:TIME:RANGe?	Command operates the same in both languages.
:REFerence	[SENSe:]SWEep:TIME:DELay:LINK	Command operates the same in both languages.
:REFerence?	[SENSe:]SWEep:TIME:DELay:LINK?	Command operates the same in both languages.
TRIGger		
:HOLDoff	TRIGger:ECOunt	In SCPI, holdoff is defined by events only (not time).
:HOLDoff?	TRIGger:ECOunt?	In SCPI, holdoff is defined by events only (not time).
:LEVel	TRIGger:LEVel	Command operates the same in both languages.
:LEVel?	TRIGger:LEVel?	Command operates the same in both languages.
:SENSitivity	TRIGger:HYSTeresis	Command operates the same in both languages.
:SENSitivity?	TRIGger:HYSTeresis?	Command operates the same in both languages.
:SLOPe	TRIGger:SLOPe	Command operates the same in both languages.
:SLOPe?	TRIGger:SLOPe?	Command operates the same in both languages.
:SOURce	TRIGger:SOURce	Command operates the same in both languages.
:SOURce?	TRIGger:SOURce?	Command operates the same in both languages.
WAVeform		
DATa	TRACe[:DATA]	In SCPI, the waveform memory (1-4) is specified as the destination, and the waveform data is the source.
DATa?	TRACe[:DATA]?	In SCPI, the input1-4, math1-2, or waveform memory (1-4) is specified as the source.
FORMat	FORMat[:DATA] INTeger,[8|16]	In SCPI, INTeger 8 is byte and 16 is returned for word.
FORMat?	FORMat[:DATA]?	In SCPI, 8 is returned for byte and 16 is word. Compressed is not available in SCPI.
POINts?	TRACe:POINts?	Command operates the same in both languages.
PREamble	TRACe:PREamble	Command operates the same in both languages.
PREamble?	TRACe:PREamble?	Command operates the same in both languages.

Command Cross Reference to HP 54503A Commands

The following table is provided as a quick cross reference of all Hewlett-Packard 54503A commands that are not supported in the HP E1426A COMPatible programming language. Any command not listed is fully functional on both the HP 54503A and the HP E1426A.

HP 54503A Command	COMP Command	Description
Common Commands		
*IST?	None	Not supported by COMP.
*OPT?	None	Not supported by COMP.
*PRE	None	Not supported by COMP.
*PRE?	None	Not supported by COMP.
Root Commands		
BEEPer	None	Not supported by COMP.
BEEPer?	None	Not supported by COMP.
EOI	None	Not supported by COMP.
EOI?	None	Not supported by COMP.
LER?	None	Not supported by COMP.
MENu	None	Not supported by COMP.
MENu?	None	Not supported by COMP.
PLOt?	None	Not supported by COMP.
PRINt?	None	Not supported by COMP.
SYSTem		
DSP	None	Not supported by COMP.
DSP?	None	Not supported by COMP.
KEY	None	Not supported by COMP.
KEY?	None	Not supported by COMP.
DISPlay		
COLumn	None	Not supported by COMP.
COLumn?	None	Not supported by COMP.
CONNect	None	Not supported by COMP.
CONNect?	None	Not supported by COMP.
FORMat	None	Not supported by COMP.
FORMat?	None	Not supported by COMP.
GRATicule	None	Not supported by COMP.
GRATicule?	None	Not supported by COMP.
INVerse	None	Not supported by COMP.
INVerse?	None	Not supported by COMP.
LINe	None	Not supported by COMP.
MASK	None	Not supported by COMP.
MASK?	None	Not supported by COMP.
PERSistence	PERSistence	Compatible only supports INFinite and SINGle.
ROW	None	Not supported by COMP.
ROW?	None	Not supported by COMP.
SCReen	None	Not supported by COMP.
SCReen?	None	Not supported by COMP.
STRing	None	Not supported by COMP.

Command Cross Reference to HP 54503A Commands—Continued

Any command not listed is fully functional on both the HP 54503A and the HP E1426A.

HP 54503A Command	COMP Command	Description
DISPlay		
TEXt	None	Not supported by COMP.
TMARker	None	Not supported by COMP.
TMARker?	None	Not supported by COMP.
VMARker	None	Not supported by COMP.
VMARker?	None	Not supported by COMP.
FUNCtion		
VERSus	None	Not supported by COMP.
HARDcopy		
LENGth	None	Not supported by COMP.
LENGth?	None	Not supported by COMP.
PAGe	None	Not supported by COMP.
PAGe?	None	Not supported by COMP.
TRIGger		
SOURce	SOURce	COMP supports TTLTrg0-7 and ECLTrg0-2 backplane trigger bus lines.
WAVeform		
DATa	DATa	COMP does not support ASCII data.
FORMat	FORMat	COMP does not support ASCII format.
SOURce	SOURce	COMP supports FUNCtion1-2.

Common Command Reference

The following table lists the IEEE 488.2 Common (*) Commands that can be executed by the Oscilloscope module. The operation of some of these commands is described in Chapter 3 of this manual. For more information on Common Commands, refer to the HP 54503A Programming Guide or the ANSI/IEEE Standard 488.2-1987.

Common Command Reference

Command	Title	Description
*CLS	Clear status register	Clears all Event Registers, the Request for OPC flag, and all Queues (except output queue).
*ESE <mask> *ESE?	Event status enable Event status enable query	Used to set the bits in the Event Status Enable Register. Queries the current contents in the Event Status Enable Register.
*ESR?	Event status register query	Queries and clears contents in the Standard Event Status Register.
*IDN?	Identification query	Returns identification string of the Oscilloscope.
*LRN?	Learn query	Returns a string that contains the current Oscilloscope setup.
*OPC *OPC?	Operation complete Operation complete query	Sets the Request for OPC flag when all pending operations have completed. Also sets OPC bit in the Event Status register. Returns a 1 to the output queue when all pending operations have been completed.
*RCL <n>	Recall saved state	Recalls previously stored Oscilloscope Module configuration. <n> (1 to 4) is the location in memory where the desired (previously stored) set-up is located.
*RST	Reset	Resets the Oscilloscope hardware and firmware to a specific state (refer to Table 3-4 for a list of reset conditions).
*SAV <n>	Save state	Stores the present Oscilloscope Module configuration in memory. Stores all Oscilloscope and Measurement Set-up parameters. <n> (1 to 4) is the location in memory where the current set-up is to be stored.
*SRE <mask> *SRE?	Service request enable Service request enable query	Used to set the Service Request Enable Register bits to generate a service request. Queries the current contents in the Service Request Enable Register.
*STB?	Read status byte query	Queries the current contents in the Status Byte Register.
*TRG?	Trigger	Used to generate a trigger event.
*TST?	Self-Test query	Returns 0 unless self test fails. Disconnect all inputs prior to performing self tests.
*WAI	Wait to Continue	Halts execution of commands and queries until the No Operation Pending message is true.

Command Quick Reference

The following tables summarize Hewlett-Packard 54503A Compatible Language (COMP) commands and IEEE 488.2 Common (*) commands for the HP E1426A Oscilloscope module.

Subsystem	Command/Parameter	Description
ACQuire	:COMPlete *<complete>*	Enters the completion criteria for data acquisition.
	:COMPlete?	Returns the current acquisition complete value.
	:COUNt *<count>*	Enters the count for average data acquisition mode.
	:COUNt?	Returns the current acquisition count value.
	:POINts *<points>*	Selects the number of time buckets for data acquisition.
	:POINts?	Returns the current points value.
	:TYPe *<mode>*	Selects the type of acquisition that will occur (average, normal, or envelope).
	:TYPe?	Returns the acquisition type currently selected.
CALibrate	:PCALibration	Probe calibration related commands.
	:ATTenuation	Probe attenuation calibration related commands.
	:BCALibration	Performs an attenuation calibration.
	:CHANnel*<number>*	Selects the channel (1-4) for the attenuation calibration.
	:TNULl:CH1TO*<number>* ,*<time>*	Enters a time null value for a specified channel pair (1-2, 1-3, 1-4).
	:REPort? *<channel>*	Returns a report of calibration results for the channel specified.
	:SCALibration	Self calibration related commands.
	:BCALibration	Performs a configured calibration, or loads default data.
	:DCALibration	Reads the default calibration data.
	:DELay *<channel>*	Configures for a delay calibration on the channel specified (1-4).
	:DOUTput *<level>*	Sets the DC Calibrator output to 0 volts or 5 volts.
	:LTCalibration	Configures for a logic trigger calibration on channel 1.
	:TNULl *<channel_skew>*	Configures for a time null calibration on the channel pair specified (1-2, 1-3, 1-4).
	:VERTical	Configures for a vertical calibration on all channels.
	:TNULl *<value1>,<value2>,<value3>*	Enters time null values for channel pairs 1-2, 1-3, and 1-4.
	:TNULl?	Returns the current time null values for channel pairs 1-2, 1-3, and 1-4.
CHANnel*<number>*	:COUPling *<type>*	Selects the specified channel's (1-4) coupling and impedance (AC 1MΩ, DC 1MΩ, or DC 50Ω).
	:COUPling?	Returns a specific channel's (1-4) current coupling selection.
	:ECL	Configures the specified channel's (1-4) vertical controls for an ECL measurement.
	:HFReject *<mode>*	Enables or disables the specified channel's (1-4) low-pass filter.
	:HFReject?	Returns a specific channel's (1-4) current low-pass filter state.
	:LFReject *<mode>*	Enables or disables the specified channel's (1-4) high-pass filter.
	:LFReject?	Returns a specific channel's (1-4) current high-pass filter state.
	:OFFSet *<value>*	Enters the specified channel's (1-4) offset.
	:OFFSet?	Returns a specific channel's (1-4) current offset value.
	:PROBe *<atten>*	Enters the specified channel's (1-4) probe attenuation factor.
	:PROBe?	Returns a specific channel's (1-4) current probe attenuation factor.
	:RANGe *<range>*	Enters the specified channel's (1-4) full scale vertical range.
	:RANGe?	Returns a specific channel's (1-4) current full scale vertical range value.
	:TTL	Configures the specified channel's (1-4) vertical controls for a TTL measurement.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
DISPlay	:DATa *<block>*	Write a binary block of waveform data to pixel memory 1 or 2.
	:DATa?	Reads a binary block of waveform data from pixel memory 0, 1 or 2.
	:PERSistence *<period>*	Selects persistence of infinite or single for pixel memory 0.
	:PERSistence?	Returns the persistence currently selected.
	:SOURce *<source>*	Selects the source (pixel memory 0-2) for the DATa command and DATa? query.
	:SOURce?	Returns the current display source selection.
FUNCtion*<number>*	:ADD *<source>*,*<source>*	Adds two specified sources and retains the result in the function number specified.
	:INVert *<source>*	Inverts the specified source and retains the result in the function number specified.
	:MULTiply *<source>*,*<source>*	Multiplies two specified sources and retains the result in the function number specified.
	:OFFSet *<value>*	Enters the specified function's (1-2) offset.
	:OFFSet?	Returns the specified function's (1-2) current offset value.
	:ONLY *<source>*	Duplicates the specified source and retains the result in the function number specified.
	:RANGe *<range>*	Enters the specified function's (1-2) full scale vertical range.
	:RANGe?	Returns a specific function's (1-2) current range value.
	:SUBTract *<source>*,*<source>*	Subtracts two specified sources and retains the result in the function number specified.
MEASure	:ALL?	Measures all parameters and returns results.
	:COMPare *<measurement>*, *<upper_limit>*,*<lower_limit>*	Selects the measurement and limits for a limit test.
	:COMPare?	Returns the current compare selection.
	:CURSor? *<type>*	Returns time and voltage values of a specific marker (delta, start, or stop).
	:DEFine *<measure_spec>*	Enters user defined definitions and thresholds.
	:DEFine? *<measure_spec>*	Returns the user defined definitions and thresholds currently selected.
	:DELay	Starts a continuous delay measurement. Results are not returned.
	:DELay?	Measures delay and returns results.
	:DESTination *<location>*	Selects the destination (memory) for a limit test violation.
	:DESTination?	Returns the destination currently selected.
	:DUTycycle	Starts a continuous duty cycle measurement. Results are not returned.
	:DUTycycle?	Measures duty cycle and returns results.
	:ESTArt *<edge>*	Positions the edge start marker on a specific rising or falling edge.
	:ESTArt?	Returns the current start edge marker position.
	:ESTOp *<edge>*	Positions the edge stop marker on a specific rising or falling edge.
	:ESTOp?	Returns the current start edge marker position.
	:FALLtime	Starts a continuous fall time measurement. Results are not returned.
	:FALLtime?	Measures fall time and returns results.
	:FREQuency	Starts a continuous frequency measurement. Results are not returned.
	:FREQuency?	Measures frequency and returns results.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
MEASure	:LIMittest *<mode>*	Turns the limit test on or off.
	:LOWer *<value>*	Enters a user defined lower threshold level.
	:LOWer?	Returns the current lower threshold level.
	:MODe *<mode>*	Selects whether measurements are made using standard IEEE or user defined parameters.
	:MODe?	Returns measurement mode currently selected.
	:NWIDth	Starts a continuous negative pulse width measurement. Results are not returned.
	:NWIDth?	Measures negative pulse width and returns results.
	:OVERshoot	Starts a continuous overshoot measurement. Results are not returned.
	:OVERshoot?	Measures negative pulse width and returns results.
	:PERiod	Starts a continuous period measurement. Results are not returned.
	:PERiod?	Measures period and returns results.
	:POSTfailure *<mode>*	Selects what occurs after a limit test violation (continue or stop).
	:POSTfailure?	Returns the current post-failure selection.
	:PRECision *<coarse>*	Not used in the Oscilloscope.
	:PRECision?	Always returns COARse.
	:PREShoot	Starts a continuous preshoot measurement. Results are not returned.
	:PREShoot?	Measures preshoot and returns results.
	:PWIDth	Starts a continuous positive pulse width measurement. Results are not returned.
	:PWIDth?	Measures positive pulse width and returns results.
	:RESults?	Returns the contents of the measurement queue (up to eight active measurement results).
	:RISetime	Starts a continuous rise time measurement. Results are not returned.
	:RISetime?	Measures rise time and returns results.
	:SCRatch	Clears the measurement queue of all measurement results.
	:SOURce *<source>*[,*<source>*]	Selects the source for all MEASure commands and queries. Second source used only for delay measurements.
	:SOURce?	Return the source(s) currently selected.
	:STATistics *<mode>*	Enables or disables the statistics mode (minimum, maximum, average, and current measurement results provided).
	:STATistics?	Returns the current statistics mode state.
	:TDELta?	Returns the time difference between time start/stop markers.
	:TMAX?	Returns the time that the maximum voltage occurred (referenced to trigger).
	:TMIN?	Returns the time that the minimum voltage occurred (referenced to trigger time).
	:TSTArt *<time>*	Positions the time start marker to a specific time (referenced to trigger time).
	:TSTArt?	Returns the current time start marker value.
	:TSTOp *<time>*	Positions the time stop marker to a specific time (referenced to trigger time).
	:TSTOp?	Returns the current time stop marker value.
	:TVOLt? *<voltage>*, *<slope><occurrence>*	Returns the time interval between the trigger and a defined occurrence.
	:UNITs *<unit>*	Selects the threshold units (volts or percent).
	:UNITs?	Returns the units currently selected.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
MEASure	:UPPer *<value>*	Enters a user defined upper threshold level.
	:UPPer?	Returns the current upper threshold level.
	:VACRms	Starts a continuous AC RMS voltage measurement. Results are not returned.
	:VACRms?	Measures AC RMS voltage and returns results.
	:VAMPlitude	Starts a continuous amplitude voltage measurement. Results are not returned.
	:VAMPlitude?	Measures amplitude voltage and returns results.
	:VAVerage	Starts a continuous average voltage measurement. Results are not returned.
	:VAVerage?	Measures average voltage and returns results.
	:VBASe	Starts a continuous base voltage measurement. Results are not returned.
	:VBASe?	Measures base voltage and returns results.
	:VDCRms	Starts a continuous DC RMS voltage measurement. Results are not returned.
	:VDCRms?	Measures DC RMS voltage and returns results.
	:VDELta?	Returns the voltage difference between the voltage start/stop markers.
	:VFIFty	Positions the voltage markers at the 50% voltage point.
	:VMAX	Starts a continuous maximum voltage measurement. Results are not returned.
	:VMAX?	Measures maximum voltage and returns results.
	:VMIN	Starts a continuous minimum voltage measurement. Results are not returned.
	:VMIN?	Measures minimum voltage and returns results.
	:VPP	Starts a continuous peak-to-peak voltage measurement. Results are not returned.
	:VPP?	Measures peak-to-peak voltage and returns results.
	:VRELative *<percent>*	Moves the voltage markers to a specified percentage point from their last established position.
	:VRELative?	Returns the current relative voltage stop marker position.
	:VRMS	Starts a continuous RMS voltage measurement. Results are not returned.
	:VRMS?	Measures RMS voltage and returns results.
	:VSTArt *<voltage>*	Positions the voltage start marker to a specific voltage (referenced to 0 volts).
	:VSTArt?	Returns the current voltage start marker value.
	:VSTOp *<voltage>*	Positions the voltage stop marker to a specific voltage (referenced to 0 volts).
	:VSTOp?	Returns the current voltage stop marker value.
	:VTIMe? *<time>*	Returns the voltage level at a specified time (referenced to the trigger).
	:VTOP	Starts a continuous top voltage measurement. Results are not returned.
	:VTOP?	Measures top voltage and returns results.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
MEMory	:VME	VME Memory allocation for acquisition and measurement data.
	:ADDRess *<address>*	Selects an A24 memory address space for acquisition data.
	:ADDRess? [MIN|MAX]	Returns the present A24 memory address space if parameter is blank. Returns MINimum, or MAXimum address available if selected.
	:MEASure	Commands for measurement data.
	:ADDRess *<address>*	Selects an A24 memory address space for measurement data.
	:ADDRess? [MIN|MAX]	Returns the present A24 memory address space if parameter is blank. Returns MINimum, or MAXimum address available if selected.
	:STATe *<mode>*	Enables or disables VME memory space for measurement data.
	:STATe?	Returns the VME memory measurement state.
	:SIZE *<bytes>*	Selects the size of the external VME memory card.
	:SIZE?	Returns the current size of the external VME memory card.
	:STATe *<mode>*	Enables or disables the VME memory subsystem (for acquisition and measurement data).
	:STATe?	Returns the current VME memory subsystem state.
OUTPut	:ECLTrg*<number>*	Selects ECL trigger bus lines 0-2.
	[:STATe] *<mode>*	Enables or disables the selected ECL trigger line.
	[:STATe]?	Returns the current ECL trigger state for the line selected.
	:EXTernal	Selects the TTL Trigger Output BNC connector on the front panel.
	[:STATe] *<mode>*	Enables or disables the TTL Trigger Output connector.
	[:STATe]?	Returns the current TTL Trigger Output connector state.
	[:STATe]	Enables or disables the entire output subsystem. Must be enabled for any of the outputs to function.
	[:STATe]?	Returns the current output subsystem state.
	:TTLTrg*<number>*	Selects TTL trigger bus lines 0-7.
	[:STATe] *<mode>*	Enables or disables the selected TTL trigger line.
	[:STATe]?	Returns the current TTL state for the line selected.
ROOT Commands	AUToscale	Evaluates all inputs, then sets conditions to present the signals.
	BLANk *<source>*[,*<source>*[,*<source>*[,*<source>*]]]	Turns off the specified source (channels, pixel memories, waveform memories, and functions).
	BNC *<output>*	Selects the signal present (probe or trigger) at the Probe Compensation AC Calibrator output connector.
	BNC?	Returns the current selection of the Probe Compensation AC Calibrator output connector.
	DIGitize *<source>*[,*<source>*[,*<source>*[,*<source>*]]]	Digitizes waveform data on the selected channel(s) (1-4).
	ERASe *<source>*	Erases waveform data in pixel memory (0-2).
	LTER?	Returns the limit test event register value. 1=violation.
	MERGe *<location>*	Merges the contents of pixel memory 0 with the contents of pixel memory 1 or 2 (as specified).
	RUN	Starts acquiring data for the active waveform.
	RUN?	Returns current acquisition state.
	SERial *<string>*	Enters the Oscilloscope serial number.
	STATus? *<source>*	Returns whether the specified source (channel, pixel memory, waveform memory, or function) is on or off.
	STOP	Stops acquiring data for the active waveform.
	STORe *<source>*,*<destination>*	Moves a previously stored , channel, or function waveform to a specified waveform memory location.
	TER?	Returns the trigger event register value. 1=triggered.
	VIEW *<source>*[,*<source>*[,*<source>*[,*<source>*]]]	Turns on the specified source (channels, pixel memories, waveform memories, and functions).

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
SUMMary	:PRESet	Sets all the questionable enable registers to "1's".
	:QUEStionable	Reports the calibration and self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:CALibration	Reports the channel 1-4, default, and probe attenuation calibration values and conditions.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:CHANnel<*number*>	Reports the specified channel's (1-4) current calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:AD	Reports the specified channel's (1-4) A/D calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:DELay	Reports the specified channel's (1-4) delay calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:GAIN	Reports the specified channel's (1-4) gain calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:HYSTeresis	Reports the specified channel's (1-4) hysteresis calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:LTRigger	Reports the channel 1 logic trigger calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
SUMMary	:QUEStionable	
	:CALibration	
	:CHANnel*<number>*	
	:OFFSet	Reports the specified channel's (1-4) offset calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:TNULl	Reports the specified channel's (2-4) time null calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:TRIGger	Reports the specified channel's (1-4) trigger calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:DCALibration	Reports default calibration load status.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:PROBe	Reports the probe calibration attenuation results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:TEST	Reports diagnostic or self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:ACQuisition	Reports the acquisition self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:AD	Reports the acquisition A/D self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
SUMMary	:QUEStionable	
	:TEST	
	:ACQuisition	
	:ATRigger	Reports the acquisition analog trigger self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:DA	Reports the acquisition D/A self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:LTRigger	Reports the acquisition logic trigger self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:TIMebase	Reports the acquisition time base self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:INTerpolator	Reports the acquisition time base interpolator self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:RAM	Reports the random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:ACQuisition	Reports the acquisition random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:DISPlay	Reports the display random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
SUMMary	:QUEStionable	
	:TEST	
	:RAM	
	:NVOLatile	Reports the non-volatile random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:SYSTem	Reports the system random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:ROM	Reports the read only memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:NPRotect	Reports the non-protected random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:SYSTem	Reports the system read only memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
SYSTem	:ERRor? [<*mode*>]	Returns system error number and, if selected, an error message.
	:HEADer <*mode*>	Enables or disables the command header returned with the measurement results.
	:HEADer?	Returns the current command header state.
	:LANGuage <*command*>	Selects the Oscilloscope programming language (COMP or SCPI).
	:LANGuage?	Returns the programming language currently selected.
	:LONGform <*mode*>	Enables or disables the command header format (long form or short form).
	:LONGform?	Returns the current long form state.
	:SETup <*setup*>	Sends a previously saved learn string to the Oscilloscope.
	:SETup?	Returns the learn string (contains Oscilloscope setup information).

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
TEST	:ACQ [<*test*>]	Performs an internal acquisition self test. All five tests (AD, analog trigger, logic trigger,time base, and DA), are performed unless an individual test is specified.
	:RAM [<*test*>]	Performs an internal random access memory self test. All four tests (display, acquisition, system, and non-volatile) are performed unless an individual test is specified.
	:ROM [<*test*>]	Performs an internal read only memory self test. Both tests (system ROM and non-volatile protected RAM) are performed unless an individual test is specified.
	:TALL	Performs all internal self tests (ACQ, RAM, and ROM).
TIMebase	:DELay <*time*>	Enters the desired time between trigger and delay reference point.
	:DELay?	Returns current time base delay value.
	:MODe <*mode*>	Selects time base mode (auto, triggered, single).
	:MODe?	Returns time base mode currently selected.
	:RANGe <*range*>	Enters full scale horizontal range.
	:RANGe?	Returns current full scale horizontal range value.
	:REFerence <*position*>	Sets the delay reference point to right, left, or center of the active waveform.
	:REFerence?	Returns the delay reference point currently selected.
	:WINDow <*mode*>	Enables or disables the expanded time base mode.
	:WINDow?	Returns expanded time base mode state.
	:DELay <*time*>	Enters expanded time base delay relative to the main sweep delay and reference setting.
	:DELay?	Returns the current expanded time base delay value.
	:RANGe <*range*>	Enters full scale horizontal range for the expanded time base.
	:RANGe?	Returns the current expanded time base range value.
TRIGger	:CENTered	Sets the trigger level to the center of the range.
	:CONDition <*argument*>	Defines a set of conditions that must exist before a trigger event.
	:CONDition?	Return the set of conditions currently selected.
	:DELay <*mode*>	Selects the delay type (time or events) and delay value (in seconds or counts) . When events are selected, delay slope and source must be defined.
	:DELay?	Returns the current delay type and value.
	:SLOPe <*polarity*>	Selects the edge (rising or falling) that will be counted when delay is set to events.
	:SLOPe?	Returns the delay slope currently selected.
	:SOURce <*source*>	Selects the source(channel 1-4) that will be counted when delay is set to events.
	:SOURce?	Returns the delay source currently selected.
	:FIELd <*number*>	Selects a field (1 or 2) for TV Trigger mode.
	:FIELd?	Returns the field value currently selected.
	:HOLDoff <*holdoff*>	Selects the holdoff type (time or events) and value (in seconds or counts) for the holdoff.
	:HOLDoff?	Returns the current holdoff type and value.
	:LEVel <*level*>	Enters the trigger level.
	:LEVel?	Return the current trigger level value.
	:LINe <*number*>	Selects the line the trigger will be generated on in TV Trigger mode.
	:LINe?	Returns the current line value.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
TRIGger	:LOGic *<level>*	Selects the "trigger on" logic level (high, low, don't care) for an enabled path. Current trigger level is used as the reference
	:LOGic?	Returns the logic level currently selected.
	:MODe *<mode>*	Selects the trigger mode (edge, pattern, state, delay, or TV).
	:MODe?	Returns the trigger mode currently selected.
	:OCCurrence *<number>*	Enters the number of trigger events that must occur before a sweep.
	:OCCurrence?	Returns the current occurrence value.
	:SLOPe *<polarity>*	Selects the edge (rising or falling) that will be counted by the occurrence command.
	:SLOPe?	Returns the occurrence slope currently selected.
	:SOURce *<source>*	Selects the source(channel 1-4) that will be counted by the occurrence command.
	:SOURce?	Returns the occurrence source currently selected.
	:PATH *<channel>*	Selects the path (channel 1-4) for logic commands.
	:PATH?	Returns the currently selected trigger source.
	:POLarity *<polarity>*	Selects the edge (rising or falling) for the trigger in TV mode.
	:POLarity?	Returns the trigger edge currently selected for TV mode.
	:QUALify *<mode>*	Selects the mode to qualify the trigger to before a delay is defined.
	:QUALify?	Returns the qualify mode currently selected.
	:SENSitivity *<mode>*	Enables or disables the noise reject. Normal is off, low is on.
	:SENSitivity?	Returns the current sensitivity (noise reject) state.
	:SLOPe *<polarity>*	Selects the edge (rising or falling) for the trigger.
	:SLOPe?	Returns the trigger edge currently selected.
	:SOURce *<source>*	Selects the source that will produce the trigger (channel 1-4, TTLTrg 0-7, ECLTrg 0-2).
	:SOURce?	Returns the trigger source currently selected.
	:STANdard *<standard>*	Selects the signal standard for TV mode (525, 625, or user defined).
	:STANdard?	Returns the signal standard selected currently selected.
WAVeform	:COUNt?	Always returns 1.
		Writes a binary block of waveform data to pixel memory 1 or 2.
	:DATa?	Reads a binary block of waveform data from pixel memory 0, 1 or 2.
	:DATa *<data>*	Reads a binary block of waveform data from channel 1-4, waveform memory 1-4, or function 1-2.
	:DATa?	Receives waveform block data from the Oscilloscope.
	:FORMat *<format>*	Selects waveform data format (word, byte, compressed).
	:FORMat?	Returns the format currently selected.
	:POINts?	Returns the data points value currently selected in the preamble.
	:PREamble *<data>*	Sends preamble data to the Oscilloscope.
	:PREamble?	Receives preamble data from the Oscilloscope.
	:SOURce *<source>*	Selects the source (or destination) for all WAVeform subsystem commands.
	:SOURce?	Returns waveform source currently selected.
	:TYPe?	Returns data acquisition mode (normal, average, or envelope).
	:XINCrement?	Returns the time difference between data points.
	:XORigin?	Returns the time of the first data point.
	:XREFerence?	Always returns 0.
	:YINCrement?	Returns the voltage difference between data points.
	:YORigin?	Returns the voltage at the center of the waveform.
	:YREFerence?	Returns the specific data point at y-origin.

IEEE 488.2 Common Commands Quick Reference

Command	Title	Description
*CLS	Clear status register	Clears all Event Registers, the Request for OPC flag, and all Queues (except output queue).
*ESE <mask> *ESE?	Event status enable Event status enable query	Used to set the bits in the Event Status Enable Register. Queries the current contents in the Event Status Enable Register.
*ESR?	Event status register query	Queries and clears contents in the Standard Event Status Register.
*IDN?	Identification query	Returns identification string of the Oscilloscope.
*LRN?	Learn query	Returns a string that contains the current Oscilloscope setup.
*OPC *OPC?	Operation complete Operation complete query	Sets the Request for OPC flag when all pending operations have completed. Also sets OPC bit in the Event Status register. Returns a 1 to the output queue when all pending operations have been completed.
*RCL <n>	Recall saved state	Recalls previously stored Oscilloscope Module configuration. <n> (1 to 4) is the location in memory where the desired (previously stored) set-up is located.
*RST	Reset	Resets the Oscilloscope hardware and firmware to a specific state (refer to Table 3-4 for a list of reset conditions).
*SAV <n>	Save state	Stores the present Oscilloscope Module configuration in memory. Stores all Oscilloscope and Measurement Set-up parameters. <n> (1 to 4) is the location in memory where the current set-up is to be stored.
*SRE <mask> *SRE?	Service request enable Service request enable query	Used to set the Service Request Enable Register bits to generate a service request. Queries the current contents in the Service Request Enable Register.
*STB?	Read status byte query	Queries the current contents in the Status Byte Register.
*TRG?	Trigger	Used to generate a trigger event.
*TST?	Self-Test query	Returns 0 unless self test fails. Disconnect all inputs prior to performing self tests.
*WAI	Wait to Continue	Halts execution of commands and queries until the No Operation Pending message is true.

Contents

Chapter 5 — Using the Oscilloscope with SCPI

5

Using the Oscilloscope with SCPI

Using This Chapter

This chapter uses typical examples to show how to use the Oscilloscope module using SCPI commands. See Chapter 3 for instructions on using COMP commands. This chapter contains the following sections:

Oscilloscope SCPI Commands

Table 5-1. Oscilloscope SCPI Commands Used In Chapter 5

Command	Description
*CLS	Clear status.
*RST	Reset the Oscilloscope to its default state.
[SENSe:]	
:AVERage	
[:STATe]	Set the acquisition type to average.
:COUNt	Select the number of averages for average mode.
:CORRection<n>	<n> is the input number 1-4.
:AFACtor	Select the input probe attenuation factor.
:INPut<n>[:STATe]	Enable or disable inputs.<n> is the input number 1-4.
:SWEep	
:POINts	Specify the number of data points for data acquisition.
:COMPlete	Specify the data acquisition completion criteria.
:TIME	
:RANGe	Specify the full scale horizontal range for the main sweep.
:VOLT<n>	<n> is the input number 1-4.
:RANGe	Set the full scale vertical range.
CALibration	
:SCALibration	Self Calibration routines.
:BCALibration	Begin a configured calibration, or load default data.
:DCALibration	Configure for a default calibration routine.
:DELay	Configure for a delay calibration routine.
:LTCalibration	Configure for a logic trigger calibration routine.
:TNULl	Configure for a time null calibration routine.
:VERTical	Configure for a vertical calibration routine.

Table 5-1. Oscilloscope SCPI Commands Used In Chapter 5 —Continued

Command	Description
CONFigure	
[:SCALar]	
:VOLTage	
:FREQuency [@yyn)]	Configure for a frequency measurement. Results are NOT returned. yyn is the measurement source.
:FTIMe [xxx] [(@yyn)]	Configure for a fall time measurement. Results are NOT returned. xxx is the upper and lower thresholds. yyn is the measurement source. If xxx blank, uses default thresholds (10/90%).
:RTIMe [xxx] [(@yyn)]	Configure for a rise time measurement. Results are NOT returned. xxx is the upper and lower thresholds. yyn is the measurement source. If xxx blank, uses default thresholds (10/90%).
FORMat	
[:DATA] INT,*n*	Selects the waveform data format, where n is 8 or 16 bit.
INITiate	
[:IMMediate]	Digitize waveform data.
:CONTinuous	Enable data acquisition for each trigger event.
MEASure	
[:SCALar]	
:VOLTage	
:AC? [@yyn)]	Measure input signal AC RMS voltage and read the measurement results. yyn is the measurement source.
:PERiod? [@yyn)]	Measure input signal period and read the measurement results. yyn is the measurement source.
READ	
[SCALar:]	
:VOLTage[:xxx]?	Perform a configured measurement, and return measurement results. xxx is the specific measurement. If xxx blank, reads a DC voltage measurement.If READ? executed, will read the last configured measurement.
STATus	
:PRESet	Preset the questionable enable registers.
:QUEStionable	
:CALibration?	Read the Calibration register.
SYSTem	
:AUToscale	Perform an Autoscale.
TRACe	
[:DATA]? yyn	Read waveform data from the Oscilloscope. yyn is the current location of the data.
:PREamble? yyn	Read preamble data from the Oscilloscope. yyn is the current location of the data.
TRIGger	
:LEVel	Specify the trigger level.
:SLOPe	Select the rising or falling edge for the trigger.
SOURce	Select the source that will produce the trigger.

Reset Conditions

When the Oscilloscope is sent a *RST (reset), certain command parameters are set to their default values. Unless these parameters are changed prior to performing the measurement, the reset values will be used.

Table 5-2 lists the reset values for the Oscilloscope module. All parameters not listed will remain in the state last selected.

Table 5-2. *RST (Reset) Conditions and Values

Parameter	Default	Description
CALibration		
:TNULl	0,0,0	Time nulls to 0 seconds.
CORRection		
:AFACtor	1:1	Probe attenuation factor is 1:1 on all inputs.
CALCulate		
:MATH1 and 2	OFF	Math function 1 and 2 off.
[:EXPRession]	INP1+INP1	Expression to add input 1 to input 1.
FORMat		
[:DATA]	INTeger 8	Waveform format to 8 bit integer.
INITiate		
:CONTinuous	AUTO	Automatically sweeps.
MEASure	NULL	Reset all configurations, last measurement is invalid.
MEMory:VME		
:ADDRess	200000H	External memory address space (hex).
:SIZE	8000H	External memory size in bytes (hex).
:STATe	OFF	External memory disabled.
:MEASure		
:ADDRess	200000H	External measurement address space.
:STATe	OFF	External measurement address is disabled.
OUTPut	OFF	Output trigger set to off.
:TTLTrg	OFF	TTL trigger lines 0-7 set to off.
:ECLTrg	OFF	ECL trigger lines 0-1 set to off.
:EXTernal	OFF	External trigger set to off.

Table 5-2. *RST (Reset) Conditions and Values —Continued

Parameter	Default	Description
[SENSe:]		
AVERage		
:COUNt	1	Acquisition complete in 1 value.
[:STATe]	OFF	Average acquisition mode off.
:TYPE	SCALar	Acquisition mode is scalar, complete in 1 count.
INPut	1	Input 1 on, inputs 2-4 off.
:COUPling	DC	Coupling to DC on all inputs.
:FILTer		
[:LPASs]	OFF	Internal low pass filter off on all inputs.
:HPASs	OFF	Internal high pass filter off on all inputs.
:IMPedance	1E6	Impedance to 1MΩ on all inputs.
SWEep		
:POINts	500	Acquisition record contains 500 pts.
:COMPlete	100	Acquisition complete when at 100%.
:TIME	1 ms	Full scale horizontal time (range/span) to 1 ms.
:CENTer	0	Center of range/span is 0 seconds.
:DELay	0	Sweep delay to 0 seconds.
:LINK	CENTer	Delay reference set to center of sweep.
:STARt	–500 µs	Start of range/span is –500 µs.
:STOP	+500 µs	Range/span stopping point is 500 µs.
VOLTage		
:RANGe		
:LOWer	–2	Bottom of range is –2 V on all inputs.
:PTPeak	4	Full scale vertical display is 4 V on all inputs.
:OFFSet	0	Center range is 0 V on all inputs.
:UPPer	2	Top of range is 2 V on all inputs.
TRIGger		
:ECOunt	2	Holdoff set to 2 counts.
:HYSTeresis	OFF	Noise reject off.
:LEVel	0	Trigger activated at 0 V.
:SLOPe	POSitive	Positive edge trigger.
:SOURce	INP1	Input 1 produces trigger.

Measurement Sequence

Oscilloscope measurements are most successful if the following measurement sequence is followed.

1. **Determine if a firmware calibration is required** (due to time, operating temperature differences, or measurement accuracy requirements). See Firmware Calibration later in this chapter for more information.
2. **Know the signal and type of measurement.** Remember, in most cases you will not have a displayed waveform to view. Have an understanding of the type of signal you want to measure; its amplitude and frequency; is it repetitive? An understanding of the signal you want to measure, and the type of measurement being performed, will help you select the correct oscilloscope and measurement setups. See Measurement Considerations later in this chapter for more information.
3. **Set the Oscilloscope controls** (input, correction, voltage, sweep, and trigger). See Oscilloscope Setup later in this chapter for more information.
4. **Set the measurement controls.** See Measurement Setup later in this chapter for more information.
5. **Digitize the Waveform.** Digitizing the waveform ensures that all measurements are performed using the same data, and that the data obtained is valid. The user defines the acquisition and completion criteria for the waveform data. This step is performed using the INITiate command, and is done automatically when using the MEASure commands. See Digitizing Waveforms in this chapter for more information.
6. **Perform the measurement.** See Performing a Measurement later in this chapter for more information.
7. **Read the results.** See Performing a Measurement later in this chapter for more information.

Notes

It is critical that the oscilloscope controls are setup for the specific measurement being performed. Just because measurement results are obtained does not guarantee that they are correct. It is highly recommended that all measurement results be analyzed to verify authenticity. **A returned +9.99999E+37 indicates an invalid measurement.**

See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Firmware Calibration

There are two levels of calibration for the Oscilloscope module. The first level, called vertical, delay, time null, and logic trigger calibrations, can be performed by the operator. Procedures are provided in this section. The second level of calibration should be performed only by qualified service personnel using the service manual.

First level calibration should **ONLY** be performed under the following conditions:

- at six month intervals or every 1000 hours of use,
- if the ambient temperature changes more than 10°C from the temperature at full calibration,
- or to optimize measurement accuracy.

Caution

Do not remove the module with power applied to the mainframe.

The firmware calibration procedures should only be performed after the instrument has run for one hour at ambient temperature installed in the mainframe.

Notes

It is **NOT** necessary to perform first level calibration procedures prior to every operation.

When performing a first level calibration, **all procedures should be done in the order given.**

After calibrating, you **MUST** perform an SYSTem:AUToscale or *RST (reset) to return to normal operation.

Vertical Cal Procedure

Vertical calibration is performed on inputs one through four (simultaneously) using the following procedure:

1. Set the CAL FACTOR PROTECT switch to **UNPROTECTED.**
2. Connect the Oscilloscope DC Calibrator Output connector to the Input 1-4 connectors.

Note

Verify that the BNC cables are not longer than 1 meter and as close in length as possible.

3. Load the "default" calibration data.
4. Perform clear status, reset, then preset the Oscilloscope.

5. Select and start the vertical calibration routine. Calibration will last for approximately 15 minutes. During calibration, access and error LED's will be on.

Note

If the calibration time exceeds 15 minutes, the calibration is invalid. Recycle power then repeat procedure. If the condition repeats the instrument is malfunctioning. Refer to the Service Manual for information on correcting the malfunction before proceeding.

6. After calibration is complete (LED's to off), read the calibration event register to verify that no errors were generated during the procedure.
   - If "0" is returned, the calibration was successful.
   - If "0" is not returned, the calibration attempt was unsuccessful. See Chapter 4, CALibrate Subsystem for more information.
7. Disconnect cables and perform Delay Calibration Procedure.

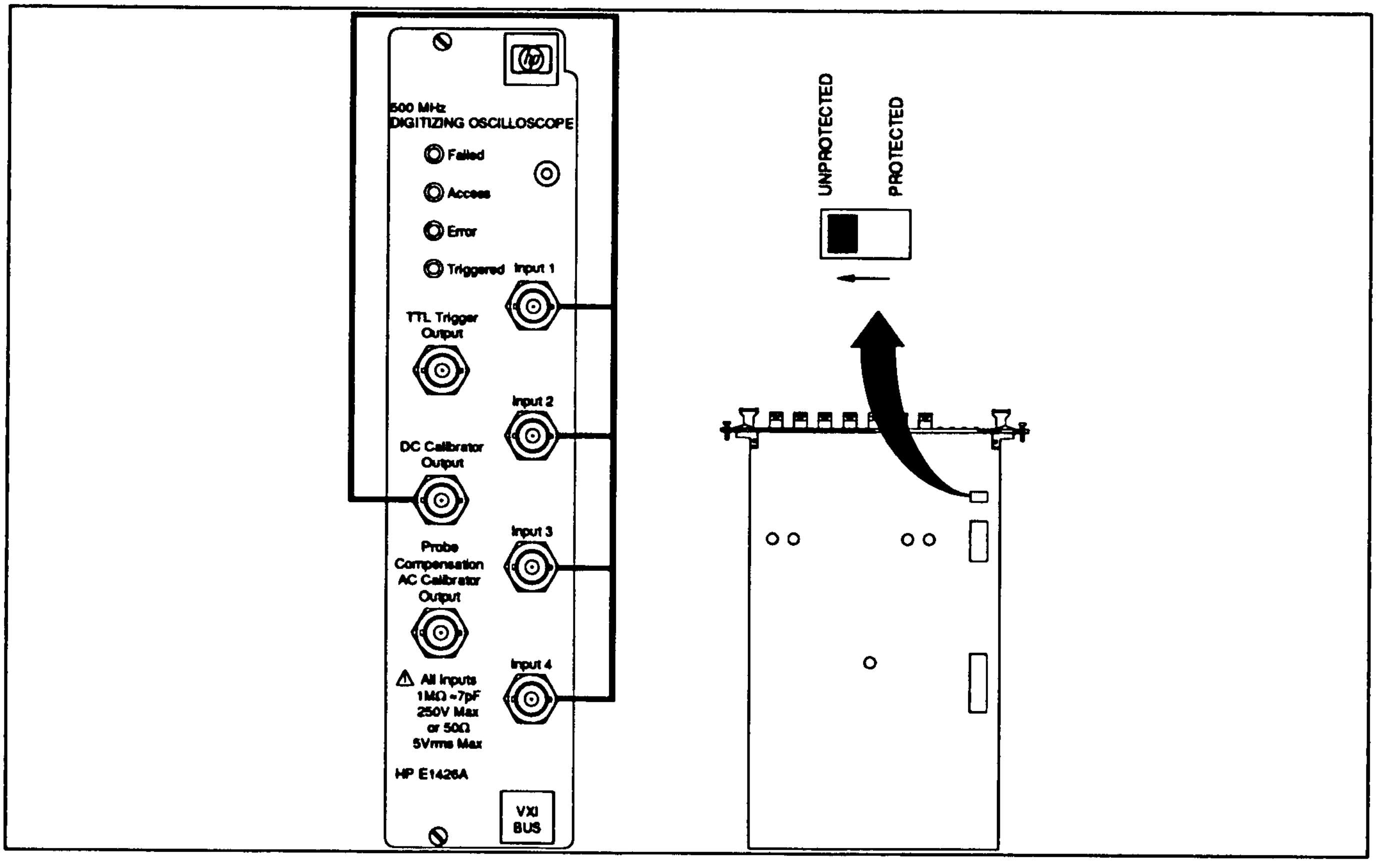


Figure 5-1. Example: Vertical Calibration Setup

Example

The following example shows how to perform a vertical calibration. The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10 OUTPUT 70905;"CAL:SCAL:DCAL"     Configure for a default calibration
                                     routine.
 20 OUTPUT 70905;"CAL:SCAL:BCAL"     Load default calibration data.
 30 OUTPUT 70905;"*CLS"              Clear status.
 40 OUTPUT 70905;"*RST"              Resets the Oscilloscope to its default
                                     state.
 50 OUTPUT 70905;"STAT:PRES"         Presets the Oscilloscope.
 60 OUTPUT 70905;"CAL:SCAL:VERT"     Selects vertical calibration routine.
 70 OUTPUT 70905;"CAL:SCAL:BCAL"     Starts vertical calibration routine.
 80 OUTPUT 70905;"STAT:QUES:CAL?"    Read calibration event register.
 90 ENTER 70905;A                    Enter calibration event register results.
100 PRINT A                          Print calibration event register results.
110 END                              Terminate program.
```

Delay Cal Procedure

Delay calibration is performed on inputs one through four (one at a time) using the following procedure:

1. Verify the CAL FACTOR PROTECT switch is set to **UNPROTECTED**.
2. Connect the Oscilloscope AC Calibrator Output connector to the Input 1 connector using a BNC cable.

Note

Verify that the BNC cable is not longer than 1 meter.

3. Perform clear status, reset, then preset the Oscilloscope.
4. Select and start the delay calibration routine. During calibration, the access and error LED's will be on.

Note

If the calibration time exceeds 1 minute, the calibration is invalid. Recycle power then repeat procedure. If the condition repeats the instrument is malfunctioning. Refer to the Service Manual for information on correcting the malfunction before proceeding.

5. After calibration is complete (LED's to off), read the calibration event register to verify that no errors were generated during the procedure.
   - If "0" is returned, the calibration was successful.
   - If "0" is not returned, the calibration attempt was unsuccessful. See Chapter 4, CALibrate Subsystem for more information.
6. Disconnect cable from input 1 and reconnect to input 2. Repeat steps 4 and 5 for input 2.
7. Disconnect cable from input 2 and reconnect to input 3. Repeat steps 4 and 5 for input 3.
8. Disconnect cable from input 3 and reconnect to input 4. Repeat steps 4 and 5 for input 4.
9. When complete, disconnect BNC cable and perform Time Null Calibration Procedure.

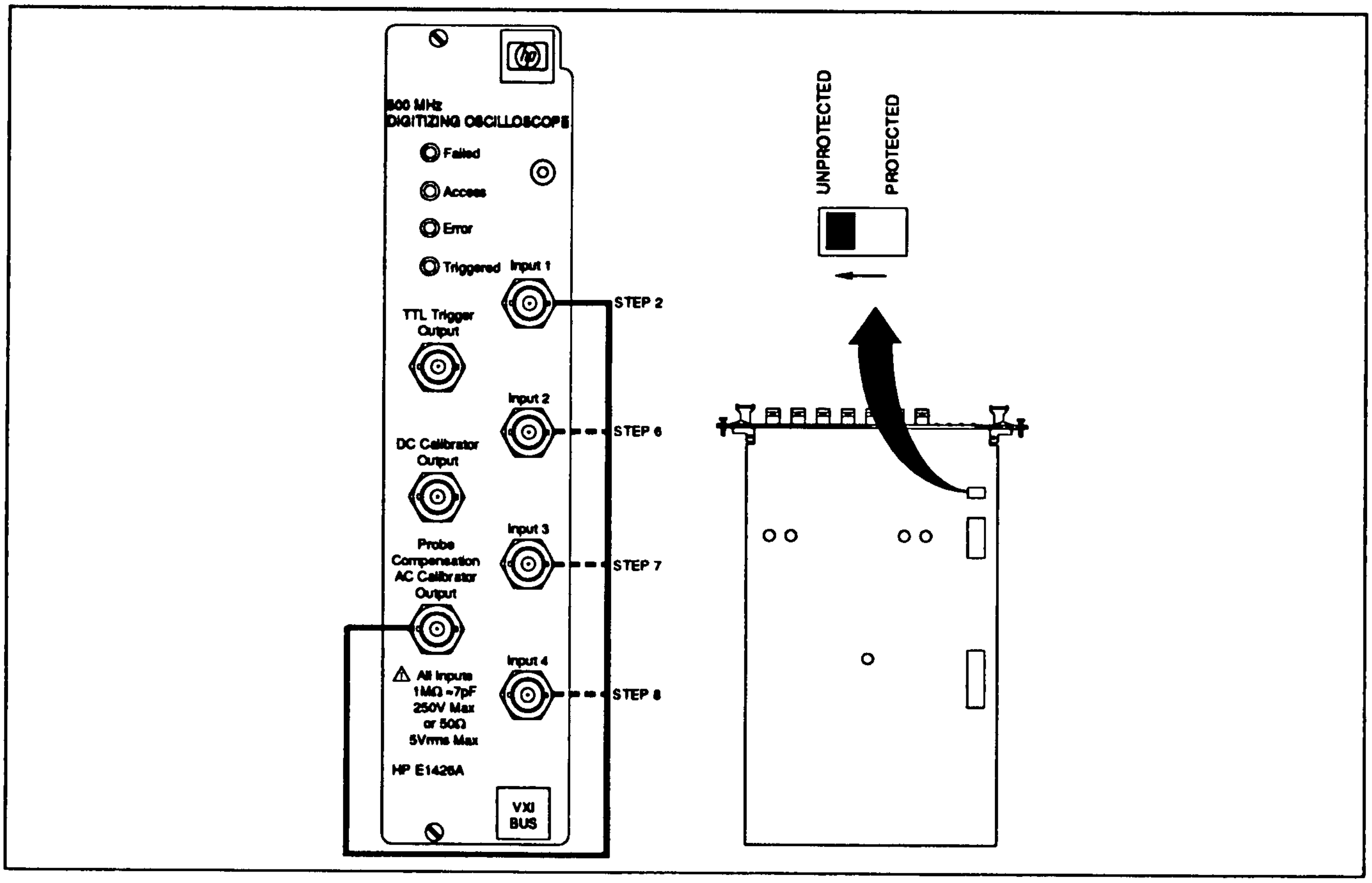


Figure 5-2. Example: Delay Calibration Setup

Example

The following example shows how to perform a delay calibration. The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10  OUTPUT 70905;"*CLS"              Clear status.
 20  OUTPUT 70905;"*RST"              Resets the Oscilloscope to its default
                                      state.
 30  OUTPUT 70905;"STAT:PRES"         Presets the Oscilloscope.
 40  FOR I=1 TO 4                     Input testing loop.
 50  OUTPUT 70905;"CAL:SCAL:DEL INP"&VAL$(I)
                                      Selects input n delay calibration
                                      routine (n=input 1-4).
 60  OUTPUT 70905;"CAL:SCAL:BCAL"     Starts delay calibration routine.
 70  OUTPUT 70905;"STAT:QUES:CAL?"    Read calibration event register.
 80  ENTER 70905;B                    Enter calibration event register results.
 90  PRINT B                          Print calibration event register results.
100  PAUSE                            Pause to connect next input.
110  NEXT I                           Repeat for inputs 2-4.
120  END                              Terminate program.
```

Time Null Cal Procedure

Time Null calibration is performed on inputs one-two, one-three, and one-four (one at a time) using the following procedure:

1. Verify the CAL FACTOR PROTECT switch is set to **UNPROTECTED**.
2. Connect the Oscilloscope AC Calibrator Output connector to both the Input 1 and Input 2 connectors.

Note

Verify that the BNC cables are not longer than 1 meter and equal in length.

3. Perform clear status, reset, then preset the Oscilloscope.
4. Select and start the time null calibration routine. During calibration, the access and error LED's will be on.

Note

If the calibration time exceeds 1 minute, the calibration is invalid. Recycle power then repeat procedure. If the condition repeats the instrument is malfunctioning. Refer to the Service Manual for information on correcting the malfunction before proceeding.

5. After calibration is complete (LED's to off), read the calibration event register to verify that no errors were generated during the procedure.
   - If "0" is returned, the calibration was successful.
   - If "0" is not returned, the calibration attempt was unsuccessful. See Chapter 4, CALibrate Subsystem for more information.
6. Disconnect cable from input 2 and reconnect to input 3. Repeat steps 4 and 5 for input 3.
7. Disconnect cable from input 3 and reconnect to input 4. Repeat steps 4 and 5 for input 4.
8. When complete, disconnect BNC cable and perform the Logic Trigger Calibration Procedure (if necessary).

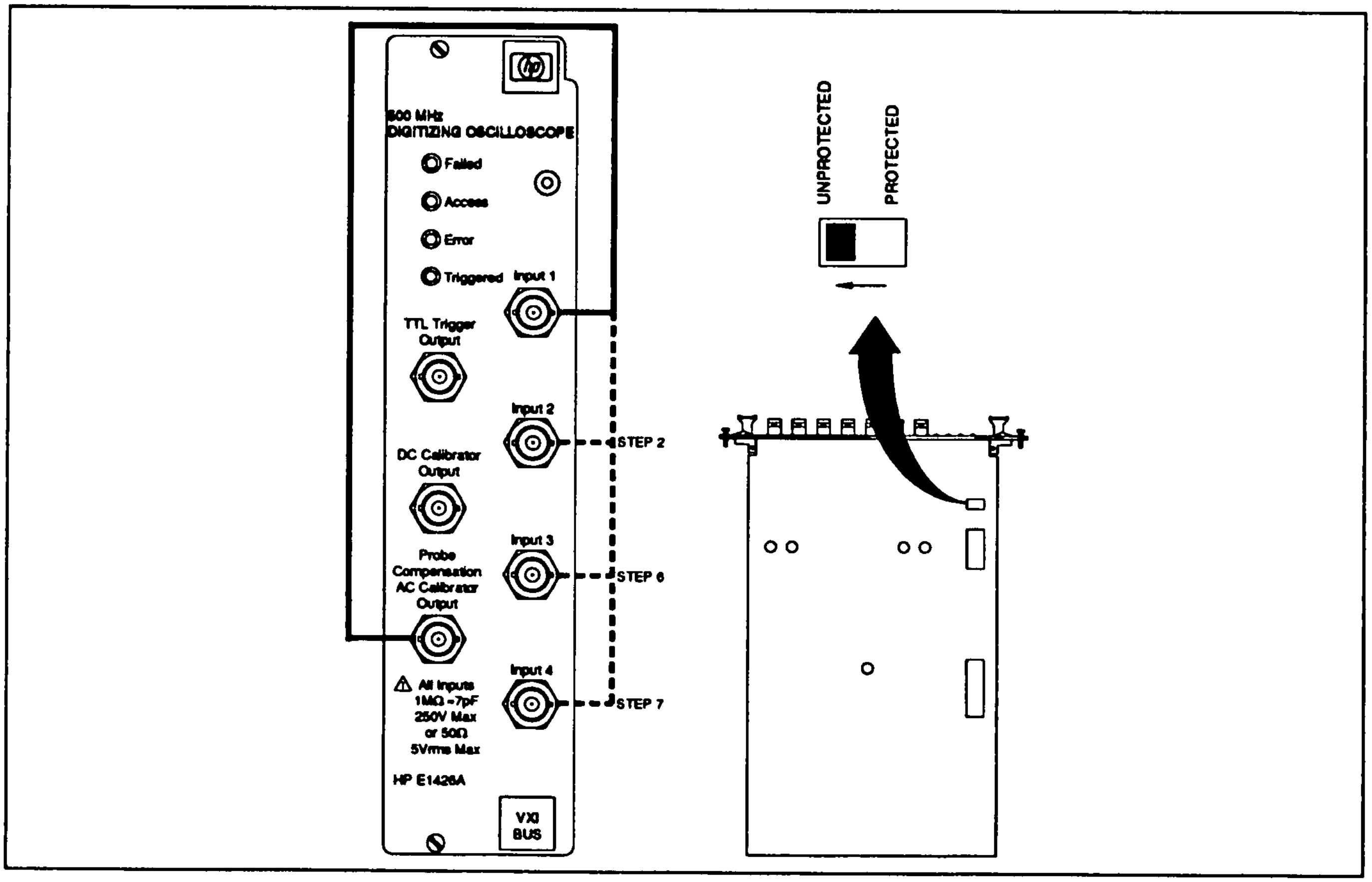


Figure 5-3. Example: Time Null Calibration Setup

Example

The following example shows how to perform a time null calibration. The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10 OUTPUT 70905;"*CLS"                  Clear status.
 20 OUTPUT 70905;"*RST"                  Resets the Oscilloscope to its default
                                         state.
 30 OUTPUT 70905;"STAT:PRES"             Presets the Oscilloscope.
 40 FOR I=2 TO 4                         Input testing loop.
 50 OUTPUT 70905;"CAL:SCAL:TNUL INP1TO";&VAL$(I)
                                         Selects input 1 to n time null
                                         calibration routine (n=input 2-4).
 60 OUTPUT 70905;"CAL:SCAL:BCAL"         Starts time null calibration routine.
 70 OUTPUT 70905;"STAT:QUES:CAL?"        Read calibration event register.
 80 ENTER 70905;C                        Enter calibration event register results.
 90 PRINT C                              Print calibration event register results.
100 PAUSE                                Pause to connect next input.
110 NEXT I                               Repeat for inputs 3-4.
120 END                                  Terminate program.
```

Logic Trigger Cal Procedure

Logic trigger calibration is performed on input one using the following procedure:

Note

It is **NOT** necessary to perform this procedure unless very accurate delay by time triggering is required during measurements.

If this procedure is not performed, set the CAL FACTOR PROTECT switch to **PROTECTED**.

1. Verify the CAL FACTOR PROTECT switch is set to **UNPROTECTED**.
2. Connect the Oscilloscope AC Calibrator Output connector to the Input 1 connector using a BNC cable.

Note

Verify that the BNC cable is not longer than 1 meter.

3. Read the calibration register to verify that no errors are present.
   - If "0" is returned, proceed with step 5.
   - If "0" is not returned, the Logic Trigger calibration must be terminated, and the cause of the error corrected. See Chapter 4, CALibrate Subsystem for more information.
4. Perform clear status, reset, then preset the Oscilloscope.

5. Locate the logic trigger adjustment on the right side of the Oscilloscope module, and the error and access LED's on the front panel.

6. Select and start the logic trigger calibration routine. Verify that the triggered LED flashes.

   Observe the access and error LED's.

   - If both are on, no adjustment is required.
   - If only one is ON, rotate the logic trigger adjustment until both LED's remain ON.

   Approximately 15 seconds after no further rotation of the adjustment, the triggered LED will flash faster, then all LED's will go out.

7. After calibration is complete (LED's to off), read the calibration register to verify that no errors were generated during the procedure.

   - If "0" is returned, the calibration was successful.
   - If "0" is not returned, the calibration attempt was unsuccessful. See Chapter 4, CALibrate Subsystem for more information.

8. Disconnect BNC cable and set the CAL FACTOR PROTECT switch to **PROTECTED**.

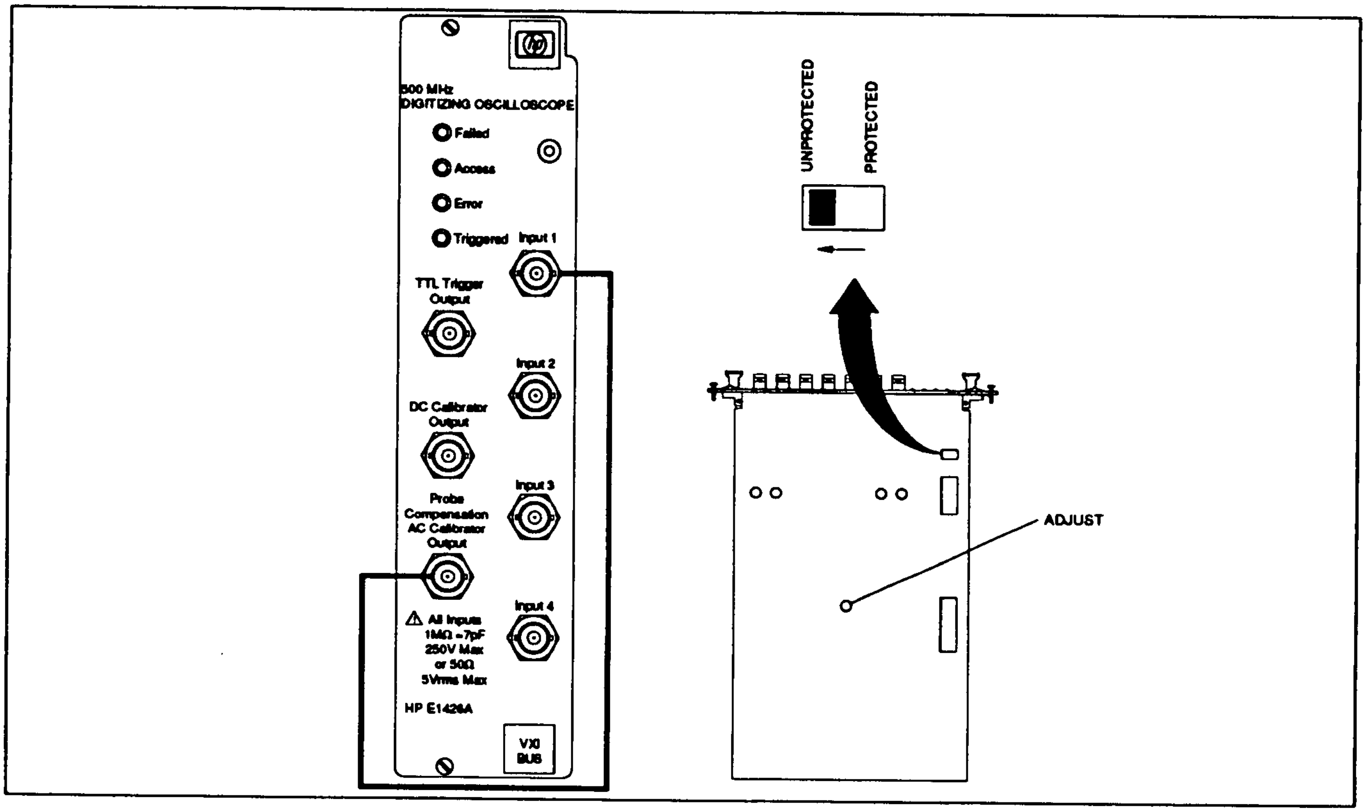


Figure 5-4. Example: Logic Trigger Calibration Setup

Example

The following example shows how to perform a logic trigger calibration. The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10 OUTPUT 70905;"*CLS"             Clear status.
 20 OUTPUT 70905;"*RST"             Resets the Oscilloscope to its default
                                    state.
 30 OUTPUT 70905;"STAT:PRES"        Presets the Oscilloscope.
 40 OUTPUT 70905;"STAT:QUES:CAL?"   Read calibration event register.
 50 ENTER 70905;D                   Enter calibration event register results.
 60 PRINT D                         Print calibration event register results.
 70 IF D<>0 THEN 130                Terminate if results not 0.
 80 OUTPUT 70905;"CAL:SCAL:LTC"     Selects logic trigger calibration
                                    routine.
 90 OUTPUT 70905;"CAL:SCAL:BCAL"    Starts logic trigger calibration routine.
100 OUTPUT 70905;"STAT:QUES:CAL?"   Read calibration event register.
110 ENTER 70905;E                   Enter calibration event register results.
120 PRINT E                         Print calibration event register results.
130 END                             Terminate program.
```

Starting a Measurement

Before configuring the Oscilloscope to perform a specific measurement, it is very beneficial to know about the signal being measured, and the type of measurement being performed.

Unknown Input Signal

If the input signal is unknown, SYSTem:AUToscale can be used with any MEASure? query to quickly determine some of the critical input signal parameters.

The following example shows how to perform a frequency measurement on a signal connected to Input 1. The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
10  OUTPUT 70905;"*CLS"                 Clear status.
20  OUTPUT 70905;"*RST"                 Resets the Oscilloscope to its
                                        default state.
30  OUTPUT 70905;"SYST:AUT"             Perform autoscale.
40  OUTPUT 70905;"MEAS:VOLT:FREQ? (@INP1)"
                                        Measure input 1 frequency.
50  ENTER 70905;Results                 Enter measurement results.
60  PRINT Results                       Print measurement results.
70  END                                 Terminate program.
```

Note

Performing certain measurements will return invalid data (+9.99999E+37) because the proper portion of the waveform was not present during the measurements.

See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Measurement Considerations

In order to make a specific measurement, the portion of the waveform required for that measurement must be setup and present on the oscilloscope. For example, to measure:

- Period or frequency - a minimum of one complete cycle must be present.
- Pulse width - the entire pulse must be present.
- Rise Time - the leading (positive-going) edge of the waveform must be present.
- Fall Time - the trailing (negative-going) edge of the waveform must be present.

Performing this function on an Oscilloscope with a display is a comparatively simple task. However, when the display is removed, certain steps must be taken to assure the correct Oscilloscope and measurement set-ups are performed prior to the actual measurement.

Oscilloscope Setup

Before a specific measurement can be performed, it is necessary to setup the Oscilloscope controls. The settings are dependent on the input signal and the desired measurement being performed. There are two ways to set these controls:

- Automatic (using SYSTem:AUToscale).
- Manual - User enters desired values.

Automatic

When selected, the Oscilloscope automatically evaluates the input signals present at inputs 1-4, and then sets the controls to present the signal. You set up the oscilloscope with the following command:

```
SYSTem:AUToscale
```

Note

Autoscale should only be used with relatively stable input signals having a duty cycle of greater than 0.5% and a frequency greater than 50Hz.

Manual

Instructions for manual selection of the Input, Correction, Voltage, Sweep, and Triggering controls are provided later in this section.

Input Setup

This section discusses the vertical or input controls you can program with the INPut command. These controls allow the selection of:

- Input State
- Input Coupling
- Input Impedance
- Input Filter State

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL INPut commands available, see Chapter 6, [SENSe:] subsystem.

Input State

Each input can be enabled or disabled. Input 1 is on, and inputs 2-4 are off at reset. You enable an input with the following command:

```
INPn:STAT xxx        (n=Input number and xxx = ON or OFF)
```

Input Coupling

Coupling for each input can be set to AC, DC. DC Coupling is selected at reset. You select coupling with the following command:

```
INPn:COUP xxx        (n=input number and xxx = AC, DC)
```

Input Impedance

Impedance for each input can be set to 1MΩ or 50Ω. 1MΩ is selected at reset. You select impedance with the following command:

```
INPn:IMP xxx         (n=input number and xxx = 50 or 1E6)
```

Input Filter State

Two input filters are selectable to provide low-pass (BW≈30 MHz) or high-pass (BW~450 Hz) filtering. Both filters are disabled at reset.

You enable the low-pass filter with the following command:

```
INPn:FILT:LPAS xxx   (n=input number and xxx = ON or OFF)
```

You enable the high-pass filter with the following command:

```
INPn:FILT:HPAS xxx   (n=input number and xxx = ON or OFF)
```

Correction Setup

This section discusses the vertical or input controls you can program with the CORRection command. These controls allow the selection of:

- Input Probe Attenuation

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available correction commands, see Chapter 6, [SENSe:] subsystem.

Input Probe Attenuation

Probe attenuation factor for each input can be entered from 0.9:1 to 1000.0:1 to match the probe currently connected at the input. 1:1 is selected at reset. You enter probe attenuation factor with the following command:

```
CORRn:AFAC xxx      (n=input number and xxx = value)
```

Note

Changing probe attenuation from 1:1 will effect current settings of input range and offset.

Voltage Setup

This section discusses the vertical or input controls you can program with the VOLTage command. These controls allow the selection of:

- Input Offset
- Input Range

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available voltage commands, see Chapter 6, [SENSe:] subsystem.

Input Offset

Offset voltage for each input can be entered to a level depending on the current full scale range (PTPeak) selection. 0 volts is selected at reset. You enter offset with the following command:

```
VOLTn:RANG:OFFS xxx      (n=input number and xxx = value in
                          volts)
```

Input Range

Full scale (not per division) vertical axis for each input can be entered from 8 mV to 40 V. 4 volts is selected at reset. You enter vertical full scale range with the following command:

```
VOLTn:RANG:PTP xxx       (n=input number and xxx = value in
                          volts)
```

Note

Changes in probe attenuation will effect current settings of input range and offset.

Sweep Setup

This section discusses the time base or horizontal controls you can program with the SWEep command. These controls allow the selection of:

- Range
- Delay
- Reference

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available sweep commands, see Chapter 6, [SENSe:] subsystem.

Range

Full scale (not per division) horizontal axis can be entered from 2ns to 50 seconds. 1 ms is selected at reset. You enter horizontal full scale range with the following command:

```
SWE:TIME:RANG xxx    (xxx = value in seconds)
```

Delay

The sweep delay (time interval between trigger event and sweep delay reference point) can be set to a value depending on the current full scale range (TIME) selected. 0 seconds is selected at reset. See Reference below for more information. You select sweep delay with the following command:

```
SWE:TIME:DEL xxx     (xxx = value in seconds + or –)
```

Reference

Three different settings are available that control the sweep reference point. STARt, CENTer, or STOP sets the reference to the left, center, or right of the sweep, respectively. CENTer is selected at reset. Refer to Delay earlier in this section for more information. You enter reference with the following command:

```
SWE:TIME:DEL:LINK xxx     (xxx = STAR, CENT, or STOP)
```

Trigger Setup

This section discusses the trigger controls you can program with the TRIGger command. These controls allow the selection of:

- Holdoff
- Level
- Noise Rejection
- Slope
- Source

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available triggering commands, see Chapter 6, TRIGger subsystem.

Holdoff

Holdoff (disabling of trigger circuit for specific duration) can be set to a count from 2 to 16 million. 2 is selected at reset. You select holdoff with the following command:

```
TRIG:ECO xxx     (xxx = value in events)
```

Level

Active trigger level voltage can be entered to a value dependent on VOLTage:PTPeak and :OFFSet settings as follows:

±0.75 of selected range from current offset

0 volts is selected at reset. You enter trigger level with the following command:

```
TRIG:LEV xxx     (xxx = value in volts)
```

Noise Rejection

Noise rejection can be turned ON or OFF for the selected SOURce. Aids in eliminating false triggering. OFF is selected at reset. You enable noise rejection with the following command:

```
TRIG:HYST xxx    (xxx = ON or OFF)
```

Slope

The POSitive (rising) or NEGative (falling) edge of the input signal can be selected as the trigger event for the selected source. POSitive is selected at reset. You select trigger slope with the following command:

```
TRIG:SLOP xxx    (xxx = POS or NEG)
```

Source

The trigger source can be selected from one of the input signals (INPut 1-4), or from one of 10 bus lines (TTLTrg0-7 or ECLTrg 0-1). Only one trigger source can be specified at a time. INPut 1 is selected at reset. You select trigger source with the following command:

```
TRIG:SOUR  xxx     (xxx = INP1, INP2, INP3, INP4, TTLT0,
                   TTLT1, TTLT2, TTLT3, TTLT4, TTLT5,
                   TTLT6, TTLT7, ECLT0 or ECLT1)
```

Measurement Setup

After the input, correction, voltage, sweep, and triggering controls are setup, it is necessary to set any unique measurement parameters prior to performing the actual measurement.

Note

The only user defined measurement parameters available in SCPI are the upper and lower threshold limits during rise time and fall time measurements. These parameters are sent with the CONFigure or MEASure command. See Performing a Measurement later in this chapter for more information.

Digitizing Waveforms

Waveforms can be digitized to provide a waveform that fulfills user defined acquisition and completion criteria. The resulting waveform can be measured by the Oscilloscope or passed to the controller as a numerical representation. In addition, preamble data is passed to the controller so the digitized data can be interpreted. The user can specify exactly what the digitized information will contain, how the waveform is acquired, when the waveform is complete, and the format of the digitized information. Use the following sequence to assist in digitizing waveforms:

1. **Waveform Acquisition:** Defines the data type, completion criteria, number of averages, and number of data points for data acquisition. See Acquisition in this section for more information.
2. **Digitize the Waveform**: Acquires data on the specified input, stores the data in the input buffer, then stops the acquisition. See Digitize in this chapter for more information.
3. **Measure the Waveform**: All measurements made by the Oscilloscope are now performed on the same data. See Performing a Measurement in this chapter for more information.

Note

When using the CONFigure commands to perform a measurement, a digitize is performed when the INITiate or READ? commands are executed. When using the MEASure commands to perform a measurement, a digitize is automatically performed.

4. **Waveform Disposition:** When the data is going to/from the bus, it defines where to get the data, the format the data is to be in, reads the digitized waveform data, and reads the preamble data. See Disposition in this section for more information.

Acquisition

This section discusses the acquisition controls you can program with the [SENSe:] command. These controls allow the selection of:

- Type
- Completion Criteria
- Count
- Points

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available acquisition commands, see Chapter 6, [SENSe:] subsystem.

Type

Three different modes define the type of acquisition that will take place when the INITiate[:IMMediate] command is executed. See Chapter 6, [SENSe:]AVERage[:STATe] and [SENSe:]AVERage:TYPE, for information on SCALar, AVERage, and ENVelope modes. SCALar is selected at reset. You select acquisition type with the following command(s):

```
AVE xxx    (xxx = ON or OFF)
                AND
AVE:TYPE xxx    (xxx = SCAL or ENV)
```

Completion Criteria

Completion criteria for an acquisition can be entered from 0 to 100%. 100% is selected at reset. You enter completion criteria with the following command:

```
SWE:POIN:COMP xxx    (xxx = 0 to 100)
```

Count

The number of points to be averaged for each acquisition can be entered from 1 to 2048. 8 is selected at reset. You enter counts with the following command:

```
AVER:COUN xxx    (xxx = 1 to 2048)
```

Points

The number of points for each acquisition record can be entered from 32 to 1024. 500 is selected at reset. You enter counts with the following command:

```
SWE:POIN xxx    (xxx = 32 to 1024)
```

Digitize

The INITiate[:IMMediate] command causes an acquisition to take place on the enabled input(s) with the resulting data placed in the input buffer. Upon completion, the data acquisition is stopped. Inputs are enabled using the INPut[:STATe] command. See Chapter 6, INITiate subsystem for more information. You digitize with the following command:

```
INITiate[:IMMediate]
```

Note

See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Disposition

This section discusses the disposition controls you can program with the TRACe command. These controls allow the selection of:

- Format
- Reading Waveform Data
- Reading Preamble Data
- Using the Digitized Data

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available waveform commands, see Chapter 6, TRACe subsystem.

Format

Two different formats are available to format digitized data when retrieved from the instrument. See Chapter 6, FORMat[:DATA] for information on 8 bit and 16 bit integer formats. 8 bit is selected at turn-on or reset. You select format with the following command:

```
FORM INT,xxx    (xxx = 8 or 16)
```

Reading Waveform Data

The digitized data is read over the bus from the waveform memory, input buffer, or math function. You read digitized data with the following query:

```
TRAC? xxx    (xxx=INP1-4, MATH1-2, or WMEM1-4)
```

Reading Preamble Data

The interpretation (preamble) data is read over the bus from the waveform memory, input buffer, or math function. This data is used to interpret the waveform data. You read preamble data with the following query:

```
TRAC:PRE? xxx    (xxx=INP1-4, MATH1-2, or WMEM1-4)
```

Using the Digitized Data

The returned data is read from the instrument starting at the left-most point on the active waveform, and must be scaled for useful interpretation. The values needed to perform this task (X and Y coordinates) are included in the preamble data. See Chapter 6, TRACe subsystem for more information.

Example: Digitize a Waveform and Send Data to the Controller

This example uses the Oscilloscope module to digitize a waveform from INPut 1, and send the results (waveform and preamble) to the controller. The waveform will be complete when 512 points have been averaged at least four times. The digitized data sent to the controller is to be in 8 bit format.

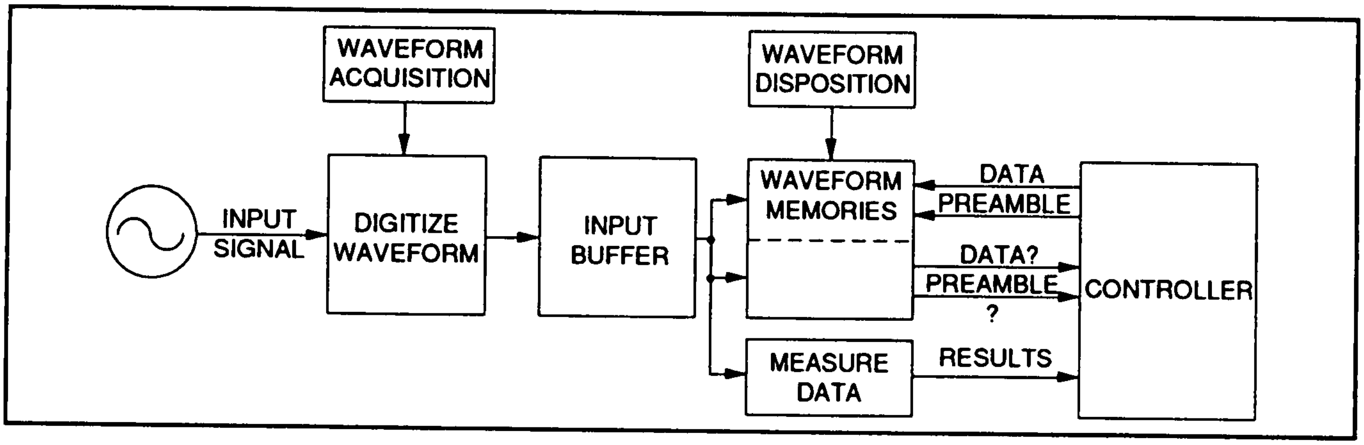


Figure 5-5. Example: Digitizing Waveforms

The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10 DIM Pre$[100]                      String for preamble data.
 20 INTEGER Waveform(2000)             Temporary dimension for
                                       waveform data data.
 30 OUTPUT 70905;"*CLS"                Clear status.
 40 OUTPUT 70905;"*RST"                Resets the Oscilloscope to its
                                       default state (table 3-2).
 50 OUTPUT 70905;"SYST:AUT"            Perform an autoscale.
 60 OUTPUT 70905;"AVER ON"             Acquisition type to average.
 70 OUTPUT 70905;"AVER:COUN 4"         Number of averages to 4.
 80 OUTPUT 70905;"SWE:POIN:COMP 100"   Completion criteria to 100%.
 90 OUTPUT 70905;"SWE:POIN 512"        Data record to 512 points.
100 OUTPUT 70905;"INP1 ON"             Input 1 is the source.
110 OUTPUT 70905;"ABOR"                Stop all data acquisition.
120 OUTPUT 70905;"INIT"                Digitize input 1 and place data in
                                       input output buffer.
130 OUTPUT 70905;"FORM INT,8"          Format for waveform data is 8
                                       bit.
140 OUTPUT 70905;"TRAC:PRE? INP1"      Read preamble data from input 1.
150 ENTER 70905;Pre                    Enter preamble data.
```

— CONTINUED—

```
160 OUTPUT 70905;"TRAC:DATA? INP1"          Read waveform data from input
                                            1.
170 GOSUB Get_data                          Go to a subroutine that will read
                                            the header to determine the size of
                                            the waveform data, then re-
                                            dimension the waveform data
                                            array.
180 STOP                                    Stop the main program.
190 !
200 Get_data: !                             Data retrieve routine.
210 ENTER 70905 USING "#,1A";One_char$      Enter one character at a time.
220 IF One_char$="#" THEN Found_pound       If "#" read, go to the read routing.
230 GOTO 210                                Loop to read next character.
240 Found_pound: !
250 ENTER 70905 USING "#,1D";Digits         Read and save first digit after "#".
260 ENTER 70905 USING "#,"&VAL$(Digits)&"D";Length
                                            Read the next XXX characters as
                                            the record length, where XXX is
                                            specified by Digits.
270 REDIM Waveform(1:Length)                Re-dimension Waveform to the
                                            actual record size.
280 ENTER 70905 USING "#,B";Waveform(*)     Enter waveform data.
290 ENTER 70905 USING "#,B";Crlf            Read carriage return.
300 RETURN                                  Return to the main program.
310 END                                     Terminate program.
```

Comments

Block Data. Both preamble and waveform data is "definite-length block response data". This method allows any type of device-dependent data to be transmitted over the system interface as a series of 8-bit binary data types. This is particularly useful for sending large quantities of data or 8-bit extended ASCII codes. The syntax is a pound sign (#) followed by a non-zero digit representing the number of digits in the decimal integer. After the non-zero digit is the decimal integer that states the number of 8-bit data bytes being sent. This is followed by the actual data. For example, for transmitting 1024 bytes of data, the syntax is:

```
                     #41024<1024 bytes of data><term>
Number of digits  __|  |           |                |
Number of bytes   _____|           |                |
Actual Data   _____________________|                |
Terminator   ___________________________________ ___|
```

Reading Block Data. The example program (lines 200 to 300) finds the "#" sign, reads the number of digits and number of bytes, then re-defines the field to the correct length. This method will work for all definite-length block response returned data (e.g., TRACe[:DATA]?.

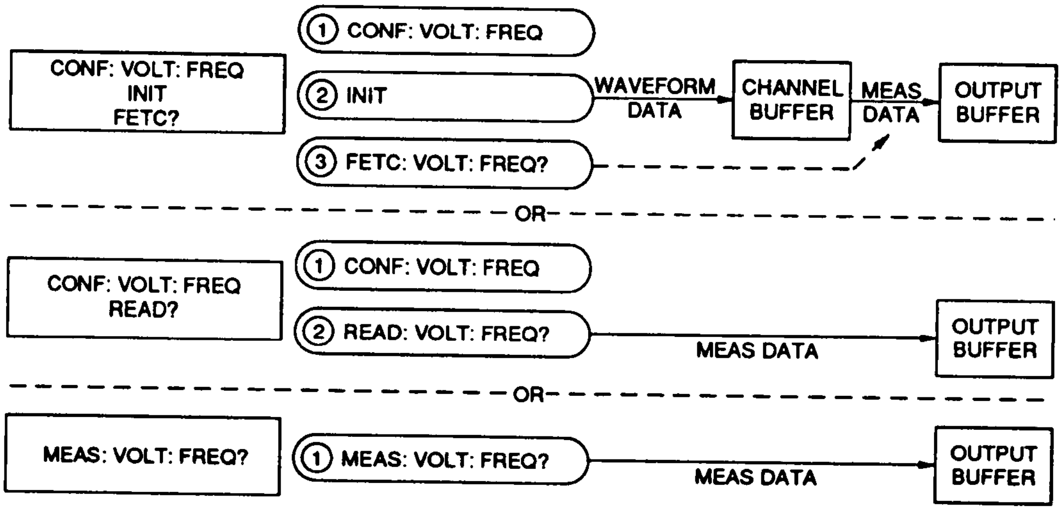
When to Read Preamble. The preamble should be read prior to the waveform data.

When to Read Waveform Data. To obtain waveform data, you must specify the TRACe parameters for the waveform data prior to sending the TRACe[:DATA]? query. After receiving the :DATA? query, the instrument will start passing the waveform information to the controller when addressed to talk.

Performing a Measurement

After the Oscilloscope and measurement have been setup, the actual measurement can be performed. Because SCPI supports a number of different "levels" of commands, a number of different commands can be used to select and perform the same measurement function.

For example, the MEASure and CONFigure commands will each setup a FREQuency measurement. The illustration below shows how the MEASure and CONFigure commands differ in how they are used with READ?, INITiate, and FETCh? commands, and how they all perform the same measurement and get the data to the output buffer.



Notes

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of ALL available measurement commands, see Chapter 6, CONFigure, INITiate, READ, FETCh, and MEASure subsystems.

Remember, just because measurement results are obtained does not guarantee that they are correct. It is highly recommended that all measurement results be analyzed to verify authenticity. Incorrect oscilloscope and measurement setup can cause undesirable measurement results. A returned +9.99999E+37 indicates an invalid measurement.

See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

MEASure

MEASure is used to configure, initiate, and perform a measurement, then read the results. You select and perform a measurement with the following query:

```
MEAS:VOLT:function? (@xxxn)
```

```
(xxxn = input, waveform memory, or math number
function = AC, AMPL, [DC], DCYC, FALL:OVER, FALL:PRE,
FALL:TIME, FREQ, FTIM, HIGH, LOW, MAX, MIN, NDUT, NWID,
PDUT, PER, PWID, RISE:OVER, RISE:PRE, RISE:TIME, RTIM,
TMAX, TMIN)
```

The measurement data obtained is stored in the output buffer. An enter statement can be used to transfer this data to the computer.

Note

On rise time and fall time measurements, the user can enter upper and lower threshold parameters.

CONFigure

CONFigure only sets up the configuration, and does not perform the actual measurement. You configure for a measurement with the following query:

```
CONF:VOLT:function? (@xxxn)
```

```
(xxxn = input, waveform memory, or math number
function = AC, AMPL, [DC], DCYC, FALL:OVER, FALL:PRE,
FALL:TIME, FREQ, FTIM, HIGH, LOW, MAX, MIN, NDUT, NWID,
PDUT, PER, PWID, RISE:OVER, RISE:PRE, RISE:TIME, RTIM,
TMAX, TMIN)
```

Once the selected measurement is configured, the measurement is performed using a READ? or INITiate command.

Note

On rise time and fall time measurements, the user can enter upper and lower threshold parameters.

READ?

READ? performs a configured measurement and transfers the result to the output buffer. The actual measurement that is performed depends on if the function is included with the read query.

You perform and read the last configured measurement with the following query:

```
READ?
```

You perform and read a specific configured measurement with the following query:

```
READ:VOLT:function?     (function = AC, AMPL, [DC], DCYC,
                        FALL:OVER, FALL:PRE, FALL:TIME,
                        FREQ, FTIM, HIGH, LOW, MAX,
                        MIN, NDUT, NWID, PDUT, PER,
                        PWID, RISE:OVER, RISE:PRE,
                        RISE:TIME, RTIM, TMAX, TMIN)
```

In either case, the measurement result is then available to be transferred into the computer.

Notes

Remember, just because measurement results are obtained does not guarantee that they are correct. It is highly recommended that all measurement results be analyzed to verify authenticity. Incorrect oscilloscope and measurement setup can cause undesirable measurement results. **A returned +9.99999E+37 indicates an invalid measurement.**

See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

INITiate

INITiate has two commands that each perform a different task.

INITiate[:IMMediate] performs the configured measurement and transfers the result to the input buffer. You perform a configured measurement on all enabled inputs with the following command:

```
INIT
```

Note

You must perform an ABORt prior to executing the INITiate[:IMMediate] command.

The selected measurement is performed, and the data is stored in input buffer. Use the FETCh? query to transfer the result from input buffer to the output buffer.

INITiate:CONTinuous selects the sweep modes that are available. INITiate:CONTinuous ON requires a trigger event for each sweep. INITiate:CONTinuous OFF requires an INITiate[:IMMediate] command for each sweep. OFF is selected at reset. You select sweep mode with the following command:

```
INIT:CONT xxx     (xxx = ON or OFF)
```

FETCh?

FETCh? retrieves the measurement information from the input buffer, waveform memory, or math function, and transfers it to the output buffer. The actual measurement result that is retrieved depends on if the function is included with the fetch query.

You retrieve the last measurement result with the following query:

```
FETC?
```

You retrieve specific measurement results with the following query:

```
FETC:VOLT:function?        (function = AC, AMPL, [DC], DCYC,
                           FALL:OVER, FALL:PRE, FALL:TIME,
                           FREQ, FTIM, HIGH, LOW, MAX,
                           MIN, NDUT, NWID, PDUT, PER,
                           PWID, RISE:OVER, RISE:PRE,
                           RISE:TIME, RTIM, TMAX, TMIN)
```

In either case, the measurement result is then available to be transferred into the computer.

Notes

Remember, just because measurement results are obtained does not guarantee that they are correct. It is highly recommended that all measurement results be analyzed to verify authenticity. Incorrect oscilloscope and measurement setup can cause undesirable measurement results. A returned +9.99999E+37 indicates an invalid measurement.

See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Measurement Examples

The following is a list of examples provided to illustrate using SCPI commands to perform basic measurements/functions using the Oscilloscope module.

- Autoscale Period Measurement
- Manual Rise Time Measurement
- Manual Rise Time and Fall Time Measurement
- AC Voltage Measurement
- Using the Backplane Trigger

All the examples are written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Caution

MAXIMUM INPUT VOLTAGE. The maximum voltage that can be applied to the four input connectors is 5 Vrms at 50Ω or ±250 V (dc+peak ac<10 kHz) at 1MΩ.

Note

The following examples are intended to provide only a brief overview of the necessary commands required for basic operation. However, these examples can be used to provide a good starting point for much more complex programs. For a complete list and description of ALL SCPI commands, see Chapter 6.

Example: Autoscale Period Measurement

This example uses the Oscilloscope module to measure the period of an unknown signal connected to Input 2.

Caution

MAXIMUM INPUT VOLTAGE. The maximum voltage that can be applied to any of the four input connectors is 5 Vrms at 50Ω or ±250 V (dc+peak ac<10 kHz) at 1MΩ using a 1:1 Probe.

Note

Disconnect any signal on inputs 1, 3, and 4.

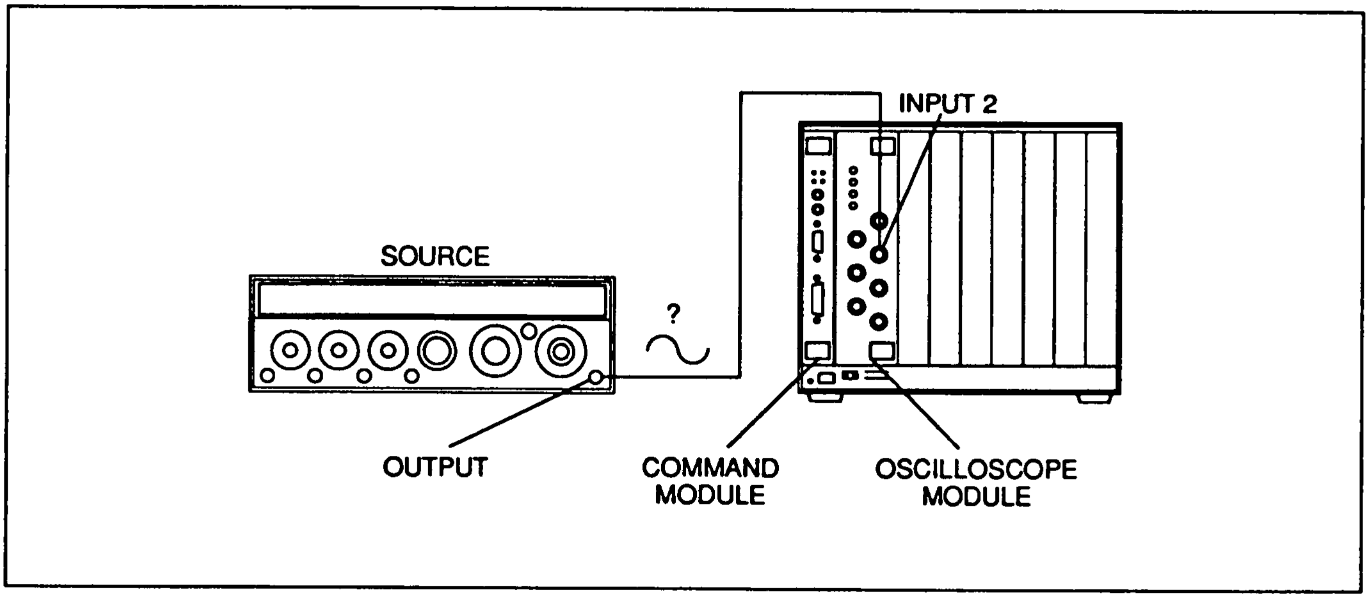


Figure 5-6. Example: Autoscale Period Measurement

Execute:

```
10  OUTPUT 70905;"*CLS"                  Clear status.
20  OUTPUT 70905;"*RST"                  Resets the Oscilloscope to its
                                         default state (table 5-2).
30  OUTPUT 70905;"SYST:AUT"              Perform autoscale.
40  OUTPUT 70905;"MEAS:VOLT:PER? (@INP2)"
                                         Measure input signal period, and
                                         read the measurement results
50  ENTER 70905;Results                  Enter measurement results.
60  PRINT Results                        Print measurement results.
70  END                                  Terminate program.
```

Comments

Signal Connection. An input signal must be connected to the Oscilloscope before initiating autoscale, and remain connected until the measurement is terminated.

Multiple Signal Connection using Autoscale. When more than one Oscilloscope input is connected to a signal source, initiating autoscale will setup the Oscilloscope controls using the signal connected closest to input 1. If a signal is not found on input 1 then input 2 is used, if a signal is not found on input 2 then input 3 is used, etc.

Measurement Time. The time it takes to complete a measurement is dependent on the input signal, acquisition type, and completion criteria. Under certain circumstances, a single measurement can take many hours to complete. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Example: Manual Rise Time Measurement

This example uses the Oscilloscope module to measure the rise time of a signal connected to Input 3 using a 1MΩ 10:1 probe. The expected input is a 1.5 V clock at 1 MHz. The upper threshold value should be set to 80% and the lower threshold value should be set to 20%. The user is notified if returned measurement results are not within specified limits.

Note

When measuring rise time, the leading (positive-going) edge of the waveform must be present. In order to obtain accurate results (example signal is a clock-minimum rise time):

- the sweep should be setup so the rising edge is maximized over the time base range, and
- the trigger should be set so the rising edge is centered.

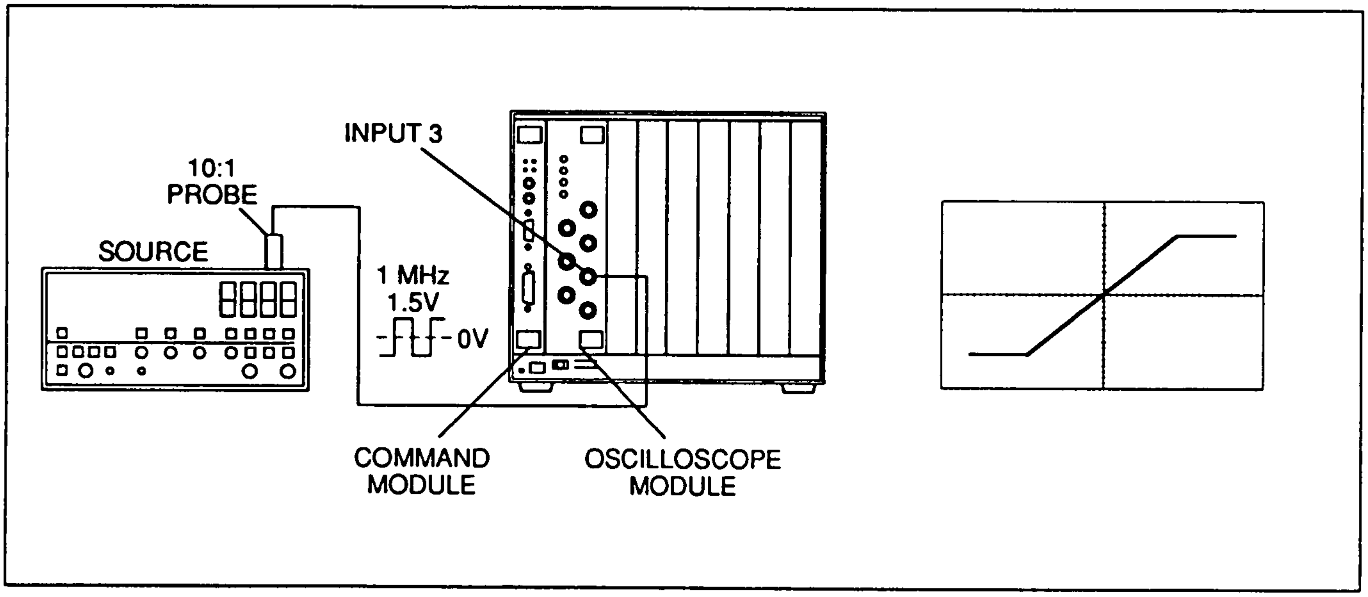


Figure 5-7. Example: Manual Rise Time Measurement

Execute:

```
10  OUTPUT 70905;"*CLS"                   Clear status.
20  OUTPUT 70905;"*RST"                   Resets the Oscilloscope to its
                                          default state (table 5-2).
30  OUTPUT 70905;"CORR3:AFAC 10"          Set input 3 probe attenuation
                                          factor at 10:1.
40  OUTPUT 70905;"VOLT3:RANG:PTP 2"       Set input 3 full scale vertical range
                                          to 2 volts (expected input is 1.5V).
50  OUTPUT 70905;"SWE:TIME:RANG 100E-9"   Set full scale horizontal range to
                                          100 ns (expected period).
60  OUTPUT 70905;"TRIG:SOUR INP3"         Trigger source to input 3.
70  OUTPUT 70905;"TRIG:SLOP POS"          Trigger slope to positive (to ensure
                                          rising edge is displayed).
80  OUTPUT 70905;"TRIG:LEV 0.75"          Trigger level to 0.75 volts (one-
                                          half the expected input).
90  OUTPUT 70905;"CONF:VOLT:RTIM 20,80 (@INP3)"
                                          Configure input 3 for rise time
                                          measurement, with thresholds of
                                          20/80%.
100 OUTPUT 70905;"READ:VOLT:RTIM?"        Read rise time measurement
                                          results.
110 ENTER 70905;Results                   Enter measurement results.
120 IF Results<21E-9 THEN 160             Verify measurement results are
                                          less than 21 nsec.
130 PRINT "Measurement out of Spec"       Print measurement flag.
140 END                                   Terminate program.
```

Comments

Signal Connection. An input signal must be connected to the Oscilloscope before initiating a measurement, and remain connected until the measurement is terminated.

Probe Attenuation Factor. Probe attenuation (correction) factor should be entered before the range and offset parameters. Setting the probe attenuation factor does not change the input sensitivity, but does scale the vertical and trigger parameters to compensate for the entered factor.

Range. Both vertical (VOLTage) and horizontal (SWEep) RANGe parameters are specified for full scale axis, and not per division values.

Measurement Time. The time it takes to complete a measurement is dependent on the input signal, acquisition type, and completion criteria. Under certain circumstances, a single measurement can take many hours to complete. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Example: Manual Rise Time and Fall Time Measurement

This example uses the Oscilloscope module to measure the rise time and fall time of a signal connected to Input 3 using a 1MΩ 10:1 probe. The expected input is a 1.5 V clock at 1 MHz. The upper threshold value should be set to 80% and the lower threshold value should be set to 20% for both measurements.

Note

When measuring rise time, the leading (positive-going) edge of the waveform must be present. When measuring fall time, the trailing (negative-going) edge of the waveform must be present. In order to obtain accurate results (example signal is a clock-minimum rise/fall time):

- the sweep should be setup so the one complete cycle is maximized over the time base range, and
- the trigger should be set so both edges are present.

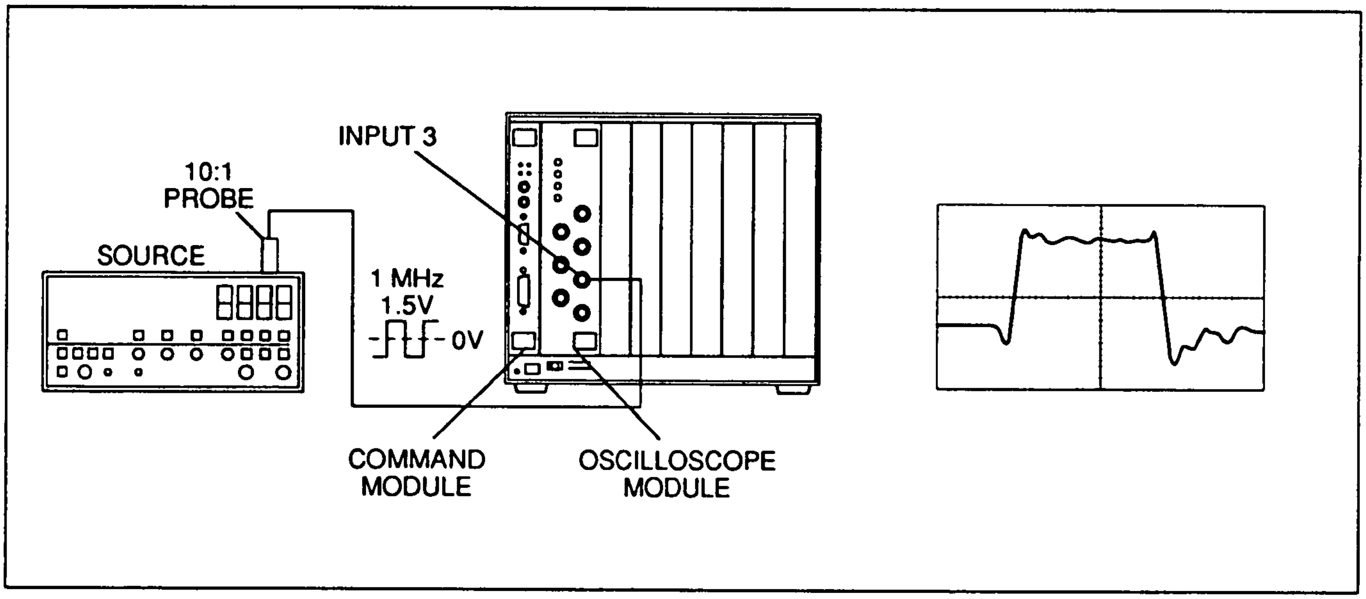


Figure 5-8. Example: Manual Rise Time and Fall Time Measurement

Execute:

```
10  OUTPUT 70905;"*CLS"                     Clear status.
20  OUTPUT 70905;"*RST"                     Resets the Oscilloscope to its
                                            default state (table 5-2).
30  OUTPUT 70905;"CORR3:AFAC 10"            Set input 3 probe attenuation
                                            factor at 10:1.
40  OUTPUT 70905;"VOLT3:RANG:PTP 2"         Set input 3 full scale vertical range
                                            to 2 volts (expected input is 1.5V).
50  OUTPUT 70905;"SWE:TIME:RANG 200E-9"     Set full scale horizontal range to
                                            200 ns (twice the expected
                                            period).
55  OUTPUT 70905;"TRIG:SOUR INP3"           Trigger source to input 3.
60  OUTPUT 70905;"TRIG:SLOP POS"            Trigger slope to positive.
70  OUTPUT 70905;"TRIG:LEV 0.75"            Trigger level to 0.75 volts (one-
                                            half the expected input).
80  OUTPUT 70905;"CONF:VOLT:RTIM 20,80, (@INP3)"
                                            Configure input 3 for rise time
                                            measurement, with thresholds of
                                            20/80%.
90  OUTPUT 70905;"CONF:VOLT:FTIM 20,80, (@INP3)"
                                            Configure input 3 for a fall time
                                            measurement, with thresholds of
                                            20/80%.
100 OUTPUT 70905;"READ:VOLT RTIM?"          Read rise time measurement
                                            results.
110 ENTER 70905;Rise_Results                Enter measurement results.
120 OUTPUT 70905;"READ:VOLT FTIM?"          Read fall time measurement
                                            results.
130 ENTER 70905;Fall_Results                Enter measurement results.
140 PRINT Rise_Results;Fall_Results         Print results.
150 END                                     Terminate program.
```

Comments

Signal Connection. An input signal must be connected to the Oscilloscope before initiating a measurement, and remain connected until the measurement is terminated.

Probe Attenuation Factor. Probe attenuation factor should be entered before the range and offset parameters. Setting the probe attenuation factor does not change the input sensitivity, but does scale the vertical and trigger parameters to compensate for the entered factor.

Range. Both vertical (VOLTage) and horizontal (SWEep) RANGe parameters are specified for full scale axis, and not per division values.

Measurement Time. The time it takes to complete a measurement is dependent on the input signal, acquisition type, and completion criteria. Under certain circumstances, a single measurement can take many hours to complete. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Measurement Parameters. Parameters for both rise time and fall time must be sent with each command, even if the parameters are identical for both measurements. After a measurement has been READ?, the parameter is returned to the default condition (10%/90%) for any other measurements.

Example: AC Voltage Measurement

This example uses the Oscilloscope module to measure the AC RMS voltage of a signal connected to Input 4. The expected input is 10Vrms (at 50Ω).at 1 kHz Because the input is greater than the maximum input voltage allowed, a 10:1 at 1MΩ probe is used. Because of the impedance mismatch, the approximate measurement result (at 50Ω) will be calculated.

Caution

MAXIMUM INPUT VOLTAGE. The maximum voltage that can be applied to any of the four input connectors is 5 Vrms at 50Ω or ±250 V (dc+peak ac<10 kHz) at 1MΩ using a 1:1 Probe.

Notes

For the example, the parameters are setup using autoscale, then adjusted as required. Disconnect any signal on inputs 1, 2, and 3.

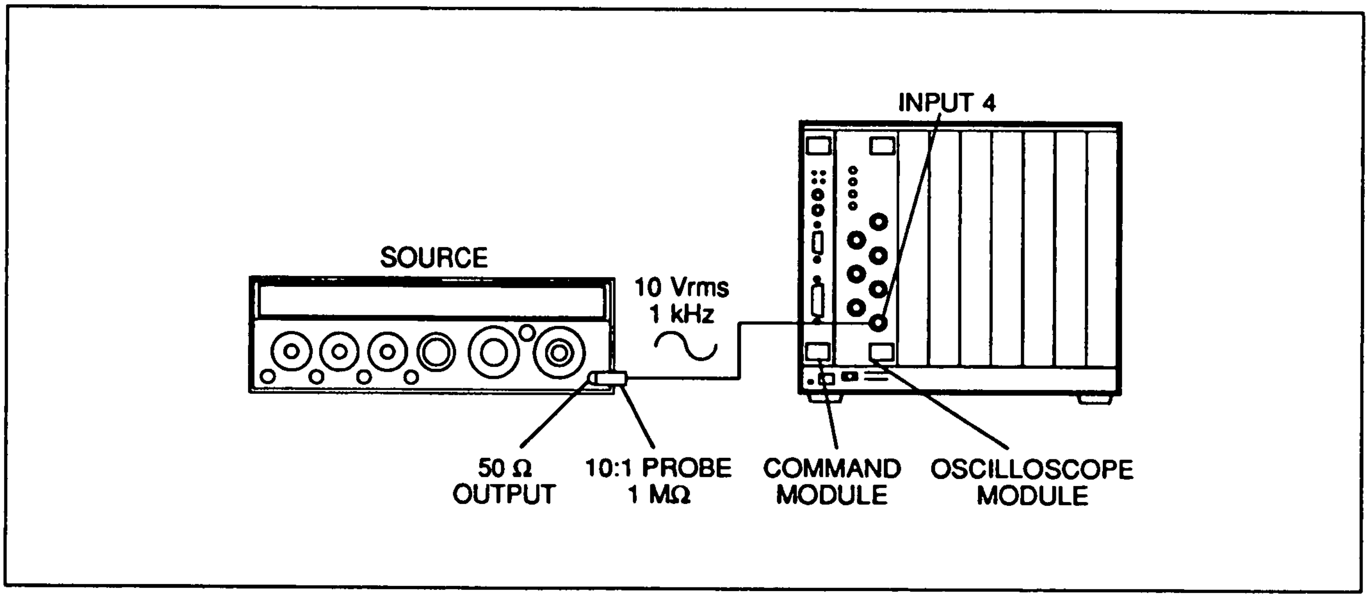


Figure 5-9. Example: AC Voltage Measurement

Execute:

```
 10  OUTPUT 70905;"*CLS"                     Clear status.
 20  OUTPUT 70905;"*RST"                     Resets the Oscilloscope to its
                                             default state (table 5-2).
 30  OUTPUT 70905;"CORR4:AFAC 10"            Set input 4 probe attenuation
                                             factor at 10:1.
 40  OUTPUT 70905;"SYST:AUT"                 Perform an Autoscale.
 50  OUTPUT 70905;"SWE:TIME:RANG 5E-3"       Set full scale horizontal range to
                                             5ms (to present five cycles).
 60  OUTPUT 70905;"MEAS:VOLT:AC (@INP4)"
                                             Configure input 4 for AC voltage
                                             measurement.
 70  ENTER 70905;Results                     Enter measurement results.
 80  LET Results_A = Results/2               Calculate the measurement
                                             results due to impedance
                                             mismatch.
 90  PRINT "Results_A"                       Print measurement results.
100  END                                     Terminate program.
```

Comments

Signal Connection. An input signal must be connected to the Oscilloscope before initiating a measurement, and remain connected until the measurement is terminated.

Probe Attenuation Factor. Probe attenuation factor should be entered before the range and offset parameters. Setting the probe attenuation factor does not change the input sensitivity, but does scale the vertical and trigger parameters to compensate for the entered factor.

Range. Both vertical (VOLTage) and horizontal (SWEep) RANGe parameters are specified for full scale axis, and not per division values.

Measurement Time. The time it takes to complete a measurement is dependent on the input signal, acquisition type, and completion criteria. Under certain circumstances, a single measurement can take many hours to complete. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Input Mismatch. Because the expected input is 10Vrms at 50Ω, and a 10:1 1MΩ probe is used, there is a mismatch at the input of the Oscilloscope. The returned measurement result is halved to compensate for the mismatch. While this will not provide an exact representation of the source output into 50Ω, the calculated result will be very close.

Example: Using a Backplane Trigger

The Oscilloscope is setup to perform an autoscale on a signal connected to input 1, then perform a frequency measurement when triggered from TTL Trigger bus line 5. The input signal is unknown.

Caution

MAXIMUM INPUT VOLTAGE. The maximum voltage that can be applied to any of the four input connectors is 5 Vrms at 50Ω or ±250 V (dc+peak ac<10 kHz) at 1MΩ using a 1:1 Probe.

Note

For the example, the parameters are setup using autoscale, then adjusted as required. Disconnect any signal on inputs 2, 3, and 4.

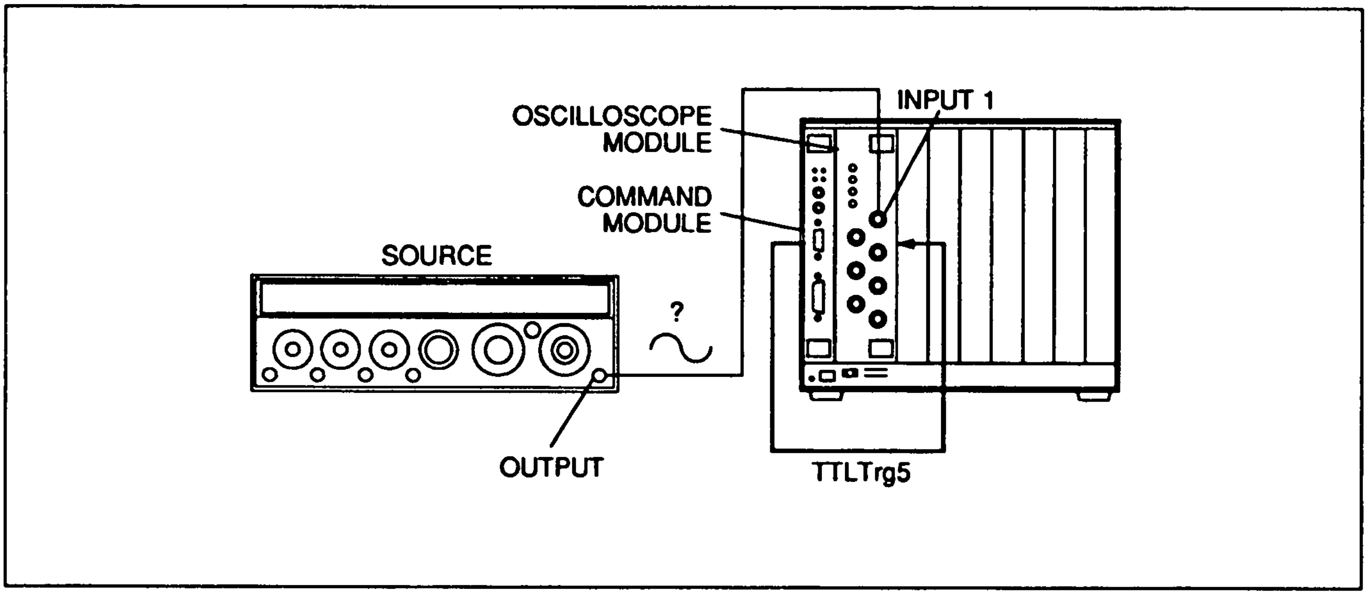


Figure 5-10. Example: Using the Backplane TTL Trigger

The example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP-IB select code of 7, primary address of 09, and secondary address of 00 for the Mainframe
- an HP Series 200/300 Computer with HP BASIC

Execute:

```
 10 OUTPUT 70905;"*CLS"                 Clear status.
 20 OUTPUT 70905;"*RST"                 Resets the Oscilloscope to its
                                        default state (table 3-2).
 30 OUTPUT 70905;"SYST:AUT"             Perform an autoscale.
 40 OUTPUT 70905;"TRIG:SOUR TTLT5"      Trigger on TTL trigger bus line 5.
 50 OUTPUT 70905;"INIT:CONT ON"         Set to initiate a measurement
                                        when a trigger is received.
 60 OUTPUT 70900;"OUTP:TTLT5:STAT ON"   Enables the command module to
                                        output a trigger on TTLTrg line 5.
 70 OUTPUT 70900;"OUTP:TTLT5:SOUR INT"  Set the command module trigger
                                        source.
 80 OUTPUT 70900;OUTP:TTLT5:IMM"        Cause a Trigger (TTL trigger bus
                                        lines).
 90 OUTPUT 70905;"CONF:VOLT:FREQ (@INP1)"
                                        Configure for a frequency
                                        measurement.
100 OUTPUT 70905;"FETC:VOLT:FREQ?"      Perform frequency measurement
                                        and return measurement results.
110 ENTER 70905;Results                 Enter measurement results.
120 PRINT Results                       Print measurement results.
130 END                                 Terminate program.
```

Comments

Multiple Signal Connection using Autoscale. When more than one Oscilloscope input is connected to a signal source, initiating autoscale will setup the Oscilloscope controls using the signal connected closest to input 1. If a signal is not found on input 1 then input 2 is used, if a signal is not found on input 2 then input 3 is used, etc.

Triggering TTL Trigger line 5. Any instrument in the mainframe can send a trigger on TTLTrg line 5 using the OUTPut:TTLTrg command. The example uses the Control Module to send the trigger after receiving the *TRG command.

After Triggering. After the trigger is received, the frequency measurement is initiated, and the results are returned.

Measurement Time. The time it takes to complete a measurement is dependent on the input signal, acquisition type, and completion criteria. Under certain circumstances, a single measurement can take many hours to complete. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

Recalling and Saving States

This section contains information about saving and recalling current Oscilloscope module states.

Storing States

The *SAV <*numeric_state*> command saves the current instrument state. The state number (1-4) is specified in the <*numeric_state*> parameter. All of the Oscilloscope and measurement setup parameters are saved.

Recalling States

The *RCL <*numeric_state*> command recalls a previously saved or existing state.

- Enter the number 0 in the <*numeric_state*> parameter to recall the configuration prior to executing the AUToscale or *RCL commands.
- Enter the number (1-4) in the <*numeric_state*> parameter of the desired saved state. If *SAV was not previously executed using the selected number, the Oscilloscope module will generate an error.

Recalling and Storing Waveforms

This section contains information about recalling and storing current Oscilloscope module waveforms. These controls allow for:

- Storing
- Turning on input connectors
- Turning off input connectors
- Erasing
- Merging

Note

The following discussion is intended to provide only a brief overview of the necessary commands required for basic operation. For a complete list and description of these commands, see Chapter 6, TRACe subsystem for more information.

Storing Waveforms

The TRACe[:DATA] command is used to save an active, previously stored, or calculated waveform in a non-volatile waveform memory location. The following waveforms are available for viewing:

INPut 1-4 - active waveform from input 1-4

WMEMory 1-4 - stored in waveform memory (non-volatile)

MATH 1-2 - calculated waveform (+,–,x)

You store a waveform with the following command:

```
TRAC xxx,yyyy        (xxx = memory destination WMEM 1-4 and
                     yyyy = source waveform INP1-4, WMEM 1-
                     4, or MATH1-2)
```

Note

Whatever is present at the specified source (waveform, baseline, etc) is what will be saved in the specified memory destination.

Turning Inputs ON/OFF

The [SENSe:]INPut[:STATe] command is used to enable or disable an input waveform. You select an input waveform with the following command:

```
INPn xxx             (n = input number and xxx = ON or OFF)
```

Note

All unused inputs should be blanked. See Appendix C, Optimizing Measurements, for additional information.

Erasing Waveforms

The TRACe[:DATA] command is used to erase the contents of pixel memory. You erase pixel memory with the following command:

```
TRAC xxx,0          (xxx = PMEM 1-2)
```

Merging Waveforms

The TRACe[:DATA] command is used to merge the contents of pixel memory 0 with the current contents of pixel memory 1 or 2. You merge pixel memories with the following command:

```
TRAC xxx,PMEM0      (xxx = PMEM 1-2)
```

Querying the Oscilloscope

This section summarizes the query commands you can use to determine the configuration or state of the Oscilloscope. All commands end with the "?" which puts the data into the output buffer where you can retrieve it to your computer. See Chapter 4 for more information.

Unless otherwise specified, <n> is the input number (1-4).

Query	Description
CALCulate	Calculate subsystem queries
:DATA?	Calculation data (last executed)
:MATH<n>:STATe?	Math state, where n is the math number
CALibration	CALibration subsystem queries
:REPort? INPut<n>	Calibration report, where n is the input number
:TNULl?	Current time null values for input pairs 1-2, 1-3, and 1-4
CONFigure?	Last configured measurement
FETCh[[:SCALar]:VOLTage[:<xxx>]]?	Configured measurement results, where xxx is the measurement
FORMat	Format subsystem queries
[:DATA]?	Waveform data format selection
INITiate	Initiate subsystem queries
:CONTinuous?	Initiate continuous state
MEASure[:SCALar]:VOLTage	Measure subsystem queries
:AC? [(@yyyn)]	Measure AC RMS voltage and return results, where yyyn is the source
:AMPLitude? [(@yyyn)]	Measure voltage amplitude and return results, where yyyn is the source
:[DC]? [(@yyyn)]	Measure DC voltage and return results, where yyyn is the source
:DCYCle? [<xxx>][(@yyyn)]	Measure duty cycle and return results, where yyyn is the source and xxx is the reference
:FALL	
:OVERshoot? [(@yyyn)]	Measure overshoot (falling edge) and return results, where yyyn is the source
:PREShoot? [(@yyyn)]	Measure preshoot and return results, where yyyn is the source
:TIME? [<xxx>] [(@yyyn)]	Measure fall time and return results, where yyyn is the source and xxx are the upper and lower thresholds
:FREQuency? [(@yyyn)]	Measure frequency and return results, where yyyn is the source
:FTIMe? [<xxx>] [(@yyyn)]	Measure fall time and return results, where yyyn is the source and xxx are the upper and lower thresholds
:HIGH? [(@yyyn)]	Measure high voltage and return results, where yyyn is the source

Querying the Oscilloscope — Continued

Unless otherwise specified, <n> is the input number (1-4).

Query	Description
MEASure[:SCALar]:VOLTage	
:LOW? [(@yyyn)]	Measure low voltage and return results, where yyyn is the source
:MAXimum? [(@yyyn)]	Measure maximum voltage and return results, where yyyn is the source
:MINimum? [(@yyyn)]	Measure minimum voltage and return results, where yyyn is the source
:NDUTycycle? [<xxx>][(@yyyn)]	Measure negative duty cycle and return results, where yyyn is the source and xxx is the reference
:NWIDth? [<xxx>][(@yyyn)]	Measure negative pulse width and return results, where yyyn is the source and xxx is the reference
:PDUTycycle? [<xxx>][(@yyyn)]	Measure positive duty cycle and return results, where yyyn is the source and xxx is the reference
:PERiod? [(@yyyn)]	Measure period and return results, where yyyn is the source
:PWIDth? [<xxx>][(@yyyn)]	Measure positive pulse width and return results, where yyyn is the source and xxx is the reference
:RISE	
:OVERshoot? [(@yyyn)]	Measure overshoot (rising edge) and return results, where yyyn is the source
:PREShoot? [(@yyyn)]	Measure preshoot (rising edge) and return results, where yyyn is the source
:TIME? [<*xxx*>]] [(@yyyn)]	Measure rise time and return results, where yyyn is the source and xxx are the upper and lower thresholds
:RTIMe [<*xxx*>]] [(@yyyn)]	Measure rise time and return results, where yyyn is the source and xxx are the upper and lower thresholds
:TMAXimum? [(@yyyn)]	Return time at maximum voltage, where yyyn is the source
:TMINimum? [(@yyyn)]	Return time at minimum voltage, where yyyn is the source
MEMory	Memory subsystem queries
:VME	
:ADDRess?	Memory address selected
:MEASure	
:ADDRess?	Memory measurement address selected
:STATe?	Memory measure state
:SIZE?	Memory size
:STATe?	Overall VME memory state

Querying the Oscilloscope — Continued

Unless otherwise specified, <n> is the input number (1-4).

Query	Description
OUTPut	Output subsystem queries
:ECLTrg<xxxx>[:STATe]?	ECL trigger bus line state, where xxxx is line number
:EXTernal[:STATe]?	Front panel trigger connector state
[:STATe]?	Overall output state
:TTLTrg<xxxx>[:STATe]?	TTL trigger bus line state, where xxxx is line number
READ[[:SCALar]:VOLTage[:<xxx>]]?	Perform a configured measurement and return results, where xxx is the measurement
[SENSe:]	Sense subsystem queries
AVERage	Average subsystem queries
:COUNt?	Acquisition count value
[:STATe]?	Acquisition type (average)
:TYPE?	Acquisition type (scalar and envelope)
CORRection<n>	Correction subsystem, where n is the input number
:AFACtor?	Input's probe attenuation
INPut<n>	Input subsystem, where n is the input number
:COUPling?	Input's coupling
:FILTer	
[:LPASs][:STATe]?	Input's low pass filter state
:HPASs[:STATe]?	Input's high pass filter state
:IMPedance?	Input's impedance value
[:STATe]?	Input's state
SWEep	Sweep subsystem
:POINts?	Acquisition points value
:COMPlete?	Acquisition complete value
:TIME	
:CENTer?	Sweep center value
:DELay?	Time base delay value
:LINK?	Delay reference selection
:RANGe?	Time base full scale range value
:SPAN?	Time base full scale range value
:STARt?	Sweep start value
:STOP?	Sweep stop value
VOLTage<n>	Voltage subsystem, where n is the input number
:RANGe	
:LOWer?	Input's lower range value
:OFFSet?	Input's offset value
[:PTPeak]?	Input's full scale range value
:UPPer?	Input's upper range value

Querying the Oscilloscope — Continued

Unless otherwise specified, <n> is the input number (1-4).

Query	Description
STATus	Status subsystem commands
:OPERation?	Operation event register value (trigger)
:QUEStionable?	Questionable event register value
:CALibration?	Calibration event register value
:INPut<*n*>?	Input's event register value
:AD?	Input's AD event register value
:DELay?	Input's delay event register value
:GAIN?	Input's gain event register value
:HYSTeresis?	Input's hysteresis event register value
:LTRigger?	Input's logic trigger event register value
:OFFSet?	Input's offset event register value
:TNULl?	Input's time null event register value
:TRIGger?	Input's trigger event register value
:DCALibration?	Default calibration event register value
:PROBe?	Probe event register value
:TEST?	Test event register value
:ACQuisition?	Acquisition test event register value
:AD?	Acquisition test AD event register value
:ATRigger?	Acquisition test analog trigger event register value
:DA?	Acquisition test DA event register value
:LTRigger?	Acquisition test logic trigger event register value
:TIMebase?	Acquisition test time base event register value
:INTerpolator?	Time base interpolator event register value
:RAM?	RAM test event register value
:ACQuisition?	Acquisition RAM test event register value
:DISPlay?	Display RAM test event register value
:NVOLatile?	Non-volatile RAM test event register value
:SYSTem?	System RAM test event register value
:ROM?	ROM test event register value
:NPRotect?	Non-protected RAM test event register value
:SYSTem?	System ROM test event register value
SYSTem	System subsystem queries
:ERRor?	Error number and message
:LANGuage?	Programming language selected
:SET?	Setup data (block)
:VERSion?	SCPI revision (date and number)
TRACe	
[:DATA]? <*xxx*>	Waveform data (block), where xxx is the source
:POINts? <*xxx*>	Waveform data points value, where xxx is the source
:PREamble? <*xxx*>	Preamble (block), where xxx is the source
TRIGger	Trigger subsystem queries
:ECOunt?	Trigger holdoff value (events)
:HYSTeresis?	Trigger noise reject selected
:LEVel?	Trigger level value
:SLOPe?	Trigger slope selected
:SOURce?	Trigger source selected

Contents

Chapter 6 — Oscilloscope SCPI Command Reference

6

Oscilloscope SCPI Command Reference

Using This Chapter

This chapter describes Standard Commands for Programmable Instruments (SCPI) commands and summarizes IEEE 488.2 Common (*) commands applicable to the Oscilloscope module.

See the HP E1405 User's Guide for additional information on SCPI and common commands. This chapter contains the following sections:

Command Types

Commands are separated into two types: IEEE 488.2 Common Commands and Standard Commands for Programmable Instruments (SCPI) Commands.

Common Command Format

The IEEE 488.2 standard defines the Common Commands that perform functions like reset, self-test, status byte query, etc. Common commands are four or five characters in length, always begin with the asterisk character (*), and may include one or more parameters. The command keyword is separated from the first parameter by a space character. Some examples of Common Commands are shown below:

```
*RST        *CLS        *STB?
```

SCPI Command Format

The SCPI commands perform functions like setting parameters, performing measurements, querying instrument states, and retrieving data. A subsystem command structure is a hierarchical structure that usually consists of a top level (or root) command, one or more lower level sub commands, and their parameters. The following example shows part of a typical subsystem:

```
[SENSe:]
        INPut<number>
                :COUPling  AC|DC
                :IMPedance?
```

[SENSe:] is the root command, INPut is the second level sub command with *<number>* as a parameter, and :COUPling and :IMPedance? are third level commands/queries with AC|DC as a parameter.

Command Separator

A colon (:) always separates one command from the next lower level command as shown below:

```
[SENSe:]INPut<number>:IMPedance?
```

Colons separate the root command from the second level command ([SENSe:]INPut), and the second level from the third level query (INPut<*number*>:IMPedance?).

Abbreviated Commands

The command syntax shows most commands as a mix of upper and lower case letters. The upper case letters indicate the abbreviated spelling for the command. For shorter program lines, send only the abbreviated form. For better program readability, you may send the entire command. The instrument will only accept either the abbreviated form or the entire command.

For example, if the command syntax shows IMPedance?, then IMP? and IMPEDANCE? are both acceptable forms. Other forms of IMPedance?, such as IM? will generate an error. You may use upper or lower case letters. Therefore, IMPEDANCE? and IMPeDaNcE? are acceptable.

Implied Commands

Implied commands are those which appear in square brackets ([ ]) in the command syntax. (Note that the brackets are not part of the command and are not sent to the instrument.) Suppose you send a root level and second level command, but do not send the third level implied command. In this case, the instrument assumes you intend to use the implied command and it responds as if you had sent it. Examine the portion of the [SENSe:] subsystem shown below:

```
[SENSe:]
        INPut<number>
               :COUPling AC|DC
               :IMPedance?
```

The first level command [SENSe:] is an implied command. To query the instrument's input 1 impedance selection, you can send either of the following command statements:

```
[SENS:]INP1:IMP?   or   INP1:IMP?
```

Parameters

Parameter Types. The following table contains explanations and examples of parameter types you might see later in this chapter.

Parameter Type	Explanations and Examples
Numeric	Accepts all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation. 123 or 1.23E2; –123 or –1.23E2; .123, 1.23E–1, or 1.23000E–01. Accepts all commonly used suffixes with decimal representations of numbers including optional signs, and decimal points. .123S or 123MS; 1234OHM or 1.234KOHM. Voltage = "UV" for E–6, "MV" for E-3, "V" for E0, "KV" for E3. Percent = "PCT". Ohms = "OHM", "KOHM" for E3, "MOHM" for E6. Frequency = "HZ" for E0, "KHZ" for E3, "MHZ" for E6, "GHZ" for E9. Time = "PS" for E–12, "NS" for E-9, "US" for E–6, "MS" for E-3, "S" for E0. Special cases include MIN and MAX. MIN (selects minimum value available), and MAX (selects maximum value available).
Boolean	Represents a single binary condition that is either true or false. 1 or ON; 0 or OFF
Block	Definite block program data format specified in IEEE 488.2.
Discrete	Selects from a finite number of values. These parameters use mnemonics to represent each valid setting. An example is the [SENSe:]INPut<n> Coupling *<mode>* command where *<mode>* can be AC or DC.

Optional Parameters. Parameters shown within square brackets ([ ]) are optional parameters. (Note that the brackets are not part of the command and are not sent to the instrument.) If you do not specify a value for an optional parameter, the instrument chooses a default value. For example, consider the MEM:VME:ADDR? [<MIN | MAX>] command. If you send the command without specifying a parameter, the present external VME memory address is returned. If you send the MIN parameter, the command returns the lowest address available (2097152). If you send the MAX parameter, the command returns the maximum address available (14647294). Be sure to place a space between the command and the parameter.

Parameters Out of Range - Set to Limit. If an out of range parameter is automatically adjusted to an acceptable value, bit 10 in the STATus:QUEStionable register will be set true (1). For example, if CORRection1:AFACtor 0.1 is entered, the value is set to 0.9 (lowest available setting) and bit 10 is set to "1".

Linking Commands

Linking IEEE 488.2 Common Commands with SCPI Commands. Use a semicolon between the commands. For example:

```
*CLS;*RST;CAL:REP? INP1
```

Linking Multiple SCPI Commands. Use both a semicolon and a colon between the commands. For example:

```
INP1:COUP AC;:SYST:ERR?
```

SCPI also allows several commands within the same subsystem to be linked with a semicolon. For example:

```
CAL:SCAL:VERT;:CAL:SCAL:BCAL
```

or

```
CAL:SCAL:VERT;BCAL
```

SCPI Command Reference

This section describes the Standard Commands for Programmable Instruments (SCPI) commands for the Oscilloscope module. Commands are listed alphabetically by subsystem and also alphabetically within each subsystem. Command guides are printed in the top margin of each page. The left guide indicates the first command listed on that page. The right guide indicates the last command listed on that page. Where only a single command appears on a page, the left and right guides will be the same.

ABORt

The ABORt command is used to stop acquiring data for the active waveform.

Subsystem Syntax

```
ABORt
```

Example **Stop acquiring data**

```
ABOR                      Stop acquiring data
```

Comments

- **Restart Acquiring Data:** Use the INITiate command to start data acquisition.
- **Related Commands:** INITiate:CONTinuous.
- ***RST Condition:** After a *RST, the instrument acts as though an ABORt has occurred.

CALCulate

The CALCulate command subsystem defines three functions that use signals acquired on INPuts 1 to 4 or stored in WMEMories 1-4 as operands to create altered waveforms. The selected input is enabled when defined as an operand. Two different functions (MATH1 and 2) can be specified. The results are read using the DATA? query, and can be transferred into waveform memory using the TRACe[:DATA] command.

CALCulate[:STATe] acts like the master switch for the CALCulate subsystem. If the MATH 1 or 2 states are on, a math function will ONLY occur when the CALCulate[:STATe] is set to ON.

Subsystem Syntax

```
CALCulate
    :DATA?
    :MATH<number>
        [:EXPRession] <function>
            :STATe <mode>
            :STATe?
        :STATe <mode>
        :STATe?
```

:DATA?

CALCulate:DATA? returns the calculated results of the last math operation performed. The data is sent to the output buffer.

Example **Query results of last math function**

`dimension statement`	*String for data*
`CALC:STAT ON`	*Enable the calculate subsystem*
`CALC:MATH1:STAT ON`	*Enable math 1 state*
`CALC:MATH1 (INP1-INP2)`	*Subtract signal present on input 2 from signal present on input 1, retain as math 1*
`CALC:DATA?`	*Query instrument to return results*
`enter statement`	*Enter math results into computer*

Comments

- **Reading MATH 1 or 2 Results.** The results of the last function performed are read using the DATA? query. Math 1 or 2 results can be selectively read into wave memory using the TRACe[:DATA]? command.
- **Returned Format:** IEEE definite block format.
- **Related Commands:** TRACe[:DATA], FORMat[:DATA].

:MATH[:EXPRession]

CALCulate:MATH<*number*>[:EXPRession] <*function*> is used to algebraically sum (+), subtract (–), or multiply (*) two defined operands (sources). Results are retained in the MATH *number* specified.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 or 2	none
function	discrete	(<*source*> + <*source*>)| (<*source*> – <*source*>)| (<*source*> * <*source*>)	none
source	discrete	INPut*n* (*n*=1 to 4) WMEMory*n* (*n*=1 to 4)	

Example

Algebraically sum input 1 with input 2 and retain results in math1

```
CALC:MATH1   (INP1-INP2)
```
Subtract signal present on input *2 from signal present on* input *1, retain as math1*

Comments

- **Performing Math Functions:** MATH:STATe must be ON before calculations can be performed.
- **Math Number:** Used to specify where math results are retained. Resulting waveform can be read using the CALCulate:DATA? query.
- **Specifying Function:** Three choices are used to specify the source(s) and math function to be performed. Parentheses "( )" are used to specify individual functions.

  (***source*** **–** ***source***) is used to algebraically subtract two defined operands.

  (***source*** **+** ***source***) is used to algebraically add two defined operands.

  (***source*** ***** ***source***) is used to algebraically multiply two defined operands.
- **Specifying Source:** Inputs 1-4 and/or waveform memories 1-4 can be defined as the operands for the math function.
- **Related Commands:** CALCulate:DATA?,CALCulate:STATe, CALCulate:MATH:STATe.
- ***RST Condition:** MATH1 and 2 default to INPut1+INPut1.

:MATH:STATe

CALCulate:MATH<*number*>:STATe <*mode*> enables or disables the specified math function.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 or 2	none
mode	boolean	OFF|0|ON|1	none

Example

Enabling math 1 function

```
CALC:MATH1:STATE  ON        Enable math function 1
```

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- ***RST Condition:** Both MATH1 and 2 default to OFF.

:MATH:STATe?

CALCulate:MATH<*number*>:STATe? returns a value to show whether use of the math function is enabled (ON) or disabled (OFF). The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 or 2	none

Example

Querying the math function 1 state

```
dimension  statement   String for data
CALC:MATH1:STAT  ON    Enable math function 1
CALC:MATH1:STAT?       Query instrument to return math
                       function state
enter  statement       Enter value into computer
```

:STATe

CALCulate:STATe *<mode>* enables or disables the CALCulate subsystem. *mode* enables (ON | 1) or disables (OFF | 0) all selected MATH (1-2) functions.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	ON|OFF|1|0	none

Example **Enabling math 1 functions**

```
CALC:MATH1:STAT  ON      Enable math 1
CALC:STAT  ON            Enable the calculate subsystem
```

Comments

- **Selecting Math Functions:** Use the CALCulate:MATH<n>:STATe command to enable a specific function. Use the CALCulate:STATe command to enable the subsystem.
- ***RST Condition:** Defaults to OFF.

:STATe?

CALCulate:STATe? queries the present state of the CALCulate subsystem. The query returns ON if the calculate subsystem is enabled or OFF if the calculate subsystem is disabled. The value is sent to the output buffer. See CALCulate:STATe command for more information.

Example **Query calculate subsystem state**

```
dimension  statement     String for data
CALC:STAT  ON            Enable the calculate subsystem
CALC:STAT?               Query instrument to return calculate
                         state
enter  statement         Enter value into computer
```

CALibration

The CALibration command subsystem contains commands to perform probe/self calibrations, and set input-to-input time nulls.

Subsystem Syntax

```
CALibration
    :PCALibration
        :ATTenuation
            :BCALibration
            :INPut<number>
        :TNULl
            :INP1TO<number> <time>
    :REPort? <input>
    :SCALibration
        :BCALibration
        :DCALibration
        :DELay <input>
        :DOUTput <level>
        :LTCalibration
        :TNULl <input_skew>
        :VERTical
    :TNULl <value1>,<value2>,<value3>
    :TNULl?
```

:PCALibration:ATTenuation:BCALibration

CALibration:PCALibration:ATTenuation:BCALibration performs an attenuation calibration on the input number specified by the CAL:PCAL:ATT:INP<n> command. Instrument calibrates input gain at the point connected to the DC Calibrator Output connector (probe, cable, etc). Probe attenuation is then calculated from the results, and a correction is automatically entered in the correct CORRection<*n*>:AFACtor setting.

Example **Perform attenuation calibration on input 4**

This example calibrates the input gain on input 4. For example, a 10:1 attenuator probe is connected to the DC Calibrator Output connector from the Input 4 connector.

`CAL:PCAL:ATT:INP4`	*Attenuation calibration input 4*
`pause`	*To connect probe to DC Calibrator Output from Input 4 connector*
`CAL:PCAL:ATT:BCAL`	*Perform attenuation calibration. Correction automatically stored in INP4:PROB*

Comments

- **Valid Calibration:** Input gain is corrected using calculated probe attenuation values from 0.9:1 to 250:1. If the measured results cause the calculated attenuation factor to be out of this range, an error will be generated.
- **Related Commands:** CAL:PCAL:ATT:INPut<n>, CORRection<n>:AFACtor.

:PCALibration:ATTenuation:INPut

CALibration:PCALibration:ATTenuation:INPut*<number>* selects the input number that will be calibrated when the CAL:PCAL:ATT:BCAL command is executed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example **Set attenuation calibration input to 4**

```
CAL:PCAL:ATT:INP4        Attenuation calibration input to 4
```

Comments

- **Related Commands:** CAL:PCAL:ATT:BCALibration.

:PCALibration:TNULl:INP1TO

CALibration:PCALibration:TNULl:IN1PTO*<number>* *<time>* is used to set the timing of inputs 2, 3, OR 4 to correspond with input 1. Use to eliminate any time discrepancies between inputs and minimize input to input skew variations. Use to manually adjust any differences in cable length.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number *time*	numeric numeric	2 to 4 –50NS to +70NS	none S

Example **Set time null from input 1 to 3 to 25 ns.**

```
CAL:PCAL:TNUL:INP1TO3   25E-9   Input 1 to 3 time null to 25
                                nsec
```

Comments

- **Query Time Null:** Use the CALibration:TNULl? query to return current time null settings.
- **CALibration:TNULl Command:** This command is similar to the CALibration:TNULl? query, except the three time null values can be entered separately.
- **Related Commands:** CALibration:TNULl.

:REPort?

CALibration:REPort? ***<input>*** is used to query the current calibration status of the instrument. Each input's status is queried separately. The data is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
input	numeric	INPut*n* (*n*=1 to 4)	none

Example

Query input 2 calibration results

```
dimension  statement    String to hold data
CAL:REPort?  INP2       Query input 2 calibration results
enter  statement        Enter value into computer
```

Comments

- **Returned Format:** The calibration results are returned in the following format:

Input1	**A/D X, Gain X, Offset X, Hysteresis X, Trigger X, Delay X, Logic Trigger X**
Input2-4	**A/D X, Gain X, Offset X, Hysteresis X, Trigger X, Delay X, Time Null X**

Where X is "P"=Passed, "F"=Failed, "D"=Defaulted, "C"=Corrupted. If X prefixed by a "*", indicates a new ROM revision without a recalibration.

- **Related Commands:** CAL:SCAL:VERTical, CAL:SCAL:DCALibration, CAL:SCAL:DELay, CAL:SCAL:LTCalibration, CAL:SCAL:TNULl.

:SCALibration:BCALibration

CALibration:SCALibration:BCALibration is used to begin a self calibration routine. The routine that is performed is dependent on the SCALibration command configured prior to executing the BCALibration command.

Example

Begin a Logic Trigger Calibration

```
CAL:SCAL:LTC           Configure logic trigger calibration
CAL:SCAL:BCAL          Begin logic trigger calibration
```

Comments

- **Self Calibration:** If the BCALibration command is executed without first defining the SCALibration routine to be performed, a bit will be set in the STATus:QUEStionable register.
- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a SCALibration routine.
- **Related Commands:** CAL:SCAL:VERTical, CAL:SCAL:DCALibration, CAL:SCAL:DELay, CAL:SCAL:LTCalibration, CAL:SCAL:TNULl, CAL:TNULl, STATus:QUEStionable.

:SCALibration:DCALibration

CALibration:SCALibration:DCALibration is used to load "default" calibration data. Default calibration data is set at the factory and is dependent on the ROM revision currently installed. This command should only be used by service personnel. Procedures for performing this calibration are provided in the Service Manual.

Example **Overwrite all existing calibration data with default calibration data**

`CAL:SCAL:DCAL`	*Configure for a default calibration routine*
`CAL:SCAL:BCAL`	*Load default calibration data*

Comments

- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a default calibration routine.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?.

:SCALibration:DELay

CALibration:SCALibration:DELay ***<input>*** performs a delay calibration on all four inputs, one at a time. Each input must be connected to the AC Calibrator Output prior to executing the calibration routine for that input. The results are stored and used by the instrument to maintain measurement accuracy.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
input	numeric	INPut*n* (*n*=1 to 4)	none

Example **Chapter 5 contains an example on performing a delay calibration**

Comments

- **Calibration Results:** Delay calibration results can be reviewed using the CALibration:REPort? query.
- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a calibration routine.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?.

:SCALibration:DOUTput

CALibration:SCALibration:DOUTput *<level>* is used to set the output level of the DC Calibrator output connector to 0 volts (ZVOLt) or 5 volts (FVOLt).

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
level	discrete	ZVOLt|FVOLt	none

Example **Set DC calibrator output connector to 5 V**

```
CAL:SCAL:DOUT  FVOL
```
DC calibrator output to 5 V

Comments

- ***RST Condition:** Defaults to ZVOLt (0 volts).

:SCALibration:LTCalibration

CALibration:SCALibration:LTCalibration performs a logic trigger calibration. Input 1 must be connected to the AC Calibrator Output prior to executing the calibration routine. The results are stored and used by the instrument to maintain measurement accuracy.

Example **Chapter 5 contains an example on performing a logic trigger calibration**

Comments

- **Prior to Logic Trigger Calibration Execution:** Prior to executing the logic trigger calibration routine, the calibration results must be reviewed using the CALibration:REPort? query. All four input calibration results must indicate "P" before the logic trigger calibration can be executed.
- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a calibration routine.
- **Calibration Results:** Logic trigger calibration results can be reviewed using the CALibration:REPort? query.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?.

:SCALibration:TNULl

CALibration:SCALibration:TNULl *<input_skew>* performs a time null calibration on one set of inputs at a time. The results are stored and used by the instrument to maintain measurement accuracy.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
input_skew	discrete	INP1TO2|INP1TO3|INP1TO4	none

Example

Chapter 5 contains an example on performing a time null calibration

Comments

- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a calibration routine.
- **Calibration Results:** Time null calibration results can be reviewed using the CALibration:REPort? query.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?.

:SCALibration:VERTical

CALibration:SCALibration:VERTical performs a vertical calibration on all four inputs simultaneously. All inputs must be connected to the DC Calibrator Output prior to executing the calibration routine. The results are stored and used by the instrument to maintain measurement accuracy.

Example

Chapter 5 contains an example on performing a vertical calibration

Comments

- **Calibration Results:** Vertical calibration results can be reviewed using the CALibration:REPort? query.
- **Calibration Protect Switch:** The Calibration Protect switch must be set to the UNPROTECTED setting prior to performing a calibration routine.
- **Related Commands:** CALibration:SCALibration:BCALibration, CALibration:REPort?.

:TNULl

CALibration:TNULl ***<value_1>,<value_2>,<value_3>*** **is used to set the timing of inputs 2, 3, AND 4 to correspond with input 1. Used to eliminate any time discrepancies between inputs and minimize input to input skew variations. Use to manually adjust any differences in cable length.**

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	2 to 4	none
value_1	numeric	–50NS to +70NS	S
value_2	numeric	–50NS to +70NS	S
value_3	numeric	–50NS to +70NS	S

Example **Set time null to 25 ns from input 1 to 4**

```
CAL:TNUL  0,0,25E-9      Input 1 to 4 time null to 25 nsec
```

Comments

- **Query Time Null:** Use the CALibration:TNULl? query to return current time null settings.
- **CALibration:TNULl Command:** This command is similar to the CALibration:PCALibration:ATTenuation:TNULl INP1TO command, except all three time null values must be entered.
- **Related Commands:** CAL:PCAL:TNULl INP1TO.

:TNULl?

CALibration:TNULl? returns the currently selected time nulls (in seconds) for inputs 1 to 2, 1 to 3, and 1 to 4 respectively. The data is sent to the output buffer. See CALibration:TNULl command for more information.

Example **Querying time nulls**

```
CAL:TNUL  0,0,25E-9      Set input 1 to 4 time null to 25 nsec
CAL:TNUL?                Query instrument to return time nulls
enter  statement         Enter value into computer
```

Comments

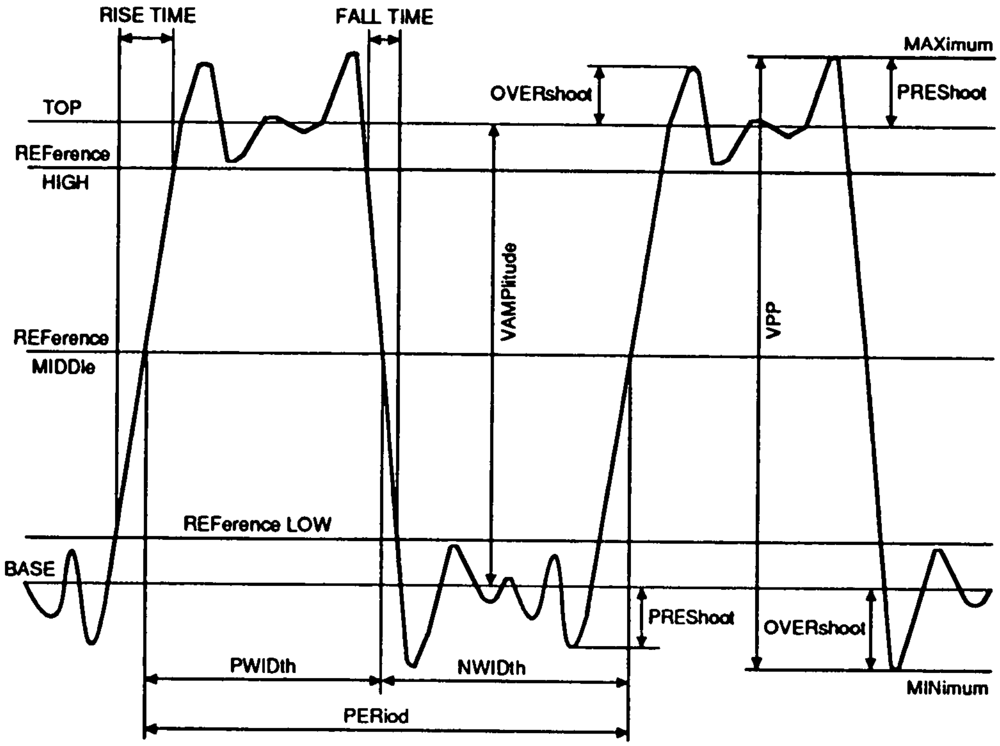
- **Related Commands:** CAL:SCAL:TNULl INP1TO, CAL:TNULl.

CONFigure

The CONFigure command subsystem sets up the instrument to perform a specified measurement, but does not perform the actual measurement. Use INITiate/FETCh[:*<function>*]?, or READ[:*<function>*]? to initiate the measurement and read the results.

All measurements can be performed using the following methods:

> The **CONFigure[:SCALar]:VOLTage** command only configures an input for a specific function, and **DOES NOT PERFORM** the measurement. Use additional commands/queries (READ[:*<function>*]?, or INIT/FETC[:*<function>*]?) to perform the measurement and read the result is necessary.
>
> The **MEASure** configures an input for a specific function, performs the measurement, and returns the results.
>
> > **[:SCALar]** specifies that a single value, not an array of readings, will be taken. **:VOLTage** specifies that the voltage characteristics of the signal will be measured.
>
> The illustration below shows the point(s) where measurements are taken.



Subsystem Syntax

```
CONFigure?
CONFigure[:SCALar]
    :VOLTage
        :AC [<channel_list>]
        :AMPLitude [<channel_list>]
        :[DC] [<channel_list>]
        :DCYCle [<reference>][<channel_list>]
        :FALL
            :OVERshoot [<channel_list>]
            :PREShoot [<channel_list>]
            :TIME [<lower_limit>[,<upper_limit>]] [<channel_list>]
        :FREQuency [<channel_list>]
        :FTIMe [<lower_limit>[,<upper_limit>]] [<channel_list>]
        :HIGH [<channel_list>]
        :LOW [<channel_list>]
        :MAXimum [<channel_list>]
        :MINimum [<channel_list>]
        :NDUTycycle [<reference>][<channel_list>]
        :NWIDth [<reference>][<channel_list>]
        :PDUTycycle [<reference>][<channel_list>]
        :PERiod [<channel_list>]
        :PWIDth [<reference>][<channel_list>]
        :RISE
            :OVERshoot [<channel_list>]
            :PREShoot [<channel_list>]
            :TIME [<lower_limit>[,<upper_limit>]] [<channel_list>]
        :RTIMe [<lower_limit>[,<upper_limit>]] [<channel_list>]
        :TMAXimum [<channel_list>]
        :TMINimum [<channel_list>]
```

CONFigure?

CONFigure? is used to return the last configured measurement.

Example **Configure input 1 for an AC RMS voltage measurement**

`dimension statement`	*String for data*
`CONF:VOLT:AC (@INP1)`	*Configure input 1 for an AC RMS Voltage measurement*
`CONF?`	*Instrument returns last configured measurement*
`enter statement`	*Enter value into computer*

Comments

- **Returned Results:** Returns the last configured measurement, including parameters and channel_list (e.g. "CONF:VOLT:AC (@INP1)").

:AC

CONFigure[:SCALar]:VOLTage:AC [<*channel_list*>] is used to configure the source specified by *channel_list* for an AC RMS voltage measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 1 for an AC RMS voltage measurement

```
CONF:VOLT:AC   (@INP1)
```
Configure input 1 for an AC RMS Voltage measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** The AC RMS Voltage measurement is made using the the first cycle present. If a complete cycle is not present, the AC RMS value of all data points is calculated.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the AC RMS voltage (with 0 volts as the reference) of the source specified.
- **Related Commands:** READ?, INITiate, FETCh?.

:AMPLItude

CONFigure[:SCALar]:VOLTage:AMPLitude [<*channel_list*>] is used to configure the source specified by *channel_list* for an Amplitude voltage measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure waveform memory 2 for an amplitude voltage measurement

```
CONF:VOLT:AMPL  (@WMEM2)
```

Configure waveform memory 2 for an Amplitude Voltage measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** Amplitude voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the amplitude voltage (with 0 volts as the reference) of the source specified.
- **Measurement Method:** The method the instrument uses to determine voltage amplitude is to measure HIGH and LOW, then calculate voltage amplitude as follows:

  voltage amplitude = HIGH – LOW
- **Related Commands:** READ?, INITiate, FETCh?.

[:DC]

CONFigure[:SCALar]:VOLTage[:DC] [<*channel_list*>] is used to configure the source specified by *channel_list* for a DC voltage measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 3 for a DC voltage measurement

```
CONF:VOLT  (@INP3)
```

Configure input 3 for a DC Voltage measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** DC Voltage measurement is made using the first cycle present. If a complete cycle is not present, all currently acquired data points are averaged.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the DC voltage (with 0 volts as the reference) of the source specified.
- **Related Commands:** READ?, INITiate, FETCh?.

:DCYCle

CONFigure[:SCALar]:VOLTage:DCYCle [<*reference*>][<*channel_list*>] is used to configure the source specified by *channel_list* for a Duty Cycle measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reference	numeric	0 to 100	PCT
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example **Configure input 4 for a duty cycle measurement at 40%**

```
CONF:VOLT:DCYC  40,(@INP4)
```

Configure input 4 for a Duty Cycle measurement at 40%

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Combinations of *reference and channel_list*.** The various combinations of *reference* and *channel_list* are entered as follows :

  <*reference*>,<*channel_list*> – when selecting both parameters

  <*channel_list*> – when selecting the channel list only (uses middle defaults value (50%)

  Specifying Reference: The measurement is performed at a specified level if the *reference* parameter is used. Values from 0 to 100% are accepted. *reference* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *reference* to 0.2 volts, send the following:

  ```
  CONF:VOLT:DCYC  0.2V,(@XXX)
  ```

- **Oscilloscope Setup:** In order to perform a Duty Cycle measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. Duty Cycle is determined at the specified reference level on the waveform.

- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the duty cycle of the source specified.
- **Measurement Method:** The method the instrument uses to determine duty cycle is to measure PWIDth and PERiod, then present duty cycle in percent as ratio of the positive pulse width to period as follows:

  duty cycle = + pulse width/period
- **DCYCle versus PDUTycycle:** CONFigure[:SCALar]:VOLTage:DCYCle command is identical to the CONFigure[:SCALar]:VOLTage:PDUTycycle command.
- **Related Commands:** READ?, INITiate, FETCh?, CONFigure[:SCALar]:VOLTage:PDUTycycle.

:FALL:OVERshoot

CONFigure[:SCALar]:VOLTage:FALL:OVERshoot [*<channel_list>*] is used to configure the source specified by *channel_list* for an Overshoot measurement on the falling edge of the waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure math function 2 for an overshoot measurement on the falling edge of the waveform

```
CONF:VOLT:FALL:OVER   (@MATH2)  Configure math 2 for an
                                overshoot measurement
                                on the falling edge
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform an Overshoot measurement, a minimum of one falling edge must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the overshoot of the falling edge (in percent) of the source specified.
- **Measurement Method:** The method the instrument uses to determine overshoot is to make three different voltage measurements of the first falling (negative-going) edge present, then calculate overshoot as follows:

  overshoot = ((LOW - MINimum)/AMPLitude) • 100
- **Related Commands:** READ?, INITiate, FETCh?.

:FALL:PREShoot

CONFigure[:SCALar]:VOLTage:FALL:PREShoot [<*channel_list*>] is used to configure the source specified by *channel_list* for a Preshoot measurement on the falling edge of the waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 3 for a preshoot measurement on the falling edge of the waveform

```
CONF:VOLT:FALL:PRES  (@INP3)
```

Configure input 3 for a preshoot measurement on the falling edge

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Preshoot measurement, a minimum of one falling edge must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the preshoot of the falling edge (in percent) of the source specified.
- **Measurement Method:** The method the instrument uses to determine preshoot is to make three different voltage measurements of the first falling (negative-going) edge present, then calculate preshoot as follows:

  preshoot = ((HIGH - MAXimum)/AMPLitude) • 100
- **Related Commands:** READ?, INITiate, FETCh?.

:FALL:TIME

CONFigure[:SCALar]:VOLTage:FALL:TIME [*lower_limit*[,*upper_limit*]][<*channel_list*>] is used to configure the source specified by ***channel_list*** for a Fall Time measurement. ***lower_limit*** defines the lower measurement threshold. ***upper_limit*** defines the upper measurement threshold.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none
lower_limit	numeric	–25.00 to +125.0	PCT
upper_limit	numeric	–25.00 to +125.0	PCT

Example

Configure input 1 for a fall time measurement at 10% lower threshold limit and 90% upper threshold limit (defaults)

```
CONF:VOLT:FALL:TIME
```
Configure input 1 for a Fall Time measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Combinations of *upper_limit/lower_limit/channel_list*.** The various combinations of *upper_limit*, *lower_limit*, and *channel_list* are entered as follows :

  <*lower_limit*>,<*upper_limit*>,<*channel_list*> – when selecting all parameters

  <*lower_limit*>,<*channel_list*> – when selecting the lower limit and channel list (uses upper limit default)

  <*channel_list*> – when selecting the channel list only (uses upper and lower limit defaults)
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.

- **Selecting Limits:** The upper and lower limits for the fall time measurement can be defined by the user as follows (units of percent and voltage cannot be specified together):

  Default: The measurement is performed at 10%/90% threshold levels if the *lower_limit* and *upper_limit* parameters are omitted.

  Specifying Lower Limit: The measurement is performed at a specified lower threshold if the *lower_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *lower_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *lower_limit* to 0.2 volts, send the following:

  ```
  CONF:VOLT:FALL:TIME   0.2V,XXXX,(@XXX)
  ```

  Specifying Upper Limit: The measurement is performed at a specified upper threshold if the *upper_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *upper_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *upper_limit* to 4.5 volts, send the following:

  ```
  CONF:VOLT:FALL:TIME   XXXX,4.5V,(@XXX)
  ```

- **Oscilloscope Setup:** In order to perform a Fall Time measurement, the falling edge of the waveform must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. For best measurement accuracy, set the sweep speed as fast as possible. If the horizontal scaling is questionable when performing a Fall Time measurement, an error will be generated.

- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the fall time (in seconds) at the specified upper and lower limits of the selected source.

- **Measurement Method:** The method the instrument uses to determine fall time is to measure the time at the upper threshold then the lower threshold of the first falling (negative-going) edge present. Measurement results are used to calculate fall time as follows:

  fall time = lower threshold time – upper threshold time

- **FALL:TIME versus FTIMe:** CONFigure[:SCALar]:VOLTage:FALL:TIME command is identical to the CONFigure[:SCALar]:VOLTage:FTIMe command.

- **Related Commands:** READ?, INITiate, FETCh?, CONFigure[:SCALar]:VOLTAGE:FTIMe.

- ***RST Conditions:** Lower limit defaults to 10% and upper limit defaults to 90%.

:FREQuency

CONFigure[:SCALar]:VOLTage:FREQuency [<*channel_list*>] is used to configure the source specified by *channel_list* for a Frequency measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure waveform memory 4 for a frequency measurement

```
CONF:VOLT:FREQ   (@WMEM4)
```

Configure waveform memory 4 for a Frequency measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (*@source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Frequency measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the frequency (in hertz) of the source specified.
- **Measurement Method:** The method the instrument uses to determine frequency is to measure the time of the first complete cycle, then calculate frequency as follows:

  If first edge of waveform is rising, then:
  frequency = 1/(time at second rising edge – time at first rising edge)

  If first edge of waveform is falling, then:
  frequency = 1/(time at second falling edge – time at first falling edge)
- **Related Commands:** READ?, INITiate, FETCh?.

:FTIMe

CONFigure[:SCALar]:VOLTage:FTIMe [*lower_limit*[,*upper_limit*]][<*channel_list*>] is used to configure the source specified by *channel_list* for a Fall Time measurement. *lower_limit* defines the lower measurement threshold. *upper_limit* defines the upper measurement threshold.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none
lower_limit	numeric	–25.00 to +125.0	PCT
upper_limit	numeric	–25.00 to +125.0	PCT

Example

Configure waveform memory 1 for a fall time measurement at 20% lower threshold limit and 70% upper threshold limit

```
CONF:VOLT:FTIM  20,70,(@WMEM1)
```

Configure waveform memory 1 for a Fall Time measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Combinations of *upper_limit/lower_limit/channel_list.*** The various combinations of *upper_limit, lower_limit,* and *channel_list* are entered as follows :

  <lower_limit>,<upper_limit>,<channel_list> – when selecting all parameters

  <lower_limit>,<channel_list> – when selecting the lower limit and channel list (uses upper limit default)

  <channel_list> – when selecting the channel list only (uses upper and lower limit defaults)
- **Selecting Limits:** The upper and lower limits for the fall time measurement can be defined by the user as follows (units of percent and voltage cannot be specified together):

  Default: The measurement is performed at 10%/90% threshold levels if the *lower_limit* and *upper_limit* parameters are omitted.

  Specifying Lower Limit: The measurement is performed at a specified lower threshold if the *lower_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *lower_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *lower_limit* to 0.2 volts, send the following:

  ```
  CONF:VOLT:FTIM  0.2V,XXXX,(@X)
  ```

Specifying Upper Limit: The measurement is performed at a specified upper threshold if the *upper_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *upper_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *upper_limit* to 4.5 volts, send the following:

```
CONF:VOLT:FTIM   XXXX,4.5V,(@X)
```

- **Selecting *channel_list*.** *channel_list* has the form (*@source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Fall Time measurement, the falling edge of the waveform must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. For best measurement accuracy, set the sweep speed as fast as possible. If the horizontal scaling is questionable when performing a Fall Time measurement, an error will be generated.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the fall time (in seconds) at the specified upper and lower limits of the selected source.
- **Measurement Method:** The method the instrument uses to determine fall time is to measure the time at the upper threshold then the lower threshold of the first falling (negative-going) edge present. Measurement results are used to calculate fall time as follows:

  fall time = lower threshold time – upper threshold time

- **FALL:TIME versus FTIMe:** CONFigure[:SCALar]:VOLTage:FTIMe command is identical to the CONFigure[:SCALar]:VOLTage:FALL:TIME command.
- **Related Commands:** READ?, INITiate, FETCh?, CONFigure[:SCALar]:VOLTAGE:FALL:TIME.
- ***RST Conditions:** Lower limit defaults to 10% and upper limit defaults to 90%.

:HIGH

CONFigure[:SCALar]:VOLTage:HIGH [<*channel_list*>] is used to configure the source specified by *channel_list* for a High voltage measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example **Configure input 2 for a high voltage measurement**

```
CONF:VOLT:HIGH   (@INP2)
```
Configure input 2 for a High Voltage measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** High Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the voltage value (with 0 volts as the reference) at the highest point of the source specified.
- **Related Commands:** READ?, INITiate, FETCh?.

:LOW **CONFigure[:SCALar]:VOLTage:LOW [<*channel_list*>]** is used to configure the source specified by *channel_list* for a Low voltage measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example **Configure input 3 for a low voltage measurement**

```
CONF:VOLT:LOW   (@INP3)
```

Configure input 3 for a Low Voltage measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** Low Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the voltage value (with 0 volts as the reference) at the lowest point of the source specified.
- **Related Commands:** READ?, INITiate, FETCh?.

:MAXimum

CONFigure[:SCALar]:VOLTage:MAXimum [<*channel_list*>] is used to configure the source specified by *channel_list* for a Maximum voltage measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 4 for a maximum voltage measurement

```
CONF:VOLT:MAX  (@INP4)
```
Configure input 4 for a Maximum Voltage measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** Maximum Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the absolute maximum voltage (with 0 volts as the reference) of the source specified.
- **Related Commands:** READ?, INITiate, FETCh?.

:MINimum

CONFigure[:SCALar]:VOLTage:MINimum [<*channel_list*>] is used to configure the source specified by *channel_list* for a Minimum voltage measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure math function 1 for a minimum voltage measurement

```
CONF:VOLT:MIN  (@MATH1)
```
Configure math function 1 for a Minimum Voltage measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** Minimum Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the absolute minimum voltage (with 0 volts as the reference) of the source specified.
- **Related Commands:** READ?, INITiate, FETCh?.

:NDUTycycle

CONFigure[:SCALar]:VOLTage:NDUTycycle [<*reference*>][<*channel_list*>] is used to configure the source specified by *channel_list* for a Negative Duty Cycle measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reference	numeric	0 to 100	PCT
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 2 for a negative duty cycle measurement

```
CONF:VOLT:NDUT   (@INP2)
```
Configure input 2 for a Negative Duty Cycle measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Combinations of *reference and channel_list*.** The various combinations of *reference* and *channel_list* are entered as follows :

  <*reference*>,<*channel_list*> – when selecting both parameters

  <*channel_list*> – when selecting the channel list only (uses middle defaults value (50%)

  Specifying Reference: The measurement is performed at a specified level if the *reference* parameter is used. Values from 0 to 100% are accepted. *reference* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *reference* to 0.2 volts, send the following:

  ```
  CONF:VOLT:NDUT   0.2V, (@XXX)
  ```
- **Oscilloscope Setup:** In order to perform a Negative Duty Cycle measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. Negative Duty Cycle is determined at the specified reference level on the waveform.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the negative duty cycle of the source specified.
- **Measurement Method:** The method the instrument uses to determine negative duty cycle is to measure NWIDth and PERiod, then present duty cycle in percent as ratio of the negative pulse width to period as follows:

  negative duty cycle = – pulse width/period
- **Related Commands:** READ?, INITiate, FETCh?.

:NWIDth

CONFigure[:SCALar]:VOLTage:NWIDth [*<reference>*][*<channel_list>*] is used to configure the source specified by *channel_list* for a Negative Pulse Width measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reference	numeric	0 to 100	PCT
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 3 for a negative pulse width measurement

```
CONF:VOLT:NWID  (@INP3)
```
Configure input 3 for a Negative Pulse Width measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Combinations of *reference and channel_list.*** The various combinations of *reference* and *channel_list* are entered as follows :

  <reference>,<channel_list> – when selecting both parameters

  <channel_list> – when selecting the channel list only (uses middle defaults value (50%)

  Specifying Reference: The measurement is performed at a specified level if the *reference* parameter is used. Values from 0 to 100% are accepted. *reference* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *reference* to 0.2 volts, send the following:

  ```
  CONF:VOLT:NWID  0.2V,(@XXX)
  ```

- **Oscilloscope Setup:** In order to perform a Negative Pulse Width measurement, a minimum of one negative pulse must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. Negative Pulse Width is determined at the specified reference level on the waveform.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the negative pulse width (in seconds) of the source specified.

- **Measurement Method:** The method the instrument uses to determine pulse width is to measure the time of the first complete negative pulse, then calculate pulse width as follows:

  If first edge of waveform is rising, then:
  pulse width = time at second rising edge – time at first falling edge

  If first edge of waveform is falling, then:
  pulse width = time at first rising edge – time at first falling edge

- **Related Commands:** READ?, INITiate, FETCh?.

:PDUTycycle

CONFigure[:SCALar]:VOLTage:PDUTycycle [<*reference*>][<*channel_list*>] is used to configure the source specified by *channel_list* for a Positive Duty Cycle measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reference	numeric	0 to 100	PCT
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure waveform memory 4 for a duty cycle measurement

```
CONF:VOLT:PDUT  (@WMEM4)
```

Configure waveform memory 4 for a Positive Duty Cycle measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Combinations of *reference and channel_list*.** The various combinations of *reference* and *channel_list* are entered as follows :

  <*reference*>,<*channel_list*> – when selecting both parameters

  <*channel_list*> – when selecting the channel list only (uses middle defaults value (50%)

  Specifying Reference: The measurement is performed at a specified level if the *reference* parameter is used. Values from 0 to 100% are accepted. *reference* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *reference* to 0.2 volts, send the following:

  ```
  CONF:VOLT:PDUT  0.2V,(@XXX)
  ```

- **Oscilloscope Setup:** In order to perform a Positive Duty Cycle measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. Positive Duty Cycle is determined at the specified reference level on the waveform.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the positive duty cycle of the source specified.

- **Measurement Method:** The method the instrument uses to determine duty cycle is to measure PWIDth and PERiod, then present duty cycle in percent as ratio of the positive pulse width to period as follows:

  positive duty cycle = + pulse width/period

- **DCYCle versus PDUTycycle:** CONFigure[:SCALar]:VOLTage:DCYCle command is identical to the CONFigure[:SCALar]:VOLTage:PDUTycycle command.

- **Related Commands:** READ?, INITiate, FETCh?, CONFigure[:SCALar]:VOLTage:DCYCle.

:PERiod

CONFigure[:SCALar]:VOLTage:PERiod [<*channel_list*>] is used to configure the source specified by *channel_list* for a Period measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure math function 1 for a period measurement

```
CONF:VOLT:PER   (@MATH1)  Configure math function 1 for a
                          Period measurement
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (*@source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Period measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the period (in seconds) of the source specified.
- **Measurement Method:** The method the instrument uses to determine period is to measure the time of the first complete cycle, then calculate period as follows:

  If first edge of waveform is rising, then:
  period = time at second rising edge – time at first rising edge

  If first edge of waveform is falling, then:
  period = time at second falling edge – time at first falling edge

- **Related Commands:** READ?, INITiate, FETCh?.

:PWIDth

CONFigure[:SCALar]:VOLTage:PWIDth [<*reference*>][<*channel_list*>] is used to configure the source specified by *channel_list* for a Positive Pulse Width measurement.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reference	numeric	0 to 100	PCT
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 2 for a positive pulse width measurement

```
CONF:VOLT:PWID  (@INP2)
```
Configure input 2 for a Positive Pulse Width measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Combinations of *reference and channel_list*.** The various combinations of *reference* and *channel_list* are entered as follows :

  <*reference*>,<*channel_list*> – when selecting both parameters

  <*channel_list*> – when selecting the channel list only (uses middle defaults value (50%)

  Specifying Reference: The measurement is performed at a specified level if the *reference* parameter is used. Values from 0 to 100% are accepted. *reference* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *reference* to 0.2 volts, send the following:

  ```
  CONF:VOLT:PWID  0.2V,(@XXX)
  ```

- **Oscilloscope Setup:** In order to perform a Positive Pulse Width measurement, a minimum of one positive pulse must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. Positive Pulse Width is determined at the specified reference level on the waveform.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the positive pulse width (in seconds) of the source specified.

- **Measurement Method:** The method the instrument uses to determine pulse width is to measure the time of the first complete positive pulse, then calculate pulse width as follows:

  If first edge of waveform is rising, then:
  pulse width = time at second falling edge – time at first rising edge

  If first edge of waveform is falling, then:
  pulse width = time at first falling edge – time at first rising edge

- **Related Commands:** READ?, INITiate, FETCh?.

:RISE:OVERshoot

CONFigure[:SCALar]:VOLTage:RISE:OVERshoot [<*channel_list*>] is used to configure the source specified by *channel_list* for an Overshoot measurement on the rising edge of the waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 3 for an overshoot measurement on the rising edge of the waveform

```
CONF:VOLT:RISE:OVER   (@INP3)  Configure input 3 for an
                               overshoot measurement
                               on the rising edge
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform an Overshoot measurement, a minimum of one rising edge must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the overshoot of the rising edge (in percent) of the source specified.
- **Measurement Method:** The method the instrument uses to determine overshoot is to make three different voltage measurements of the first rising (positive-going) edge present, then calculate overshoot as follows:

  overshoot = ((HIGH - MAXimum)/AMPLitude) • 100

- **Related Commands:** READ?, INITiate, FETCh?.

:RISE:PREShoot

CONFigure[:SCALar]:VOLTage:RISE:PREShoot [<*channel_list*>] is used to configure the source specified by *channel_list* for a Preshoot measurement on the rising edge of the waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 4 for a preshoot measurement on the rising edge of the waveform

```
CONF:VOLT:RISE:PRES   (@INP4)   Configure input 4 for a
                                preshoot measurement on
                                the rising edge
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Preshoot measurement, a minimum of one rising edge must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the preshoot of the rising edge (in percent) of the source specified.
- **Measurement Method:** The method the instrument uses to determine preshoot is to make three different voltage measurements of the first rising (positive-going) edge present, then calculate preshoot as follows:

  preshoot = ((LOW - MINimum)/AMPLitude) * 100

- **Related Commands:** READ?, INITiate, FETCh?.

:RISE:TIME

CONFigure[:SCALar]:VOLTage:RISE:TIME [*lower_limit*[*,upper_limit*]][*<channel_list>*] is used to configure the source specified by *channel_list* for a Rise Time measurement. *lower_limit* defines the lower measurement threshold. *upper_limit* defines the upper measurement threshold.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none
lower_limit	numeric	−25.00 to +125.0	PCT
upper_limit	numeric	−25.00 to +125.0	PCT

Example

Configure waveform memory 1 for a rise time measurement at 10% lower threshold limit and 90% upper threshold limit

```
CONF:VOLT:FALL:TIME  (@WMEM1)
```

Configure waveform memory 1 for a Rise Time measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Combinations of *upper_limit/lower_limit/channel_list.*** The various combinations of *upper_limit*, *lower_limit*, and *channel_list* are entered as follows :

  <lower_limit>,<upper_limit>,<channel_list> – when selecting all parameters

  <lower_limit>,<channel_list> – when selecting the lower limit and channel list (uses upper limit default)

  <channel_list> – when selecting the channel list only (uses upper and lower limit defaults)

- **Selecting Limits:** The upper and lower limits for the rise time measurement can be defined by the user as follows (units of percent and voltage cannot be specified together):

  Default: The measurement is performed at 10%/90% threshold levels if the *lower_limit* and *upper_limit* parameters are omitted.

  Specifying Lower Limit: The measurement is performed at a specified lower threshold if the *lower_limit* parameter is used. Values from −25.00% to +125.0% are accepted. *lower_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *lower_limit* to 0.2 volts, send the following:

  ```
  CONF:VOLT:RISE:TIME  0.2V,XXXX,(@X)
  ```

Specifying Upper Limit: The measurement is performed at a specified upper threshold if the *upper_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *upper_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *upper_limit* to 4.5 volts, send the following:

```
CONF:VOLT:RISE:TIME   XXXX,4.5V,(@X)
```

- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Rise Time measurement, the rising edge of the waveform must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. For best measurement accuracy, set the sweep speed as fast as possible. If the horizontal scaling is questionable when performing a Rise Time measurement, an error will be generated.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the rise time (in seconds) at the specified upper and lower limits of the selected source.
- **Measurement Method:** The method the instrument uses to determine rise time is to measure the time at the lower threshold then the upper threshold of the first rising (positive-going) edge present. Measurement results are used to calculate rise time as follows:

  rise time = upper threshold time – lower threshold time
- **RISE:TIME versus RTIMe:** CONFigure[:SCALar]:VOLTage:RISE:TIME command is identical to the CONFigure[:SCALar]:VOLTage:RTIMe command.
- **Related Commands:** READ?, INITiate, FETCh?, CONFigure[:SCALar]:VOLTAGE:RTIMe.
- ***RST Conditions:** Lower limit defaults to 10% and upper limit defaults to 90%.

:RTIMe

CONFigure[:SCALar]:VOLTage:RTIMe [*lower_limit*[,*upper_limit*]][<*channel_list*>] is used to configure the source specified by *channel_list* for a Rise Time measurement. *lower_limit* defines the lower measurement threshold. *upper_limit* defines the upper measurement threshold.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none
lower_limit	numeric	–25.00 to +125.0	PCT
upper_limit	numeric	–25.00 to +125.0	PCT

Example

Configure input 2 for a rise time measurement at 20% lower threshold limit and 70% upper threshold limit

```
CONF:VOLT:RTIM  20,70,(@INP2)
```

Configure input 2 for a Rise Time measurement

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Combinations of *upper_limit/lower_limit/channel_list*.** The various combinations of *upper_limit*, *lower_limit*, and *channel_list* are entered as follows :

  <*lower_limit*>,<*upper_limit*>,<*channel_list*> – when selecting all parameters

  <*lower_limit*>,<*channel_list*> – when selecting the lower limit and channel list (uses upper limit default)

  <*channel_list*> – when selecting the channel list only (uses upper and lower limit defaults)

- **Selecting Limits:** The upper and lower limits for the rise time measurement can be defined by the user as follows (units of percent and voltage cannot be specified together):

  Default: The measurement is performed at 10%/90% threshold levels if the *lower_limit* and *upper_limit* parameters are omitted.

  Specifying Lower Limit: The measurement is performed at a specified lower threshold if the *lower_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *lower_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *lower_limit* to 0.2 volts, send the following:

  ```
  CONF:VOLT:RTIM   0.2V,XXXX,(@X)
  ```

Specifying Upper Limit: The measurement is performed at a specified upper threshold if the *upper_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *upper_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *upper_limit* to 4.5 volts, send the following:

```
CONF:VOLT:RTIM   XXXX,4.5V,(@X)
```

- **Selecting *channel_list*.** *channel_list* has the form (*@source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Rise Time measurement, the rising edge of the waveform must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. For best measurement accuracy, set the sweep speed as fast as possible. If the horizontal scaling is questionable when performing a Rise Time measurement, an error will be generated.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the rise time (in seconds) at the specified upper and lower limits of the selected source.
- **Measurement Method:** The method the instrument uses to determine rise time is to measure the time at the lower threshold then the upper threshold of the first rising (positive-going) edge present. Measurement results are used to calculate rise time as follows:

  rise time = upper threshold time – lower threshold time
- **RISE:TIME versus RTIMe:** CONFigure[:SCALar]:VOLTage:RTIMe command is identical to the CONFigure[:SCALar]:VOLTage:RISE:TIME command.
- **Related Commands:** READ?, INITiate, FETCh?, CONFigure[:SCALar]:VOLTAGE:RISE:TIME.
- ***RST Conditions:** Lower limit defaults to 10% and upper limit defaults to 90%.

:TMAXImum

CONFigure[:SCALar]:VOLTage:TMAXimum [<*channel_list*>] returns the time at which the first maximum voltage occurred on the present waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 3 for a first maximum voltage time measurement

```
CONF:VOLT:TMAX   (@INP3)   Configure input 3 for a time at
                           maximum voltage measurement
```

Comments

- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the time (in seconds, referenced to the trigger) that the maximum voltage occurred on the selected source.

:TMINimum

CONFigure[:SCALar]:VOLTage:TMINimum [<*channel_list*>] returns the time at which the first minimum voltage occurred on the present waveform. Data is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 4 for a first minimum voltage time measurement

```
CONF:VOLT:TMIN   (@INP4)  Configure input 4 for a time at
                          minimum voltage measurement
```

Comments

- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being configured for measurement. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Executing the Measurement:** When the measurement is executed using the INITiate/FETCh? or READ? command(s), the instrument will measure and output the time (in seconds, referenced to the trigger) that the minimum voltage occurred on the selected source.

FETCh?

The FETCh? query retrieves measurement results performed by the most recent INITiate command, and places them in the output buffer. Measurement must have been previously configured using the CONFigure command.

Subsystem Syntax FETCh[[:SCALar]:VOLTage[:<*function*>]]?

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
function	discrete	AC|AMPLitude|[DC]|DCYCle| FALL:OVERshoot|FALL:PREShoot| FALL:TIME|FREQuency|FTIMe| HIGH|LOW|MAXimum|MINimum| NDUTycycle|NWIDth|PDUTycycle| PERiod|PWIDth|RISE:OVERshoot| RISE:PREShoot|RISE:TIME|RTIMe| TMAXimum|TMINimum	none

Comments

- **Selecting Function:** Depending on the desired action, *function* can either be specified or omitted as follows:

  FETCh?: When executed, will retrieve results from the measurement that was last initiated. A measurement must have been previously performed to return any measurement results. If *RST is executed prior to sending the FETCh?, an error will be generated.

  FETCh[:SCALar]:VOLTage[:<*function*>]?: When executed, will retrieve the results from the previously configured measurement as specified using *function*.

  For example,

`dimension statement`	*String for data*
`CONF:VOLT:FREQ (@1)`	*Configure input 1 for a frequency measurement*
`MEAS:VOLT:AC? (@2)`	*Configure input 2 for an AC voltage measurement*
`CONF:VOLT:PER (@3)`	*Configure input 3 for a period measurement*
`ABOR`	*Stop all measurements*
`INIT`	*Acquire waveform data*
`FETC?`	***Would return AC voltage measurement results***

OR

`FETC:VOLT:FREQ?`	***Would return frequency measurement results***
`enter statement`	*Enter measurement results into computer*

- **FETCh? and INITiate versus READ?:** Performing a measurement using the INITiate and FETCh? commands are identical to performing the READ? query.
- **Return Format:** Previous data stored in the output buffer is lost when a FETCh? is executed. A returned 9.99999E+37 indicates an invalid measurement. See the individual CONFigure commands for more information on returned measurement results.

FORMat

The FORMat command subsystem is used to specify the format of waveform data and set data transmission mode.

Subsystem Syntax

```
FORMat
    [:DATA] <type>,<format>
    [:DATA]?
```

[:DATA]

FORMat[:DATA] *<type>,<format>* is used to specify how the data is formatted on the bus when sent from the instrument, and set the data transmission mode for waveform data output.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
type *format*	discrete numeric	INTeger 8 or 16	none 8

Example

Format waveform data (from oscilloscope over the bus) in 16 bit INTeger

The following example illustrates the use of the FORMat[:DATA] command only. Chapter 5 contains an example on performing a complete digitizing operation.

```
FORM  INT,16
```

Waveform data sent over the bus will be in 16 bit integer format

Comments

- **Block Data:** Formatted waveform records are transmitted using the definite block program data format specified in IEEE 488.2. When using this format, the ASCII character string "#6<DD..D>" is sent before the actual data. The 6 indicates how many <D> 's will follow. The < D > 's are ASCII numbers, which indicate how many data bytes will follow.

  For example, if 512 points were acquired the Block Header "#3512" would be sent. The 3 indicates that three length bytes follow, 512 indicates that 512 data bytes (binary) follow.

- **Selecting Format:** Format is selected using the following guidelines:

  16 bit: Useful in applications where the information is read directly into an integer array in a controller. This format also returns the most accurate data values and greatest resolution. Formatted data transfers as 16-bit binary integers in two bytes. The number of data bytes is twice the number of words (data points). The most significant byte of each word is sent first. If there is a hole in the data, it will be represented by the 16-bit value of −1. The range of data in the WORD format is from 0 to 32640.

  8 bit: Will transfer over the bus faster than 16 bit formatted data, but has less resolution. Only seven bits are used to represent the voltage values. The first bit is a 0 or −1. If there is a hole in the data, it is represented by a value of −1.

- **Pixel Data:** Pixel data is always returned in 8 bit format.
- **Learn String:** The learn string returned using the SYSTem:SET? query is not effected by the FORMat[:DATA] command.
- **Related Commands:** TRACe[:DATA]?, TRACe:POINts.
- ***RST Conditions:** Defaults to 8 bit.

[:DATA]? FORMat[:DATA]? returns the currently selected output format (8 or 16 bit) for transfer of waveform data. Data is sent to the output buffer.

Example **Query current data format**

`dimension statement`	*String to hold data*
`FORM INT,16`	*Waveform data sent over the bus will be in 16 bit integer format*
`FORM?`	*Query instrument to return selected format*
`enter statement`	*Enter data into computer*

INITiate

The INITiate command subsystem acquires waveform data. The INITiate command performs two functions:

- Initiates or begins a previously configured measurement.
- Digitizes waveform data.

Subsystem Syntax

```
INITiate
    :CONTinuous <mode>
    :CONTinuous?
    [:IMMediate]
```

:CONTInuous

INITiate:CONTinuous ***<mode>*** is used to enable or disable acquiring waveform data.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Unit
mode	boolean	AUTO|ON|OFF	none

Example

Set acquisition mode to continuous

```
INIT:CONT  ON            Acquire data continuously
```

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Continuous Data Acquisition:** Continuous data acquisition is enabled with INITiate:CONTinuous to ON | 1. A data acquisition takes place each time a trigger that satisfies the selected TRIGger requirements is received.
- **Single Data Acquisition:** Single data acquisition is enabled with INITiate:CONTinuous to OFF | 0. Sending the INITiate[:IMMediate] command causes one data acquisition to take place. Another data acquisition will only take place if the INITiate[:IMMediate] command is sent again.
- **Automatic Data Acquisition:** Automatic data acquisition will acquire data regardless of trigger requirements. If selected and no trigger is present, available data is acquired. Provides a baseline in the absence of a signal. If a signal is present but the instrument is not triggered, the waveform will be unsynchronized (not a baseline).
- **DC Measurement:** SCPI requires that a DC measurement be performed in AUTOmode, otherwise the oscilloscope may not trigger and the digitize will not complete. If the oscilloscope is not set to CONT AUTO, then a setting conflict will occur when a DC measurement is attempted.
- **Stop Acquiring Data:** See the ABORt command.
- **Related Commands:** ABORt, TRIGger.
- ***RST Condition:** Defaults to AUTO.

:CONTinuous?

INITiate:CONTinuous? returns a value to show the data acquisition state. ON = Continuous, OFF = Single, AUTO = Automatically triggered. The value is sent to the output buffer.

Example **Query acquisition state**

`dimension statement`	*String for the data*
`INIT:CONT ON`	*Acquire data continuously*
`INIT:CONT?`	*Query instrument to return acquisition state*
`enter statement`	*Enter results into computer*

[:IMMediate]

INITiate[:IMMediate] starts the data acquisition process on all inputs selected using the [SENSe:]INPut<n>[:STATe] command.

Example **Acquire data present on input 3 one time and save in the input 3 buffer**

`INP3 ON`	*Enable input 3*
`INIT:CONT OFF`	*Acquire data once*
`INIT`	*Input 3 waveform data acquired*

Comments

- **Starting Data Acquisition:** The INITiate[:IMMediate] command starts data acquisition by acquiring the waveform data on each active input selected using the [SENSe:]INPut<n>:STATe command, with the resulting digitized data being placed in the input buffer. Further action is dependent on the INITiate:CONTinuous state:

  INITiate:CONTinuous OFF: Data is acquired one time only.

  INITiate:CONTinuous ON: An ABORt command must be sent prior to the INITiate[:IMMediate] command. Failure to do so will generate an error.

- **Before INITiate[:IMMediate]:** Before the waveform data can be acquired, certain conditions must be setup. These conditions depend on the function being performed as follows:

  Measure Parameters: To measure the data (e.g. FREQuency, PERiod, etc) the specific measurement must be setup or CONFigured. See the CONFigure subsystem for more information.

  Digitize the Waveform: To digitize the data conditions such as AVERage:TYPE, number of SWEep:POINts, and the AVERage:COUNt must be selected. See the [SENSe:] subsystem for more information on these commands.

- **After Data Acquisition:** After the waveform data has been acquired (using INITiate[:IMMediate] command), the data is read depending on the function being performed as follows:

  Measure Parameters: When measuring specific parameters, the results are FETChed to the output buffer. See the FETCh? subsystem for more information on using this query.

  Digitize the Waveform: When digitizing the waveform, the TRACe DATA and PREamble are read. See the TRACe subsystem for more information on using these commands.

- **Loss of Acquired Data:** When the INITiate command is complete the instrument is placed in the stopped mode. When restarted (another INITiate command), the digitized data stored in the input buffers will be overwritten. Before executing another INITiate command, verify that all operations that require the digitized data are completed.
- **Related Commands:** [SENSe:], TRACe:, INITiate:CONTinuous.

MEASure

The MEASure command subsystem sets up the instrument to perform a specified measurement, and then performs the measurement. After the measurement is performed, the reading is placed in the output buffer.

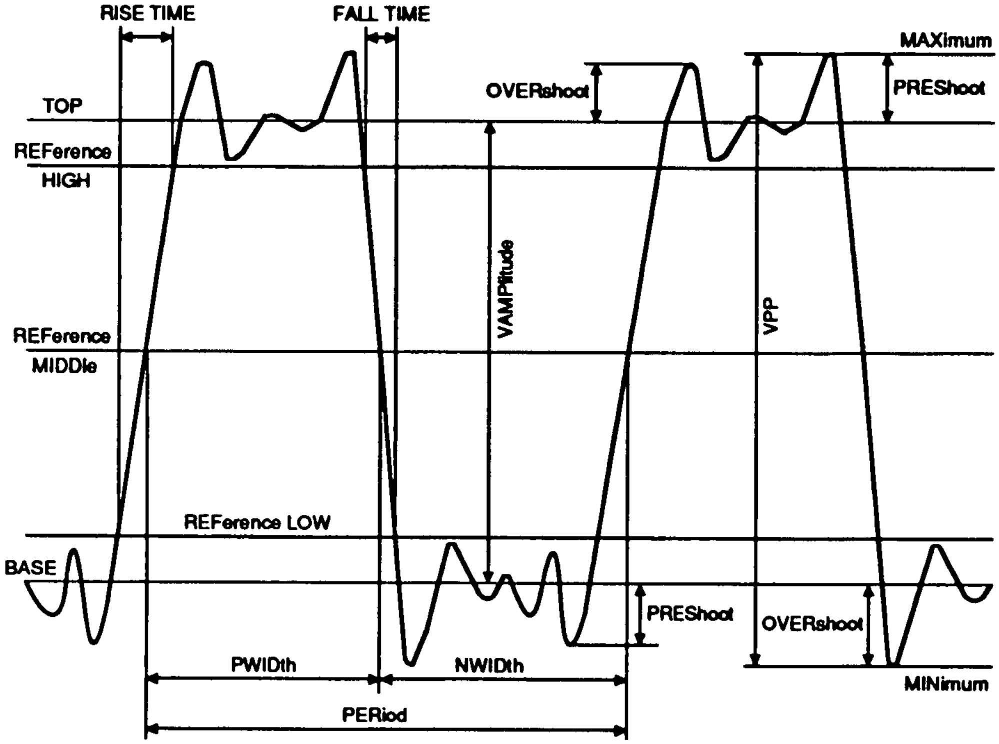
All measurements can be performed using the following methods:

> The **MEASure** configures an input for a specific function, performs the measurement, and returns the results.
>
> The **CONFigure[:SCALar]:VOLTage** command only configures an input for a specific function, and does not perform the measurement. Use additional commands/queries (READ[:<function>]?, or INIT/FETC[:<function>]?) to perform the measurement and read the result is necessary.
>
> > **[:SCALar]** specifies that a single value, not an array of readings, will be taken. **:VOLTage** specifies that the voltage characteristics of the signal will be measured.

Output Format: After the measurement is complete, the results are sent to the output buffer. Previous data in the output buffer is lost when the MEAS command is executed. A returned 9.99999E+37 indicates an invalid measurement.

Individual MEASure Commands: Refer to the individual commands for information on how the measurements are made and the returned measurement results. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.

The illustration below shows the point(s) where measurements are taken.



Subsystem Syntax

```
MEASure[:SCALar]:VOLTage
    :AC? [<channel_list>]
    :AMPLitude? [<channel_list>]
    :[DC]? [<channel_list>]
    :DCYCle? [<reference>][<channel_list>]
    :FALL
        :OVERshoot? [<channel_list>]
        :PREShoot? [<channel_list>]
        :TIME? [<lower_limit>[,<upper>]] [<channel_list>]
    :FREQuency? [<channel_list>]
    :FTIMe? [<lower_limit>[,<upper>]] [<channel_list>]
    :HIGH? [<channel_list>]
    :LOW? [<channel_list>]
    :MAXimum? [<channel_list>]
    :MINimum? [<channel_list>]
    :NDUTycycle? [<reference>][<channel_list>]
    :NWIDth?? [<reference>][<channel_list>]
    :PDUTycycle? [<reference>][<channel_list>]
    :PERiod? [<channel_list>]
    :PWIDth? [<reference>][<channel_list>]
    :RISE
        :OVERshoot? [<channel_list>]
        :PREShoot? [<channel_list>]
        :TIME? [<lower_limit>[,<upper>]] [<channel_list>]
    :RTIMe? [<lower_limit>[,<upper>]] [<channel_list>]
    :TMAXimum? [<channel_list>]
    :TMINimum? [<channel_list>]
```

:AC?

MEASure[:SCALar]:VOLTage:AC? [<*channel_list*>] is used to configure the source specified by *channel_list* for an AC RMS voltage measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make an AC RMS voltage measurement on input 1

`MEAS:VOLT:AC?  (@INP1)`	*Configure input 1 for an AC RMS Voltage measurement, perform the measurement, and transfer the result to the output buffer*
`enter  statement`	*Enter measurement into computer*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (*@source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** The AC Voltage measurement is made using the the first cycle present. If a complete cycle is not present, the AC value of all data points is calculated.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the AC RMS voltage (with 0 volts as the reference) of the source specified.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:AMPLitude?

MEASure[:SCALar]:VOLTage:AMPLitude? [<*channel_list*>] is used to configure the source specified by *channel_list* for an Amplitude voltage measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make an amplitude voltage measurement on math function 2

```
MEAS:VOLT:AMPL?  (@MATH2)
```

Configure math 2 for an Amplitude Voltage measurement, perform the measurement, and transfer the result to the output buffer

```
enter  statement
```

Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** Amplitude voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the amplitude voltage (with 0 volts as the reference) of the source specified.
- **Measurement Method:** The method the instrument uses to determine voltage amplitude is to measure HIGH and LOW, then calculate voltage amplitude as follows:

  voltage amplitude = HIGH – LOW
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

[:DC]?

MEASure[:SCALar]:VOLTage[:DC]? [<*channel_list*>] is used to configure the source specified by ***channel_list*** for a DC voltage measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a DC voltage measurement on input 3

```
MEAS:VOLT?  (@INP3)
```

Configure input 3 for a DC Voltage measurement, perform the measurement, and transfer the result to the output buffer

```
enter  statement
```

Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** DC Voltage measurement is made using the the first cycle present. If a complete cycle is not present, all currently acquired data points are averaged.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the DC voltage (with 0 volts as the reference) of the source specified.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:DCYCle?

MEASure[:SCALar]:VOLTage:DCYCle? [<*reference*>][<*channel_list*>] is used to configure the source specified by *channel_list* for a Duty Cycle measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reference	numeric	0 to 100	PCT
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a duty cycle measurement on waveform memory 4

```
MEAS:VOLT:DCYC?  (@WMEM4)   Configure waveform memory
                            4 for a Duty Cycle
                            measurement, perform the
                            measurement, and transfer the
                            result to the output buffer
enter  statement            Enter measurement into
                            computer
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Combinations of *reference and channel_list*.** The various combinations of *reference* and *channel_list* are entered as follows :

  <*reference*>,<*channel_list*> – when selecting both parameters

  <*channel_list*> – when selecting the channel list only (uses middle defaults value (50%)

  Specifying Reference: The measurement is performed at a specified level if the *reference* parameter is used. Values from 0 to 100% are accepted. *reference* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *reference* to 0.2 volts, send the following:

  ```
  CONF:VOLT:DCYC  0.2V, (@XXX)
  ```

- **Oscilloscope Setup:** In order to perform a Duty Cycle measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. Duty Cycle is determined at the specified reference level on the waveform. If the signal is not present, 9.99999E+37 is returned.

- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the duty cycle of the source specified.
- **Measurement Method:** The method the instrument uses to determine duty cycle is to measure PWIDth and PERiod, then present duty cycle in percent as ratio of the positive pulse width to period as follows:

  duty cycle = + pulse width/period
- **DCYCle versus PDUTycycle:** MEASure[:SCALar]:VOLTage:DCYCle command is identical to the MEASure[:SCALar]:VOLTage:PDUTycycle command.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?, MEASure[:SCALar]:VOLTage:PDUTycycle.

:FALL:OVERshoot?

MEASure[:SCALar]:VOLTage:FALL:OVERshoot? [<*channel_list*>] is used to configure the source specified by *channel_list* for an Overshoot measurement on the falling edge of the waveform, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make an overshoot measurement on the falling edge of the waveform on input 2

```
MEAS:VOLT:FALL:OVER? (@INP2)
```
Configure input 2 for an overshoot measurement on the falling edge, perform the measurement, and transfer the result to the output buffer

```
enter statement
```
Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform an Overshoot measurement, a minimum of one falling edge must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. If the signal is not present, 9.99999E+37 is returned.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the overshoot of the falling edge (in percent) of the source specified.
- **Measurement Method:** The method the instrument uses to determine overshoot is to make three different voltage measurements of the first falling (negative-going) edge present, then calculate overshoot as follows:

  overshoot = ((LOW - MINimum)/AMPLitude) • 100
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:FALL:PREShoot?

MEASure[:SCALar]:VOLTage:FALL:PREShoot? [<*channel_list*>] is used to configure the source specified by *channel_list* for a Preshoot measurement on the falling edge of the waveform, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a preshoot measurement on the falling edge of the waveform on input 3

```
MEAS:VOLT:FALL:PRES?  (@INP3)
```

Configure input 3 for a preshoot measurement on the falling edge, perform the measurement, and transfer the result to the output buffer

```
enter  statement
```

Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (*@source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Preshoot measurement, a minimum of one falling edge must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. If the signal is not present, 9.99999E+37 is returned.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the preshoot of the falling edge (in percent) of the source specified.
- **Measurement Method:** The method the instrument uses to determine preshoot is to make three different voltage measurements of the first falling (negative-going) edge present, then calculate preshoot as follows:

  preshoot = ((HIGH - MAXimum)/AMPLitude) • 100

- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:FALL:TIME?

MEASure[:SCALar]:VOLTage:FALL:TIME? [*lower_limit*[,*upper_limit*]][<*channel_list*>] is used to configure the source specified by *channel_list* for a Fall Time measurement, perform the measurement on the signal present, then transfer the results to the output buffer. *lower_limit* defines the lower measurement threshold. *upper_limit* defines the upper measurement threshold.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none
lower_limit	numeric	−25.00 to +125.0	PCT
upper_limit	numeric	−25.00 to +125.0	PCT

Example

Make a fall time measurement at 10% lower threshold limit and 90% upper threshold limit on input 1 (defaults)

`MEAS:VOLT:FALL:TIME?`	*Configure input 1 for a Fall Time measurement, perform the measurement, and transfer the result to the output buffer*
`enter statement`	*Enter measurement into computer*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Combinations of *upper_limit/lower_limit/channel_list*.** The various combinations of *upper_limit*, *lower_limit*, and *channel_list* are entered as follows :

  <*lower_limit*>,<*upper_limit*>,<*channel_list*> – when selecting all parameters

  <*lower_limit*>,<*channel_list*> – when selecting the lower limit and channel list (uses upper limit default)

  <*channel_list*> – when selecting the channel list only (uses upper and lower limit defaults)

- **Selecting Limits:** The upper and lower limits for the fall time measurement can be defined by the user as follows (units of percent and voltage cannot be specified together):

  Default: The measurement is performed at 10%/90% threshold levels if the *lower_limit* and *upper_limit* parameters are omitted.

Specifying Lower Limit: The measurement is performed at a specified lower threshold if the *lower_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *lower_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *lower_limit* to 0.2 volts, send the following:

```
MEAS:VOLT:FALL:TIME?  0.2V,XXXX,(@X)
```

Specifying Upper Limit: The measurement is performed at a specified upper threshold if the *upper_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *upper_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *upper_limit* to 4.5 volts, send the following:

```
MEAS:VOLT:FALL:TIME?  XXXX,4.5V,(@X)
```

- **Selecting *channel_list*.** *channel_list* has the form (*@source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the fall time (in seconds) at the specified upper and lower limits of the selected source.
- **Oscilloscope Setup:** In order to perform a Fall Time measurement, the falling edge of the waveform must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. For best measurement accuracy, set the sweep speed as fast as possible. If the horizontal scaling is questionable when performing a Fall Time measurement, an error will be generated.
- **Measurement Method:** The method the instrument uses to determine fall time is to measure the time at the upper threshold then the lower threshold of the first falling (negative-going) edge present. Measurement results are used to calculate fall time as follows:

  fall time = lower threshold time – upper threshold time

- **FALL:TIME versus FTIMe:** MEASure[:SCALar]:VOLTage:FALL:TIME command is identical to the MEASure[:SCALar]:VOLTage:FTIMe command.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?, MEASure[:SCALar]:VOLTAGE:FTIMe.
- ***RST Conditions:** Lower limit defaults to 10% and upper limit defaults to 90%.

:FREQuency?

MEASure[:SCALar]:VOLTage:FREQuency? [<*channel_list*>] is used to configure the source specified by *channel_list* for a Frequency measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a frequency measurement on waveform memory 4

`MEAS:VOLT:FREQ?  (@WMEM4)`	*Configure waveform memory 4 for a Frequency measurement, perform the measurement, and transfer the result to the output buffer*
`enter  statement`	*Enter measurement into computer*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Frequency measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. If the signal is not present, 9.99999E+37 is returned.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the frequency (in hertz) of the source specified.
- **Measurement Method:** The method the instrument uses to determine frequency is to measure the time of the first complete cycle, then calculate frequency as follows:

  If first edge of waveform is rising, then:
  frequency = 1/(time at second rising edge – time at first rising edge)

  If first edge of waveform is falling, then:
  frequency = 1/(time at second falling edge – time at first falling edge)
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:FTIMe?

MEASure[:SCALar]:VOLTage:FTIMe? [*lower_limit*[,*upper_limit*]][<*channel_list*>] is used to configure the source specified by *channel_list* for a Fall Time measurement, perform the measurement on the signal present, then transfer the results to the output buffer. *lower_limit* defines the lower measurement threshold. *upper_limit* defines the upper measurement threshold.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none
lower_limit	numeric	–25.00 to +125.0	PCT
upper_limit	numeric	–25.00 to +125.0	PCT

Example

Make a fall time measurement at 20% lower threshold limit and 70% upper threshold limit on math function 1

```
MEAS:VOLT:FTIM?  20,70,(@MATH1)
```

Configure input 1 for a Fall Time measurement, perform the measurement, and transfer the result to the output buffer

```
enter  statement
```

Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Combinations of *upper_limit/lower_limit/channel_list*.** The various combinations of *upper_limit*, *lower_limit*, and *channel_list* are entered as follows :

  <*lower_limit*>,<*upper_limit*>,<*channel_list*> – when selecting all parameters

  <*lower_limit*>,<*channel_list*> – when selecting the lower limit and channel list (uses upper limit default)

  <*channel_list*> – when selecting the channel list only (uses upper and lower limit defaults)

- **Selecting Limits:** The upper and lower limits for the fall time measurement can be defined by the user as follows (units of percent and voltage cannot be specified together):

  Default: The measurement is performed at 10%/90% threshold levels if the *lower_limit* and *upper_limit* parameters are omitted.

  Specifying Lower Limit: The measurement is performed at a specified lower threshold if the *lower_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *lower_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *lower_limit* to 0.2 volts, send the following:

  ```
  MEAS:VOLT:FTIM  0.2V,XXXX,(@X)
  ```

  Specifying Upper Limit: The measurement is performed at a specified upper threshold if the *upper_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *upper_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *upper_limit* to 4.5 volts, send the following:

  ```
  MEAS:VOLT:FTIM  XXXX,4.5V,(@X)
  ```

- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Fall Time measurement, the falling edge of the waveform must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. For best measurement accuracy, set the sweep speed as fast as possible. If the horizontal scaling is questionable when performing a Fall Time measurement, an error will be generated.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the fall time (in seconds) at the specified upper and lower limits of the selected source.
- **Measurement Method:** The method the instrument uses to determine fall time is to measure the time at the upper threshold then the lower threshold of the first falling (negative-going) edge present. Measurement results are used to calculate fall time as follows:

  fall time = lower threshold time – upper threshold time

- **FALL:TIME versus FTIMe:** MEASure[:SCALar]:VOLTage:FTIMe command is identical to the MEASure[:SCALar]:VOLTage:FALL:TIME command.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?, MEASure[:SCALar]:VOLTAGE:FALL:TIME.
- ***RST Conditions:** Lower limit defaults to 10% and upper limit defaults to 90%.

:HIGH?

MEASure[:SCALar]:VOLTage:HIGH? [<*channel_list*>] is used to configure the source specified by *channel_list* for a High voltage measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a high voltage measurement on input 2

`MEAS:VOLT:HIGH? (@INP2)`	*Configure input 2 for a High Voltage measurement, perform the measurement, and transfer the result to the output buffer*
`enter statement`	*Enter measurement into computer*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (*@source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** High Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the voltage value (with 0 volts as the reference) at the highest point of the source specified.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:LOW?

MEASure[:SCALar]:VOLTage:LOW? [<*channel_list*>] is used to configure the source specified by *channel_list* for a Low voltage measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a low voltage measurement on waveform memory 3

```
MEAS:VOLT:LOW?  (@WMEM3)
```

Configure waveform memory 3 for a Low Voltage measurement, perform the measurement, and transfer the result to the output buffer

```
enter  statement
```

Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** Low Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the voltage value (with 0 volts as the reference) at the lowest point of the source specified.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:MAXImum?

MEASure[:SCALar]:VOLTage:MAXimum? [<*channel_list*>] is used to configure the source specified by *channel_list* for a Maximum voltage measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a maximum voltage measurement on input 4

```
MEAS:VOLT:MAX  (@INP4)
enter  statement
```

`MEAS:VOLT:MAX (@INP4)`	*Configure input 4 for a Maximum Voltage measurement, perform the measurement, and transfer the result to the output buffer*
`enter statement`	*Enter measurement into computer*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** Maximum Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the absolute maximum voltage (with 0 volts as the reference) of the source specified.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:MINImum?

MEASure[:SCALar]:VOLTage:MINimum? [<*channel_list*>] is used to configure the source specified by *channel_list* for a Minimum voltage measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a minimum voltage measurement on input 1

```
MEAS:VOLT:MIN?
enter  statement
```

Configure input 1 for a Minimum Voltage measurement, perform the measurement, and transfer the result to the output buffer (MEAS:VOLT:MIN?)

Enter measurement into computer (enter statement)

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** Minimum Voltage measurement is made using the entire waveform. When performing a measurement on a particular cycle, set the controls to present only that cycle.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the absolute minimum voltage (with 0 volts as the reference) of the source specified.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:NDUTycycle?

MEASure[:SCALar]:VOLTage:NDUTycycle? [<*reference*>][<*channel_list*>] is used to configure the source specified by *channel_list* for a Negative Duty Cycle measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reference	numeric	0 to 100	PCT
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a negative duty cycle measurement on math function 2

`MEAS:VOLT:NDUT?  (@MATH2)`	*Configure input 2 for a Negative Duty Cycle measurement, perform the measurement, and transfer the result to the output buffer*
`enter  statement`	*Enter measurement into computer*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Combinations of *reference and channel_list*.** The various combinations of *reference* and *channel_list* are entered as follows :

  <*reference*>,<*channel_list*> – when selecting both parameters

  <*channel_list*> – when selecting the channel list only (uses middle defaults value (50%)

  Specifying Reference: The measurement is performed at a specified level if the *reference* parameter is used. Values from 0 to 100% are accepted. *reference* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *reference* to 0.2 volts, send the following:

  ```
  CONF:VOLT:NDUT  0.2V,(@XXX)
  ```

- **Oscilloscope Setup:** In order to perform a Negative Duty Cycle measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. Negative Duty Cycle is determined at the specified reference level on the waveform. If the signal is not present, 9.99999E+37 is returned.

- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the negative duty cycle of the source specified.
- **Measurement Method:** The method the instrument uses to determine negative duty cycle is to measure NWIDth and PERiod, then present duty cycle in percent as ratio of the negative pulse width to period as follows:

  negative duty cycle = – pulse width/period
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:NWIDth?

MEASure[:SCALar]:VOLTage:NWIDth? [<*reference*>][<*channel_list*>] is used to configure the source specified by *channel_list* for a Negative Pulse Width measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reference *channel_list*	numeric numeric	0 to 100 INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	PCT none

Example

Make a negative pulse width measurement on input 3

```
MEAS:VOLT:NWID?  (@INP3)
```
Configure input 3 for a Negative Pulse Width measurement, perform the measurement, and transfer the result to the output buffer

```
enter  statement
```
Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Combinations of *reference and channel_list*.** The various combinations of *reference* and *channel_list* are entered as follows :

  <*reference*>,<*channel_list*> – when selecting both parameters

  <*channel_list*> – when selecting the channel list only (uses middle defaults value (50%)

  Specifying Reference: The measurement is performed at a specified level if the *reference* parameter is used. Values from 0 to 100% are accepted. *reference* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *reference* to 0.2 volts, send the following:

  ```
  CONF:VOLT:NWID  0.2V,(@XXX)
  ```

- **Oscilloscope Setup:** In order to perform a Negative Pulse Width measurement, a minimum of one negative pulse must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. Negative Pulse Width is determined at the specified reference level on the waveform. If the signal is not present, 9.99999E+37 is returned.

- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the negative pulse width (in seconds) of the source specified.
- **Measurement Method:** The method the instrument uses to determine pulse width is to measure the time of the first complete negative pulse, then calculate pulse width as follows:

  If first edge of waveform is rising, then:
  pulse width = time at second rising edge – time at first falling edge

  If first edge of waveform is falling, then:
  pulse width = time at first rising edge – time at first falling edge

- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:PDUTycycle?

MEASure[:SCALar]:VOLTage:PDUTycycle? [<*reference*>][<*channel_list*>] is used to configure the source specified by *channel_list* for a Positive Duty Cycle measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reference	numeric	0 to 100	PCT
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a duty cycle measurement on input 4

`MEAS:VOLT:PDUT?  (@INP4)`	*Configure input 4 for a Positive Duty Cycle measurement, perform the measurement, and transfer the result to the output buffer*
`enter  statement`	*Enter measurement into computer*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Combinations of *reference and channel_list*.** The various combinations of *reference* and *channel_list* are entered as follows :

  <*reference*>,<*channel_list*> – when selecting both parameters

  <*channel_list*> – when selecting the channel list only (uses middle defaults value (50%)

  Specifying Reference: The measurement is performed at a specified level if the *reference* parameter is used. Values from 0 to 100% are accepted. *reference* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *reference* to 0.2 volts, send the following:

  ```
  CONF:VOLT:PDUT   0.2V,(@XXX)
  ```

- **Oscilloscope Setup:** In order to perform a Positive Duty Cycle measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. Positive Duty Cycle is determined at the specified reference level on the waveform. If the signal is not present, 9.99999E+37 is returned.

- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the positive duty cycle of the source specified.
- **Measurement Method:** The method the instrument uses to determine duty cycle is to measure PWIDth and PERiod, then present duty cycle in percent as ratio of the positive pulse width to period as follows:

  positive duty cycle = + pulse width/period
- **DCYCle versus PDUTycycle:** MEASure[:SCALar]:VOLTage:DCYCle command is identical to the MEASure[:SCALar]:VOLTage:PDUTycycle command.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?, MEASure[:SCALar]:VOLTage:DCYCle.

:PERiod?

MEASure[:SCALar]:VOLTage:PERiod? [<*channel_list*>] is used to configure the source specified by *channel_list* for a Period measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a period measurement on math function 1

```
MEAS:VOLT:PER?  (@MATH1)
```

Configure math function 1 for a Period measurement, perform the measurement, and transfer the result to the output buffer

```
enter  statement
```

Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Period measurement, a minimum of one full cycle must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. If the signal is not present, 9.99999E+37 is returned.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the period (in seconds) of the source specified.
- **Measurement Method:** The method the instrument uses to determine period is to measure the time of the first complete cycle, then calculate period as follows:

  If first edge of waveform is rising, then:
  period = time at second rising edge – time at first rising edge

  If first edge of waveform is falling, then:
  period = time at second falling edge – time at first falling edge

- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:PWIDth?

MEASure[:SCALar]:VOLTage:PWIDth? [*<reference>*][*<channel_list>*] is used to configure the source specified by *channel_list* for a Positive Pulse Width measurement, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
reference	numeric	0 to 100	PCT
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a positive pulse width measurement on waveform memory 2

```
MEAS:VOLT:PWID?  (@WMEM2)
```
Configure waveform memory 2 for a Positive Pulse Width measurement, perform the measurement, and transfer the result to the output buffer

```
enter  statement
```
Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Combinations of *reference and channel_list*.** The various combinations of *reference* and *channel_list* are entered as follows :

  <reference>,<channel_list> – when selecting both parameters

  <channel_list> – when selecting the channel list only (uses middle defaults value (50%)

  Specifying Reference: The measurement is performed at a specified level if the *reference* parameter is used. Values from 0 to 100% are accepted. *reference* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *reference* to 0.2 volts, send the following:

  ```
  CONF:VOLT:PWID  0.2V,(@XXX)
  ```
- **Oscilloscope Setup:** In order to perform a Positive Pulse Width measurement, a minimum of one positive pulse must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. Positive Pulse Width is determined at the specified reference level on the waveform. If the signal is not present, 9.99999E+37 is returned.

- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the positive pulse width (in seconds) of the source specified.
- **Measurement Method:** The method the instrument uses to determine pulse width is to measure the time of the first complete positive pulse, then calculate pulse width as follows:

  If first edge of waveform is rising, then:
  pulse width = time at second falling edge – time at first rising edge

  If first edge of waveform is falling, then:
  pulse width = time at first falling edge – time at first rising edge

- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:RISE:OVERshoot?

MEASure[:SCALar]:VOLTage:RISE:OVERshoot? [<*channel_list*>] is used to configure the source specified by *channel_list* for an Overshoot measurement on the rising edge of the waveform, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make an overshoot measurement on the rising edge of the waveform on input 1

```
MEAS:VOLT:RISE:OVER?        Configure input 1 for an
                            overshoot measurement on
                            the rising edge, perform the
                            measurement, and transfer the
                            result to the output buffer
enter  statement            Enter measurement into
                            computer
```

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform an Overshoot measurement, a minimum of one rising edge must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. If the signal is not present, 9.99999E+37 is returned.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the overshoot of the rising edge (in percent) of the source specified.
- **Measurement Method:** The method the instrument uses to determine overshoot is to make three different voltage measurements of the first rising (positive-going) edge present, then calculate overshoot as follows:

  overshoot = ((HIGH - MAXimum)/AMPLitude) • 100
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:RISE:PREShoot?

MEASure[:SCALar]:VOLTage:RISE:PREShoot? [<*channel_list*>] is used to configure the source specified by *channel_list* for a Preshoot measurement on the rising edge of the waveform, perform the measurement on the signal present, then transfer the results to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a preshoot measurement on the rising edge of the waveform on input 4

```
MEAS:VOLT:RISE:PRES?  (@INP4)
```

Configure input 4 for a preshoot measurement on the rising edge, perform the measurement, and transfer the result to the output buffer

```
enter  statement
```

Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Preshoot measurement, a minimum of one rising edge must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. If the signal is not present, 9.99999E+37 is returned.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the preshoot of the rising edge (in percent) of the source specified.
- **Measurement Method:** The method the instrument uses to determine preshoot is to make three different voltage measurements of the first rising (positive-going) edge present, then calculate preshoot as follows:

  preshoot = ((LOW - MINimum)/AMPLitude) • 100
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?.

:RISE:TIME?

MEASure[:SCALar]:VOLTage:RISE:TIME? [*lower_limit*[,*upper_limit*]][<*channel_list*>] is used to configure the source specified by *channel_list* for a Rise Time measurement, perform the measurement on the signal present, then transfer the results to the output buffer. *lower_limit* defines the lower measurement threshold. *upper_limit* defines the upper measurement threshold.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none
lower_limit	numeric	−25.00 to +125.0	PCT
upper_limit	numeric	−25.00 to +125.0	PCT

Example

Make a rise time measurement at 10% lower threshold limit and 90% upper threshold limit on waveform memory 1 (defaults)

`MEAS:VOLT:FALL:TIME?   (@WMEM1)`	*Configure waveform memory 1 for a Rise Time measurement, perform the measurement, and transfer the result to the output buffer*
`enter  statement`	*Enter measurement into computer*

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Combinations of *upper_limit/lower_limit/channel_list*.** The various combinations of *upper_limit*, *lower_limit*, and *channel_list* are entered as follows :

  <*lower_limit*>,<*upper_limit*>,<*channel_list*> – when selecting all parameters

  <*lower_limit*>,<*channel_list*> – when selecting the lower limit and channel list (uses upper limit default)

  <*channel_list*> – when selecting the channel list only (uses upper and lower limit defaults)
- **Selecting Limits:** The upper and lower limits for the rise time measurement can be defined by the user as follows (units of percent and voltage cannot be specified together):

  Default: The measurement is performed at 10%/90% threshold levels if the *lower_limit* and *upper_limit* parameters are omitted.

Specifying Lower Limit: The measurement is performed at a specified lower threshold if the *lower_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *lower_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *lower_limit* to 0.2 volts, send the following:

```
MEAS:VOLT:RISE:TIME  0.2V,XXXX,(@X)
```

Specifying Upper Limit: The measurement is performed at a specified upper threshold if the *upper_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *upper_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *upper_limit* to 4.5 volts, send the following:

```
MEAS:VOLT:RISE:TIME  XXXX,4.5V,(@X)
```

- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Rise Time measurement, the rising edge of the waveform must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. For best measurement accuracy, set the sweep speed as fast as possible. If the horizontal scaling is questionable when performing a Rise Time measurement, an error will be generated.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the rise time (in seconds) at the specified upper and lower limits of the selected source.
- **Measurement Method:** The method the instrument uses to determine rise time is to measure the time at the lower threshold then the upper threshold of the first rising (positive-going) edge present. Measurement results are used to calculate rise time as follows:

  rise time = upper threshold time – lower threshold time

- **RISE:TIME versus RTIMe:** MEASure[:SCALar]:VOLTage:RISE:TIME command is identical to the MEASure[:SCALar]:VOLTage:RTIMe command.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?, MEASure[:SCALar]:VOLTAGE:RTIMe.
- ***RST Conditions:** Lower limit defaults to 10% and upper limit defaults to 90%.

:RTIMe?

MEASure[:SCALar]:VOLTage:RTIMe? [*lower_limit*[,*upper_limit*]][<*channel_list*>] is used to configure the source specified by *channel_list* for a Rise Time measurement, perform the measurement on the signal present, then transfer the results to the output buffer. *lower_limit* defines the lower measurement threshold. *upper_limit* defines the upper measurement threshold.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none
lower_limit	numeric	−25.00 to +125.0	PCT
upper_limit	numeric	−25.00 to +125.0	PCT

Example

Make a rise time measurement at 20% lower threshold limit and 70% upper threshold limit on input 2

```
MEAS:VOLT:RTIM?  20,70,(@INP2)
```

Configure input 2 for a Rise Time measurement, perform the measurement, and transfer the result to the output buffer

```
enter  statement
```

Enter measurement into computer

Comments

- **Measurement Specifications:** See Appendices A and C for measurement specifications.
- **Combinations of *upper_limit/lower_limit/channel_list*.** The various combinations of *upper_limit*, *lower_limit*, and *channel_list* are entered as follows :

  <*lower_limit*>,<*upper_limit*>,<*channel_list*> – when selecting all parameters

  <*lower_limit*>,<*channel_list*> – when selecting the lower limit and channel list (uses upper limit default)

  <*channel_list*> – when selecting the channel list only (uses upper and lower limit defaults)

- **Selecting Limits:** The upper and lower limits for the rise time measurement can be defined by the user as follows (units of percent and voltage cannot be specified together):

  Default: The measurement is performed at 10%/90% threshold levels if the *lower_limit* and *upper_limit* parameters are omitted.

Specifying Lower Limit: The measurement is performed at a specified lower threshold if the *lower_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *lower_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *lower_limit* to 0.2 volts, send the following:

```
MEAS:VOLT:RTIM  0.2V,XXXX,(@X)
```

Specifying Upper Limit: The measurement is performed at a specified upper threshold if the *upper_limit* parameter is used. Values from –25.00% to +125.0% are accepted. *upper_limit* can also be entered in voltage units (±250,000 volts) by specifying "V" in the parameter. For example, to set the *upper_limit* to 4.5 volts, send the following:

```
MEAS:VOLT:RTIM  XXXX,4.5V,(@X)
```

- **Selecting *channel_list*.** *channel_list* has the form (*@source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Oscilloscope Setup:** In order to perform a Rise Time measurement, the rising edge of the waveform must be present. If more than one waveform, edge, or pulse is present, the measurement is made on the first edge acquired. For best measurement accuracy, set the sweep speed as fast as possible. If the horizontal scaling is questionable when performing a Rise Time measurement, an error will be generated.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the rise time (in seconds) at the specified upper and lower limits of the selected source.
- **Measurement Method:** The method the instrument uses to determine rise time is to measure the time at the lower threshold then the upper threshold of the first rising (positive-going) edge present. Measurement results are used to calculate rise time as follows:

  rise time = upper threshold time – lower threshold time
- **RISE:TIME versus RTIMe:** MEASure[:SCALar]:VOLTage:RTIMe command is identical to the MEASure[:SCALar]:VOLTage:RISE:TIME command.
- **Related Commands:** CONFigure:, READ?, INITiate, FETCh?, MEASure[:SCALar]:VOLTAGE:RISE:TIME.
- ***RST Conditions:** Lower limit defaults to 10% and upper limit defaults to 90%.

:TMAXimum?

MEASure[:SCALar]:VOLTage:TMAXimum? [<*channel_list*>] returns the time at which the first maximum voltage occurred on the present waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel_list	numeric	INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Measure the first maximum voltage time on input 1

`MEAS:VOLT:TMAX?`	*Configure input 2 for a time at maximum voltage measurement, perform the measurement, and transfer the result to the output buffer*
`enter statement`	*Enter data into computer*

Comments

- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the time (in seconds) that the maximum voltage occurred. The trigger point is used as the reference (time 0).

:TMINImum?

MEASure[:SCALar]:VOLTage:TMINimum? [<*channel_list*>] returns the time at which the first minimum voltage occurred on the present waveform.

Example **Measure the first minimum voltage time for input 3**

`MEAS:VOLT:TMIN? (@INP3)`	*Configure input 3 for a time at minimum voltage measurement, perform the measurement, and transfer the result to the output buffer*
`enter statement`	*Enter data into computer*

Comments

- **Selecting *channel_list*.** *channel_list* has the form (@*source*) where *source* is the specified input, waveform memory, or math function being measured. Only one source can be specified at a time. If *channel_list* is not specified, INPut1 is used.
- **Executing the Measurement:** When the measurement is executed, the instrument will measure and output the time (in seconds) that the minimum voltage occurred. The trigger point is used as the reference (time 0).

MEMory

The MEMory command subsystem enables the Oscilloscope to use external A24 VME memory space for routing acquired data from the internal buffers to the external buffer. It also enables measurements to be performed using data from the internal buffer or external buffer.

Subsystem Syntax

```
MEMory
   :VME
      :ADDRess <address>
      :ADDRess? [MINimum|MAXimum]
      :MEASure
         :ADDRess <address>
         :ADDRess? [MINimum|MAXimum]
         :STATe <mode>
         :STATe?
      :SIZE <bytes>
      :SIZE?
      :STATe <mode>
      :STATe?
```

The MEMory command allows the Oscilloscope to use external A24 VME memory for data acquisition, when the VME STATe is ON. Once the data has been acquired, it is available in "raw" format. Data is processed depending on type and format selected using the following guidelines:

- **Acquisition TYPE NORMal** — The acquired data is 16 bits in length. The starting location, in VME space, of each INPUT buffer is to be calculated as follows:

  INPut1 = MEM:VME:ADDR
  INPut2 = MEM:VME:ADDR + 8192
  INPut3 = MEM:VME:ADDR + 16384
  INPut4 = MEM:VME:ADDR + 24576

```
Data — 16 bits
1111110000000000
5432109876543210
xxxxxxxxxxxxxxxx
```

```
         BYTE                      WORD                    COMPRESSED
15                   0  15                   0  15                       0
xxxxxxx--------->  xxxxxxxxxxxxxxx>  xxxxxxxxxx-------->
  shift right 9 bits        shift right 1 bit          shift right 8 bits
```

*NOTE: x = bits to shift. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the TRACe:PREamble? query.

- **Acquisition TYPE AVERage — The acquired data is 32 bits in length. The starting location, in VME space, of each INPUT buffer is to be calculated as follows:**

  INPut1 = MEM:VME:ADDR
  INPut2 = MEM:VME:ADDR + 8192
  INPut3 = MEM:VME:ADDR + 16384
  INPut4 = MEM:VME:ADDR + 24576

```
                              Data — 32 bits
                     33222222222211111111110000000000
                     10987654321098765432109876543210
                             xxxxxxxxxxxxxxxx

         BYTE                     WORD                  COMPRESSED
15                   0  15                   0  15                       0
xxxxxxx---------->  xxxxxxxxxxxxxxx>  xxxxxxxxxx-------->
    shift right 9 bits       shift right 1 bit        shift right 8 bits
```

*NOTE: x = bits to shift. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the TRACe:PREamble? query.

- **Acquisition TYPE ENVelope — The acquired data is 16 bits in length. The starting location, in VME space, of each INPUT buffer is be to calculated as follows:**

  If data is digitized:

  MINIMUM

  INPut1 = MEM:VME:ADDR
  INPut2 = MEM:VME:ADDR + 8192
  INPut3 = MEM:VME:ADDR + 16384
  INPut4 = MEM:VME:ADDR + 24576

  MAXIMUM

  INPut1 = MEM:VME:ADDR + (2 * SWEep:POINts)
  INPut2 = MEM:VME:ADDR + 8192 + (2 * SWEep:POINts)
  INPut3 = MEM:VME:ADDR + 16384 + (2 * SWEep:POINts)
  INPut4 = MEM:VME:ADDR + 24576 + (2 * SWEep:POINts)

  If data is NOT digitized:

  MINIMUM

  INPut1 = MEM:VME:ADDR
  INPut2 = MEM:VME:ADDR + 8192
  INPut3 = MEM:VME:ADDR + 16384
  INPut4 = MEM:VME:ADDR + 24576

  MAXIMUM

  INPut1 = MEM:VME:ADDR + 1002
  INPut2 = MEM:VME:ADDR + 8192 + 1002
  INPut3 = MEM:VME:ADDR + 16384 + 1002
  INPut4 = MEM:VME:ADDR + 24576 + 1002

Data — 16 bits

```
                         1111110000000000
                         5432109876543210
                         xxxxxxxxxxxxxxxx

        BYTE                     WORD                   COMPRESSED
15                0   15                  0   15                       0
xxxxxxx--------->     xxxxxxxxxxxxxxx>        xxxxxxxxxx-------->
  shift right 9 bits     shift right 1 bit       shift right 8 bits
```

*NOTE: x = bits to shift. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the TRACe:PREamble? query.

:VME:ADDRess

MEMory:VME:ADDRess ***<address>*** sets the address of the external memory board in A24 memory address space where acquisition data will be available. ***address*** must be on an even boundary or a settings conflict will be generated.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
address	numeric	2097152-14647294| #H200000-#HDF7FFE|MIN|MAX	none

Example

Setting the VME memory address

```
MEM:VME:ADDR  #H250000    Set memory address location
```

Comments

- **Entering Address:** Address location can be specified in:

  Decimal or hexadecimal (#H....)

  MIN – sets the address to 2097152 (#H200000)

  MAX – sets the address to 14647294 (#HDF7FFE).

- ***RST Condition:** MEM:VME:ADDR #H200000

:VME:ADDRess?

MEMory:VME:ADDRess? [MINimum|MAXimum] returns one of the following numbers to the output buffer:

- The present hexadecimal address selected if MIN or MAX are not specified.
- The lowest hexadecimal address available (H200000) if MIN is specified.
- The highest hexadecimal address available (HDF7FFE) if MAX is specified.

Example

Querying the VME memory address

```
dimension statement     Dimension computer string array
MEM:VME:ADDR  #H250000  Set memory address location
MEM:VME:ADDR?           Query instrument to return
                        memory address (in decimal)
enter statement         Enter string into computer
```

:VME:MEASure:ADDRess

MEMory:VME:MEASure:ADDRess *<address>* sets the address of the external memory board in A24 memory address space where measurement data will be available. *address* must be on an even boundary or a settings conflict will be generated.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
address	numeric	2097152-14647294| #H200000-#HDF7FFE|MIN|MAX	none

Example **Setting the VME memory measure address**

```
MEM:VME:ADDR  #H250000    Set memory address location
```

Comments

- **Entering Address:** Address location can be specified in:

  Decimal or hexadecimal (#H....)

  MIN – sets the address to 2097152 (#H200000)

  MAX – sets the address to 14647294 (#HDF7FFE).

- ***RST Condition:** MEM:VME:ADDR #H200000

:VME:MEASure:ADDRess?

MEMory:VME:MEASure:ADDRess? [MINimum | MAXimum] returns one of the following numbers to the output buffer:

- The present hexadecimal address selected if MIN or MAX are not specified.
- The lowest hexadecimal address available (H200000) if MIN is specified.
- The highest hexadecimal address available (HDF7FFE) if MAX is specified.

Example **Querying the VME memory measure address**

```
dimension  statement     Dimension computer string array
MEM:VME:ADDR  #H250000   Set memory address location
MEM:VME:ADDR?            Query instrument to return
                         memory address (in decimal)
enter  statement         Enter string into computer
```

:VME:MEASure:STATe

MEMory:VME:MEASure:STATe ***<mode>*** **enables or disables use of the external memory board in A24 memory where data can be used for making a measurement.**

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	OFF|0|ON|1	none

Example

Enabling VME measure memory

```
MEM:VME:MEAS:ADDR  #H250000  Set measurement memory
                             address location
MEM:VME:MEAS:STAT  ON        Enable use of external
                             measurement data
```

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Memory VME State:** When MEMory:VME:STATe is set to ON, MEMory:VME:MEASure:STATe will automatically be set to ON.
- ***RST Condition:** MEM:VME:MEAS:STAT defaults to OFF.

:VME:MEASure:STATe?

MEMory:VME:MEASure:STATe? returns the current state of the VME measurement memory. ON is returned when the external memory board in A24 memory is being used for measurements. OFF is returned if the internal buffers are being used. The value is sent to the output buffer.

Example

Querying the VME measure memory state

```
MEM:VME:MEAS:STAT 1    Enables use of external memory card
MEM:VME:MEAS:STAT?     Query instrument to return external
                       memory state
enter  statement       Enter value into computer
```

:VME:SIZE

MEMory:VME:SIZE ***<bytes>*** **sets the size, in bytes, of the external VME memory card.**

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
bytes	numeric	#8000 through #HC00000	bytes

Example

Setting the VME memory size

```
MEM:VME:SIZE  64000
```
Set memory size to 64 kbytes

Comments

- **Entering Size:** Memory size can be specified in decimal or hexadecimal (#H....).
- **Minimum Memory Required:** A minimum of H8000 bytes of VME memory are required to use the external VME feature.
- ***RST Condition:** MEM:VME:SIZE defaults to H8000.

:VME:SIZE?

MEMory:VME:SIZE? returns the current external VME memory allocation (in hexadecimal) to the output buffer.

Example

Querying the VME memory size

`MEM:VME:SIZE  64000`	*Set memory size to 64 kbytes*
`MEM:VME:SIZE?`	*Query instrument to return memory size*
`enter  statement`	*Enter string into computer*

:VME:STATe

MEMory:VME:STATe *<mode>* enables or disables use of an external VME memory card for acquisition data storage.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	OFF|0|ON|1	none

Example

Enabling VME memory

```
MEM:VME:ADDR  #H250000   Set memory address location
MEM:VME:SIZE  64000      Set memory size to 64 kbytes
MEM:VME:STAT  ON         Enable use of external memory
                         card
```

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Memory VME Measure State:** When MEMory:VME:STATe is set to ON, MEMory:VME:MEASure:STATe will automatically be set to ON.
- ***RST Condition:** MEM:VME:STAT defaults to OFF.

:VME:STATe?

MEMory:VME:STATe? returns whether the external VME memory feature is enabled (ON) or disabled (OFF). The value is sent to the output buffer.

Example

Querying the VME memory state

```
MEM:VME:STAT  ON       Enables use of external memory card
MEM:VME:STAT?          Query instrument to return external
                       memory state
enter  statement       Enter value into computer
```

OUTPut

The OUTPut command subsystem selects the source of the output trigger generated when the Oscilloscope generates an internal trigger event. The selected output can be enabled, disabled, and queried. The three available outputs are the ECL Trigger bus (lines 0 to 1), TTL Trigger bus (lines 0 to 7), or the "TTL Trigger Output" BNC port on the Oscilloscope front panel.

OUTPut[:STATe] acts like the master switch for the OUTPut subsystem. If the ECLTrg, TTLTrg, or EXTernal states are on, an output will ONLY occur when the OUTPut[:STATe] is set to ON.

Subsystem Syntax

```
OUTPut
    :ECLTrg<number>
        [:STATe] <mode>
        [:STATe]?
    :EXTernal
        [:STATe] <mode>
        [:STATe]?
    [:STATe]
    [:STATe]?
    :TTLTrg<number>
        [:STATe] <mode>
        [:STATe]?
```

:ECLTrg[:STATe]

OUTPut:ECLTrg<*number*>[:STATe] <*mode*> selects and enables which ECL Trigger bus line (0 or 1) will output a trigger when the Oscilloscope triggers. It also is used to disable a selected ECL Trigger bus line. ***number*** specifies the ECL Trigger bus line (0 or 1). ***mode*** enables (ON | 1) or disables (OFF | 0) the specified bus line.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	0 or 1	none
mode	boolean	ON|OFF|1|0	none

Example

Enabling ECL trigger bus line 0

```
OUTP:ECLT0:STAT  1        Enable ECL Trigger bus line 0
OUTP  1                   Enable output subsystem
```

Comments

- **Enabling ECL Trigger bus:** When enabled, a pulse is output to the selected ECL Trigger bus line (0 or 1) when the Oscilloscope triggers. If disabled, a pulse is not output. The output is a positive going pulse.
- **Numerous outputs selected at a time:** All outputs (ECLTrg 0 or 1; TTLTrg 0, 1, 2, 3, 4, 5, 6, or 7; and EXTernal) can be enabled at one time. Before any output is enabled, both the individual trigger state and the output state must be enabled. For example, to enable TTLTrg1 and ECLTrg2, send the following commands:

  ```
  OUTP:TTLT1 ON
  OUTP:ECLT2 ON
  OUTP ON
  ```
- **Related Commands:** TRIGger subsystem.
- ***RST Condition:** Default is OFF.

:ECLTrg[:STATe]?

OUTPut:ECLTrg<*number*>[:STATe]? queries the present state of the specified ECL Trigger bus line. The query returns ON if the specified bus line is enabled or OFF if the specified bus line is disabled. The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	0 or 1	none

Example

Query ECL trigger bus line 0 state

```
dimension  statement      String for data
OUTP:ECLT0:STAT  1        Enable ECL Trigger bus line 0
OUTP:ECLT0?               Query instrument to return  ECL line
                          0 bus enable state
enter  statement          Enter result into computer
```

:EXTernal[:STATe]

OUTPut:EXTernal[:STATe] ***<mode>*** enables or disables the "TTL Trigger Output" BNC port on the Oscilloscope Module to output a trigger when the Oscilloscope triggers. *mode* enables (ON | 1) or disables (OFF | 0) the BNC port.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	ON|OFF|1|0	none

Example **Enabling the TTL trigger output connector**

`OUTP:EXT 1`	*Enable "TTL Trigger Output" BNC port to output pulse*
`OUTP 1`	*Enable output subsystem*

Comments

- **Enabling Trig Out Port:** When enabled, a pulse is output from the "TTL Trigger Output" BNC port on the Oscilloscope Module. If disabled, a pulse is not output. The output is a negative going pulse.
- **Numerous outputs selected at a time:** All outputs (ECLTrg 0 or 1; TTLTrg 0, 1, 2, 3, 4, 5, 6, or 7; and EXTernal) can be enabled at one time. Before any output is enabled, both the individual trigger state and the output state must be enabled. For example, to enable TTLTrgl and ECLTrg2, send the following commands:

```
OUTP:TTLT1 ON
OUTP:ECLT2 ON
OUTP ON
```

- **Related Commands:** TRIGger subsystem.
- ***RST Condition:** Defaults to OFF.

:EXTernal[:STATe]?

OUTPut:EXTernal:STATe? queries the present state of the "TTL Trigger Output" BNC port. The query returns ON if the port is enabled or OFF if the port is disabled. The value is sent to the output buffer.

Example **Query TTL trigger output connector enable state**

`dimension statement`	*String for data*
`OUTP:EXT ON`	*Enable "TTL Trigger Output" BNC port*
`OUTP:EXT?`	*Query instrument to return port enable state*
`enter statement`	*Enter value into computer*

[:STATe]

OUTPut[:STATe] ***<mode>*** **enables or disables the OUTPut subsystem.** ***mode*** **enables (ON | 1) or disables (OFF | 0) all selected TTLTrg, ECLTrg, and EXTernal outputs.**

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	ON|OFF|1|0	none

Example **Enabling TTL trigger output connector**

```
OUTP:EXT  1          Enable "TTL Trigger Output" BNC
                     port to output pulse
OUTP  1              Enable output subsystem
```

Comments

- **Selecting Outputs:** Use the TTLTrg, ECLTrg, or EXTernal commands to enable a specific output. Use the OUTPut[:STATe] command to enable the subsystem.
- ***RST Condition:** Defaults to OFF.

[:STATe]?

OUTPut[:STATe]? queries the present state of the OUTPut subsystem. The query returns ON if the output is enabled or OFF if the output is disabled. The value is sent to the output buffer. See [:STATe] command for more information.

Example **Query output subsystem state**

```
dimension  statement   String for data
OUTP  1                Enable "Trigger Output" BNC port
OUTP?                  Query instrument to return port
                       enable state
enter  statement       Enter value into computer
```

:TTLTrg[:STATe]

OUTPut:TTLTrg*<number>*[:STATe] *<mode>* selects and enables which TTL Trigger bus line (0 to 7) will output a trigger when the Oscilloscope triggers. It also is used to disable a selected TTL Trigger bus line. *mode* enables (ON | 1) or disables (OFF | 0) the specified bus line.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	0 to 7	none
mode	boolean	ON|OFF|1|0	none

Example

Enabling TTL Trigger bus line 7

```
OUTP:TTLT7:STAT  1      Enable TTL Trigger bus line 7 to
                        output pulse
OUTP  1                 Enable output subsystem
```

Comments

- **Enabling TTL Trigger bus:** When enabled, a pulse is output to the selected TTL Trigger bus line (0 to 7) after the Oscilloscope triggers. If disabled, a pulse is not output. The output is a negative going pulse.
- **Numerous outputs selected at a time:** All outputs (ECLTrg 0 or 1; TTLTrg 0, 1, 2, 3, 4, 5, 6, or 7; and EXTernal) can be enabled at one time. Before any output is enabled, both the individual trigger state and the output state must be enabled. For example, to enable TTLTrgl and ECLTrg2, send the following commands:

```
OUTP:TTLT1 ON
OUTP:ECLT2 ON
OUTP ON
```

- **Related Commands:** TRIGger subsystem.
- ***RST Condition:** Defaults to OFF.

:TTLTrg[:STATe]?

OUTPut:TTLTrg*<number>*[:STATe]? queries the present state of the specified TTL Trigger bus line. The query returns ON if the specified bus line is enabled or OFF if the specified bus line is disabled. The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	0 or 7	none

Example

Query TTL trigger bus line 7 state

```
dimension  statement    String for data
OUTP:TTLT7:STAT  1      Enable TTL Trigger bus line 7
OUTP:TTLT7?             Query instrument  to return TTL bus
                        line 7 enable state
enter  statement        Enter value into computer
```

READ?

The READ? query is used to initiate a previously configured measurement, and then transfer the measurement results to the output buffer. The READ? query performs the identical function as the INITiate and FETCh? commands.

Subsystem Syntax

READ[[:SCALar]:VOLTage[:<*function*>]]?

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
function	discrete	[AC|AMPLitude|[DC]|DCYCle| FALL:OVERshoot|FALL:PREShoot| FALL:TIME|FREQuency|FTIMe| HIGH|LOW|MAXimum|MINimum| NDUTycycle|NWIDth|PDUTycycle| PERiod|PWIDth|RISE:OVERshoot| RISE:PREShoot|RISE:TIME|RTIMe| TMAXimum|TMINimum	none

Comments

- **Selecting Function:** Depending on the desired action, *function* can either be omitted or specified as follows:

  READ?: When executed, will initiate and retrieve results from the last measurement. If *RST is executed prior to sending the READ?, an error will be generated.

  READ[:SCALar]:VOLTage:<*function*>?: When executed, will initiate and retrieve the results from a previously configured measurement as specified using *function*.

  For example,

`dimension statement`	*String for data*
`CONF:VOLT:FREQ (@1)`	*Configure input 1 for a frequency measurement*
`MEAS:VOLT:AC? (@2)`	*Perform an AC voltage measurement on input 2*
`CONF:VOLT:PER (@3)`	*Configure input 3 for a period measurement*
`ABOR`	*Stop all measurements*
`READ?`	***Would initiate an AC voltage measurement, then return measurement results***
OR	
`READ:VOLT:FREQ?`	***Would initiate a frequency measurement, then return measurement results***
`enter statement`	*Enter measurement results into computer*

- **READ? versus FETCh? and INITiate:** Performing a measurement using the READ? query is identical to performing the INITiate/FETCh? commands.
- **Return Format:** Previous data stored in the output buffer is lost when a READ? is executed. A returned 9.99999E+37 indicates an invalid measurement. See the individual CONFigure commands for more information on returned measurement results.

[SENSe:]

The [SENSe:] command subsystem is used to setup the instrument's vertical, horizontal, and data acquisition controls. The [SENSe:] subsystem is comprised of five lower level subsystems shown below.

Subsystem Syntax

```
[SENSe:]
    AVERage
        :COUNt <count>
        :COUNt?
        [:STATe] <mode>
        [:STATe]?
        :TYPE <mode>
        :TYPE?
    CORRection<number>
        :AFACtor <atten>
        :AFACtor?
    INPut<number>
        :COUPling <type>
        :COUPling?
        :FILTer
            [:LPASs]
                [:STATe] <mode>
            [:LPASs]
                [:STATe]?
            :HPASs
                [:STATe] <mode>
            :HPASs
                [:STATe]?
        :IMPedance <value>
        :IMPedance?
        [:STATe] <mode>
        [:STATe]?
    SWEep
        :POINts <points>
        :POINts?
            :COMPlete <complete>
            :COMPlete?
        :TIME:
            :CENTer <center_time>
            :CENTer?
            :DELay <time>
            :DELay?
                :LINK <reference>
                :LINK?
            :RANGe <range>
            :RANGe?
            :SPAN <span>
            :SPAN?
            :STARt <start_time>
            :STARt?
            :STOP <stop_time>
            :STOP?
```

Subsystem Syntax

```
[SENSe:]—Continued
    VOLTage<number>
        :RANGe
            :LOWer <lower>
            :LOWer?
            :OFFSet <value>
            OFFSet?
            [:PTPeak] <range>
            [:PTPeak]?
            :UPPer <upper>
            :UPPer?
```

AVERage

[SENSe:]AVERage subsystem is used to select the type of data and number of averages when acquiring waveform data. Must be setup prior to executing the INITiate[:IMMediate] command when digitizing waveform data.

Note

The term "Time Buckets" is defined as — the time range divided into a specific number of horizontal time points (as defined by [SENSe:]SWEep:POINts command), each with an equal and fixed time associated with it.

AVERage:COUNt

[SENSe:]AVERage:COUNt *<count>* selects the number of values to be averaged for each time bucket before the acquisition (for that time bucket) is considered complete.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
count	numeric	1 to 2048	none

Example

Set average count to 64

The following example illustrates the use of the AVERage:COUNt command only. Chapter 5 contains an example on performing a complete digitizing operation.

```
AVER:COUN  64                 Average count is 64
```

Comments

- **Average Type:** Count values accepted are dependent on the AVERage[:STATe] and/or the AVERage:TYPE currently selected, as follows:

  AVERage[:STATe] ON: This is the Average mode. The acceptable values are from 1 to 2048, however entry will be rounded to the nearest power of 2. Selection of AVERage:TYPE is not required. Any value entered outside the range will automatically be adjusted to the nearest acceptable value.

  AVERage[:STATe] OFF — AVERage:TYPE SCALar: This is the Scalar mode. Count is not used in this mode. Any value from 1 to 2048 can be entered, but is disregarded. Query will always return a 1.

  AVERage[:STATe] OFF — AVERage:TYPE ENVelope: This is the Envelope mode. Count is not used in this mode. Any value from 1 to 2048 can be entered, but is disregarded. Query will return entered value.

- **Related Commands:** [SENSe:]AVERage:TYPE, [SENSe:]SWEep:POINts.
- ***RST Condition:** Defaults to 8.

AVERage:COUNt?

[SENSe:]AVERage:COUNt? returns the currently selected count value. The value is sent to the output buffer.

Example

Querying average count

```
AVER:COUN  64                 Average count is 64
AVER:COUN?                    Query instrument to return average
                              count value
enter  statement              Enter value into computer
```

Comments

- **Average Type Scalar:** When AVERage[:STATe] is OFF and AVERage:TYPE SCALar is selected, a count query will always return a 1.

AVERage[:STATe]

[SENSe:]AVERage[:STATe] *<mode>* is used to select the average acquisition mode. Also used with the AVERage:TYPE command to select the other acquisition modes.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	ON|OFF|1|0	none

Example

Enabling average acquisition mode

The following example illustrates the use of the AVERage:STATe command only. Chapter 5 contains an example on performing a complete digitizing operation.

```
AVER  ON                    Enable average acquisition mode
```

Comments

- **Selecting Acquisition Mode:** Acquisition mode is selected using the AVERage:TYPE and AVERage[:STATe] commands as follows:

  AVERage Mode: The average acquisition mode is selected when AVERage[:STATe] is enabled (ON | 1). AVERage:TYPE is not used (overridden), however when AVERage[:STATe] is set to OFF, TYPE is automatically set to SCALar.

  SCALar or ENVelope Mode: The Scalar and Envelope acquisition modes are selected when AVERage[:STATe] is disabled (OFF | 0). The AVERage:TYPE command then selects the ENVelope or SCALar acquisition mode.

- **AVERage Mode:** Average acquisition mode is used when reduction of signal noise and improved resolution is desired. The waveform reflects a minimum of *n* acquisitions averaged per time bucket, where *n* is the current AVERage:COUNt value.
- **Related Commands:** [SENSe:]AVERage:TYPE, [SENSe:]AVERage:COUNt.
- ***RST Condition:** Defaults to OFF.

AVERage[:STATe]?

[SENSe:]AVERage[:STATe]? queries the present state of the average acquisition mode. The query returns ON if the average mode enabled or OFF if the average mode is disabled. The value is sent to the output buffer.

Example

Query average mode state

```
dimension  statement        String for data
AVER  ON                    Enable average acquisition mode
AVER?                       Query instrument to return average
                            mode state
enter  statement            Enter value into computer
```

AVERage:TYPE

[SENSe:]AVERage:TYPE ***<mode>*** is used to select the scalar or envelope acquisition mode. Used only when AVERage[:STATe] is OFF.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	SCALar|ENVelope	none

Example

Set acquisition type to envelope

The following example illustrates the use of the AVERage:TYPE command only. Chapter 5 contains an example on performing a complete digitizing operation.

```
AVER:TYPE  ENV
```
Acquisition type is envelope

Comments

- **Selecting Acquisition Mode:** Acquisition mode is selected using the AVERage:TYPE and AVERage[:STATe] commands as follows:

  AVERage Mode: The average acquisition mode is selected when AVERage[:STATe] is enabled (ON | 1). AVERage:TYPE is not used (overridden), however when AVERage[:STATe] is set to OFF, TYPE is automatically set to SCALar.

  SCALar Mode: The Scalar acquisition mode is selected when AVERage[:STATe] is disabled (OFF | 0), and AVERage:TYPE is SCALar.

  ENVelope Mode: The Envelope acquisition mode is selected when AVERage[:STATe] is disabled (OFF | 0), and AVERage:TYPE is ENVelope.

- **Why Three Modes?:** Mode is used to select how the acquisitions are used when generating the waveform.

  SCALar: Used for general purpose type measurements. The waveform reflects the last data point (hit) in each time bucket. AVERage:COUNt has no effect in this mode.

  AVERage: Used when reduction of signal noise and improved resolution is desired. The waveform reflects a minimum of *n* acquisitions averaged per time bucket, where *n* is the current AVERage:COUNt value.

  ENVelope: Used when measuring voltage or time jitter. The waveform reflects the minimum and maximum data points (hit) in each time bucket. AVERage:COUNt has no effect in this mode.

- **Related Commands:** AVERage:COUNt, INITiate[:IMMediate].
- ***RST Condition:** Defaults to SCALar.

AVERage:TYPE?

[SENSe:]AVERage:TYPE? returns the currently selected acquisition type. Does not return active acquisition mode. The data is sent to the output buffer. See AVERage:TYPE and AVERage[:STATe] commands for more information on available types.

Example

Querying acquisition type

```
Dimension  statement   String for data
AVER  OFF              Average acquisition to off
AVER:TYPE  ENV         Acquisition type is envelope
AVER:TYPE?             Query instrument to return
                       acquisition type
enter  statement       Enter value into computer
```

CORRection

[SENSe:]CORRection<*number*> subsystem is used to select a specific input's probe attenuation factor. Inputs 1, 2, 3, and 4 are independently programmable.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Comments

- **Entering Number:** The desired input *number* from 1 to 4 is specified for the correction commands listed in this section.

CORRection:AFACtor

[SENSe:]CORRection*<number>*:AFACtor *<atten>* is used to enter a probe's attenuation factor for the input specified. The selection does not change the actual input sensitivity of the instrument, it changes the reference constants for scaling the vertical range and offset, automatic measurements, trigger levels, etc.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none
atten	numeric	0.9 to 1000.0	none

Example

Set input 1 probe attenuation to 10:1

```
CORR1:AFAC  10        Input 1 probe attenuation to 10:1
```

Comments

- **Entering Attenuation:** If *atten* is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.
- **Range and Offset:** Changing CORRection<n>:AFACtor will effect the current settings of VOLTage<n>:RANGe[:PTPeak] and OFFSet.
- **Related Commands:** CALibration:PCALibration:ATTenuation.
- ***RST Conditions:** Defaults to 1:1 on all inputs.

CORRection:AFACtor?

[SENSe:]CORRection*<number>*:AFACtor? returns the current probe attenuation factor for the input specified. The value (a ratio :1) is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example

Querying input 1 probe attenuation

```
CORR1:AFAC  10        Input 1 probe attenuation to 10:1
CORR1:AFAC?           Query instrument to return input 1
                      probe attenuation factor
enter  statement      Enter value into computer
```

INPut

[SENSe:]INPut<*number*> subsystem is used to select a specific input's coupling, impedance, filter, and on/off functions. Inputs 1, 2, 3, and 4 are independently programmable.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Comments

- **Entering Number:** The desired input *number* from 1 to 4 is specified for the input commands listed in this section.

INPut:COUPling

[SENSe:]INPut<*number*>:COUPling <*type*> is used to select the coupling for the input specified. The coupling for each input can be set to AC or DC.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number *type*	numeric discrete	1 to 4 AC|DC	None None

Example **Set input 1 coupling to AC**

```
INP1:COUP  AC               Input 1 coupling to AC
```

Comments

- **Coupling versus IMPedance:** AC coupling is not available when the IMPedance is set to 50Ω. Setting coupling to AC sets impedance to 1MΩ.
- **Coupling versus FILTer:** DC coupling is not available when the high pass filter is enabled.
- **Related Commands:** [SENSe:]INPut<n>:IMPedance, [SENSe:]INPut<n>:FILTer:HPASs.
- ***RST Condition:** Defaults to DC on all inputs.

INPut:COUPling?

[SENSe:]INPut<*number*>:COUPling? returns the currently selected coupling type (AC or DC) for the input specified. The data is sent to the output buffer. See INPut<n>:COUPling for more information on coupling types.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example **Querying input 1 coupling**

```
dimension  statement   String for data
INP1:COUP  AC          Input 1 coupling to AC
INP1:COUP?             Query instrument to return input 1
                       coupling selection
enter  statement       Enter value into computer
```

INPut:FILTer:HPASs[:STATe]

[SENSe:]INPut<*number*>:FILTer:HPASs[:STATe] <*mode*> is used to select an internal high pass filter. When ON, the bandwidth of the specified input is limited to approximately 450 Hz. The bandwidth limit filter may be used only with AC coupling.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none
mode	boolean	OFF|0|ON|1	none

Example **Enable high pass filter on input 3**

```
INP3:FILT:HPAS  ON      Input 3 high pass filter to on
```

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Related Commands:** [SENSe:]INPut<n>:COUPling.
- ***RST Conditions:** Defaults to OFF for all inputs.

INPut:FILTer:HPASs[:STATe]?

[SENSe:]INPut<*number*>:FILTer:HPASs[:STATe]? returns the currently selected high pass filter state (ON or OFF) for the input specified. The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example **Querying input 3 high pass filter state**

```
dimension  statement    String for data
INP3:FILT:HPAS  ON      Input 3 high pass filter to on
INP3:FILT:HPAS?         Query instrument to return input 3
                        high pass filter state
enter  statement        Enter value into computer
```

INPut:FILTer[:LPASs][:STATe]

[SENSe:]INPut*<number>*:FILTer[:LPASs][:STATe] ***<mode>*** is used to select an internal low pass filter. When ON, the bandwidth of the specified input is limited to approximately 30 MHz. The bandwidth limit filter may be used with all coupling selections.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none
mode	boolean	OFF|0|ON|1	none

Example **Enable low pass filter on input 1**

```
INP1:FILT  1              Input 1 low pass filter to on
```

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Related Commands:** [SENSe:]INPut<n>:COUPling.
- ***RST Conditions:** Defaults to OFF for all inputs.

INPut:FILTer[:LPASs][:STATe]?

[SENSe:]INPut*<number>*:FILTer[:LPASs][STATe]? returns the currently selected low pass filter state (ON or OFF) for the input specified. The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example **Querying input 1 low pass filter state**

```
dimension  statement   String for data
INP1:FILT  1           Input 1 low pass filter to on
INP1:FILT?             Query instrument to return input 1
                       low pass filter state
enter  statement       Enter value into computer
```

INPut:IMPedance

[SENSe:]INPut<*number*>:IMPedance <*value*> is used to select the impedance for the input specified. The impedance for each input can be set to 1MΩ or 50Ω. The 50Ω impedance selection may be used only with DC coupling.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	None
value	numeric	50|1E6	OHM

Example

Set input 1 impedance to 50Ω

```
INP1:IMP  50
```
Input 1 impedance to 50Ω

Comments

- **Coupling versus IMPedance:** Setting impedance to 50Ω automatically sets coupling to DC.
- ***RST Condition:** Defaults to 1MΩ on all inputs.
- **Related Commands:** [SENSe:]INPut<n>:COUPling.

CAUTION

Because the 50Ω selection sinks more current than the 1MΩ setting, make sure you do not exceed the maximum rated input of the channel when switching from 1MΩ to 50Ω.

INPut:IMPedance?

[SENSe:]INPut<*number*>:IMPedance? returns the currently selected input impedance (50 or 1E6) for the input specified. The data is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example

Querying input 1 coupling

```
INP1:IMP  50        Input 1 impedance to 50Ω
INP1:IMP?           Query instrument to return input 1
                    impedance selection
enter  statement    Enter value into computer
```

INPut[:STATe]

[SENSe:]INPut<*number*>[:STATe] <*mode*> is used to enable (ON | 1) or disable (OFF | 0) the specified input.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none
mode	boolean	ON|OFF|1|0	none

Example **Enabling input 4**

```
INP4:STAT 1            Enable input 4
```

Comments

- **Related Commands:** [SENSe:]INPut<n>[:STATe]?
- ***RST Condition:** Defaults to input 1 on, input 2-4 off.

INPut[:STATe]?

[SENSe:]INPut<*number*>[:STATe]? queries the present state of the specified input. The query returns ON if the specified input is enabled or OFF if the specified input is disabled. The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example **Query input 4 state**

```
INP4:STAT 1            Enable input 4
INP4?                  Query instrument to return input 4
                       state
enter statement        Enter value into computer
```

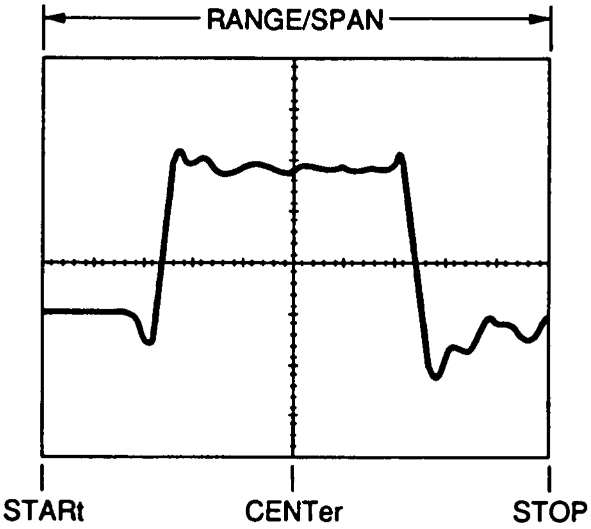
SWEep

[SENSe:]SWEep subsystem is used to control the horizontal axis, or "X-axis," functions.

Note

The term "Time Buckets" is defined as — the time range divided into a specific number of horizontal time points (as defined by SWEep:POINts command), each with an equal and fixed time associated with it.

For purposes of selecting the SWEep commands STARt, STOP, CENTer, RANGe/SPAN are as shown below.



SWEep:POINts

[SENSe:]SWEep:POINts *<points>* selects the number of time buckets for each acquisition.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
points	numeric	32|64|128|256|500|512|1024	none

Example

Set acquisition points to 1024

The following example illustrates the use of the SWEep:POINts command only. Chapter 5 contains an example on performing a complete digitizing operation.

```
SWE:POIN  1024          Acquisition points are 1024
```

Comments

- **Entering Points:** Any value between 32 and 1024 can be entered, however entry will be rounded to the nearest acceptable value.
- **Waveform Points:** To determine the ACTUAL number of time buckets acquired, send the TRACe:POINts? query.
- **Related Commands:** TRACe:POINts?.
- ***RST Condition:** Defaults to 500.

SWEep:POINts?

[SENSe:]SWEep:POINts? returns the currently selected points value. The value is sent to the output buffer.

Example

Querying acquisition points

```
SWE:POIN  1024          Acquisition points are 1024
SWE:POIN?               Query instrument to return
                        acquisition points value
enter  statement        Enter value into computer
```

SWEep:POINts:COMPlete

[SENSe:]SWEep:POINts:COMPlete ***<complete>*** specifies the completion criteria for an acquisition. Specifies what percentage of the time buckets need to be "full" before an acquisition is considered complete.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
complete	numeric	0 to 100	PCT

Example **Set acquisition complete to 50%**

The following example illustrates the use of the SWEep:POINts:COMPlete command only. Chapter 5 contains an example on performing a complete digitizing operation.

```
SWE:POIN:COMP  50        Acquisition complete is 50%
```

Comments

- **Time Buckets = "full":** A time bucket is considered "full" dependent on the acquisition mode selected as follows:

  SCALar Mode: The instrument only needs one data point per time bucket for that time bucket to be considered full.

  AVERage or ENVelope Mode: A specified number of data points per time bucket (set using AVERage:COUNt) must be acquired.

- **Recommended Completion Value:** 60% is the recommended completion criteria. See Appendix C, Optimizing Measurements, for additional information on measurement techniques.
- **Completion of 0%:** If the complete value is set to 0, then one acquisition cycle will take place.
- **Related Commands:** [SENSe:]AVERage:TYPE, [SENSe:]AVERage:COUNt.
- ***RST Condition:** Defaults to 100%.

SWEep:POINts:COMPlete?

[SENSe:]SWEep:POINts:COMPlete? returns the completion value (in percent) for the currently selected acquisition mode. The value is sent to the output buffer. See SWEep:POINts:COMPlete command for more information.

Example **Querying current acquisition complete value**

```
SWE:POIN:COMP  50        Acquisition complete is 50%
SWE:POIN:COMP?           Query instrument to return
                         acquisition complete value
enter  statement         Enter value into computer
```

SWEep:TIME:CENTer

[SENSe:]SWEep:TIME:CENTer *<center_time>* is used to set the time interval (in seconds) between the trigger event and the center of the currently specified range/span. The range is set to a specific time using the SWEep:TIME:RANGe or SPAN command, and does not change.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
center_time	numeric	Depends on SWEep:TIME:RANGe	S

Example

Set center time to 2 msec

```
SWE:TIME:CENT   2E-3     Center to 2 msec
```

Comments

- **Entering Center Time:** *center_time* selects the time at the center of the selected SWEep:TIME:RANGe or SPAN.
- **Effects on Other SWEep Selections:** Center changes will cause the following commands to change their current parameters:

  SWEep:DELay: Amount of change is proportional to the amount that center was changed.

  SWEep:TIME:STARt: New start time is calculated and entered as follows:

  start = center – (span/2)

  SWEep:TIME:STOP: New stop time is calculated and entered as follows:

  stop = center + (span/2)
- **Entering SWEep:TIME:RANGE or SPAN:** Range/span CANNOT be entered or changed using the TIME:STARt, STOP, or CENTer commands.
- **TIME:STARt/:STOP/:CENTer versus DELay:** Setting TIME:STARt, TIME:STOP, and TIME:CENTer is identical to setting the same values using DELay and TIME:DELay:LINK commands.
- **Related Commands:** [SENSe:]SWEep:DELay, [SENSe:]SWEep:TIME:(STARt and STOP).
- ***RST Condition:** Defaults to 0 sec.

SWEep:TIME:CENTer?

[SENSe:]SWEep:TIME:CENTer? returns a number representing the current time interval between the trigger event and the center of the currently specified range/span. The value (in ± seconds) is sent to the output buffer.

Example

Querying current center time value

```
SWE:TIME:CENT   2E-3     Center to 2 msec
SWE:TIME:CENT?           Query instrument to return center
                         time value in seconds
enter  statement         Enter value into computer
```

SWEep:TIME:DELay

[SENSe:]SWEep:TIME:DELay *<time>* is used to set the time interval between the trigger event and the delay reference point. The delay reference point is set to the STARt, CENTer, or STOP position of the waveform using the SWEep:TIME:DELay:LINK command.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
time	numeric	Depends on SWEep:TIME:RANGe	seconds

Example

Set the time interval between the trigger event and the delay reference point to 2 msec

```
SWE:TIME:DEL  2E-3
```
Set delay to 2 msec

Comments

- **Entering Time:** When 0 is entered, the trigger event occurs at the delay reference point. Positive values set the trigger event to occur before the delay reference point (to capture post-trigger events). Negative values set the trigger event to occur after the delay reference point (to capture pre-trigger events). The range of acceptable DELay values is dependent on the current SWEep:TIME:RANGe setting. If DELay is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.
- **DELay versus TIME:STARt/:STOP/:CENTer:** Setting DELay and TIME:DELay:LINK is identical to setting the same values using the TIME:STARt, TIME:STOP, and TIME:CENTer commands.
- **Related Commands:** [SENSe:]SWEep:TIME.
- ***RST Condition:** Defaults to 0 seconds.

SWEep:TIME:DELay?

[SENSe:]SWEep:TIME:DELay? returns a number representing the current time interval between the trigger event and the delay reference point. The value (in ± seconds) is sent to the output buffer.

Example

Querying current delay value

`SWE:TIME:DEL  2E-3`	*Set delay to 2 msec*
`SWE:TIME:DEL?`	*Query instrument to return delay value in seconds*
`enter  statement`	*Enter value into computer*

SWEep:TIME:DELay:LINK

[SENSe:]SWEep:TIME:DELay:LINK *<position>* sets the delay reference to the start, stop, or to the center of the active waveform.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
position	discrete	STARt|CENTer|STOP	none

Example **Set the reference to the start (left side) of the waveform**

```
SWE:TIME:DEL:LINK  STAR
```
Reference to start (post-trigger)

Comments

- **Selecting Position:** Position entered is used with the SWEep:DELay command to set the time interval between the trigger event and the delay reference point. For example, if DELay is 0 seconds, and LINK is CENTer, pre-trigger data is on the left and post-trigger data is on the right of the waveform.
- **Effects on Other SWEep Selections:** A change in LINK will cause the following SWEep commands to change their current parameters:

  TIME:STARt, TIME:STOP, TIME:CENTer
- **Related Commands:** [SENSe:]SWEep:DELay, [SENSe:]SWEep:TIME:(STARt, STOP, and CENTer).

SWEep:TIME:DELay:LINK?

[SENSe:]SWEep:TIME:DELay:LINK? returns the currently selected delay reference point. The data is sent to the output buffer. Returns STARt, CENTer, or STOP depending on the current position selected. See SWEep:TIME:DELay:LINK command for more information.

Example **Query current reference point selection**

`Dimension statement`	*String for data*
`SWE:TIME:DEL:LINK STAR`	*Reference to start*
`SWE:TIME:DEL:LINK?`	*Query instrument to return position*
`enter statement`	*Enter data into computer*

SWEep:TIME:RANGe

[SENSe:]SWEep:TIME:RANGe *<range>* is used to define the full scale horizontal axis, or "X-axis" of the main sweep.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
range	numeric	2NS to 50S	S

Example

Set horizontal range to 2 msec (full scale)

```
SWE:TIME:RANG  2E-3     Range to 2 msec
```

Comments

- **Entering Range:** Range values are entered in a 1,2,5 sequence. If a value is entered that is not in a 1,2,5 sequence, it is automatically rounded to the closest allowable value without generating an error.
- **Effects on Other SWEep Selections:** A change in range may cause the following SWEep commands to change their current parameters:

  DELay, TIME:STARt, TIME:STOP, TIME:CENTer
- **SWEep:TIME:RANGe versus SWEep:TIME:SPAN:** Both commands perform the identical function.
- **Related Commands:** [SENSe:]SWEep:DELay, [SENSe:]SWEep:TIME:(SPAN, STARt, STOP, CENTER, DELay:LINK).
- ***RST Condition:** Defaults to 1 msec.

SWEep:TIME:RANGe?

[SENSe:]SWEep:TIME:RANGe? returns a numeric value representing the current range setting for the horizontal axis. The value (in seconds) is sent to the output buffer.

Example

Querying full scale horizontal range setting

```
SWE:TIME:RANG  2E-3     Range to 2 msec
SWE:TIME:RANG?          Query instrument to return current
                        range setting
enter  statement        Enter value into computer
```

SWEep:TIME:SPAN

[SENSe:]SWEep:TIME:SPAN *<span>* is used to define the full scale horizontal axis, or "X-axis" of the main sweep.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
range	numeric	2NS to 50S	S

Example

Set horizontal span to 2 msec (full scale)

```
SWE:TIME:SPAN  2E-3
```
Span to 2 msec

Comments

- **Entering Span:** Span values are entered in a 1,2,5 sequence. If a value is entered that is not in a 1,2,5 sequence, it is automatically rounded to the closest allowable value without generating an error.
- **Effects on Other SWEep Selections:** A change in span may cause the following SWEep commands to change their current parameters:

  DELay, TIME:STARt, TIME:STOP, TIME:CENTer
- **SWEep:TIME:SPAN versus SWEep:TIME:RANGe:** Both commands perform the identical function.
- **Related Commands:** [SENSe:]SWEep:DELay, [SENSe:]SWEep:TIME:(RANGe, STARt, STOP, CENTER, DELay:LINK).
- ***RST Condition:** Defaults to 1 msec.

SWEep:TIME:SPAN?

[SENSe:]SWEep:TIME:SPAN? returns a numeric value representing the current span setting for the horizontal axis. The value (in seconds) is sent to the output buffer.

Example

Querying full scale horizontal span setting

`SWE:TIME:SPAN  2E-3`	*Span to 2 msec*
`SWE:TIME:SPAN?`	*Query instrument to return current span setting*
`enter statement`	*Enter value into computer*

SWEep:TIME:STARt

[SENSe:]SWEep:TIME:STARt *<start_time>* is used to set the time interval (in seconds) between the trigger event and the start of the currently specified range/span. The range is set to a specific time using the SWEep:TIME:RANGe or SPAN command, and does not change.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
start_time	numeric	Depends on SWEep:TIME:RANGe	S

Example

Set start time to 1μsec

```
SWE:TIME:STAR  1E-6      Start to 1μsec
```

Comments

- **Entering Start Time:** *start_time* specifies the starting time of the selected SWEep:TIME:RANGe or SPAN.
- **Effects on Other SWEep Selections:** Start changes will cause the following commands to change their current parameters:

  SWEep:DELay: Amount of change is proportional to the amount that start was changed.

  SWEep:TIME:CENTer: New center time is calculated and entered as follows:

  center = (start + stop)/2

  SWEep:TIME:STOP: New stop time is calculated and entered as follows:

  stop = center + (span/2)
- **Entering SWEep:TIME:RANGE or SPAN:** Range/span CANNOT be entered or changed using the TIME:STARt, STOP, or CENTer commands.
- **TIME:STARt/:STOP/:CENTer versus DELay:** Setting TIME:STARt, TIME:STOP, and TIME:CENTer is identical to setting the same values using DELay and TIME:DELay:LINK commands.
- **Related Commands:** [SENSe:]SWEep:DELay, [SENSe:]SWEep:TIME:(CENTer and STOP).
- ***RST Condition:** Defaults to –500 μsec.

SWEep:TIME:STARt?

[SENSe:]SWEep:TIME:STARt? returns a number representing the current time interval between the trigger event and the start of the currently specified range/span. The value (in ± seconds) is sent to the output buffer.

Example

Querying current start time value

```
SWE:TIME:STAR  1E-6      Start to 1μsec
SWE:TIME:STAR?           Query instrument to return start time
                         value in seconds
enter  statement         Enter value into computer
```

SWEep:TIME:STOP

[SENSe:]SWEep:TIME:STOP *<stop_time>* is used to set the time interval (in seconds) between the trigger event and the stop of the currently specified range/span. The range/span is set to a specific time using the SWEep:TIME:RANGe or SPAN command, and does not change.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
stop_time	numeric	Depends on SWEep:TIME:RANGe	S

Example

Set stop time to 10µsec

```
SWE:TIME:STOP  10E-6   Stop to 10µsec
```

Comments

- **Entering Stop Time:** *stop_time* specifies the stopping time of the selected SWEep:TIME:RANGe or SPAN.
- **Effects on Other SWEep Selections:** Stop changes will cause the following commands to change their current parameters:

  SWEep:Delay: Amount of change is proportional to the amount that stop was changed.

  SWEep:TIME:CENTer: New center time is calculated and entered as follows:

  center = (start + stop)/2

  SWEep:TIME:STARt: New start time is calculated and entered as follows:

  start = center – (span/2)
- **Entering SWEep:TIME:RANGE or SPAN:** Range/span <u>CANNOT</u> be entered or changed using the TIME:STARt, STOP, or CENTer commands.
- **TIME:STARt/:STOP/:CENTer versus DELay:** Setting TIME:STARt, TIME:STOP, and TIME:CENTer is identical to setting the same values using DELay and TIME:DELay:LINK commands.
- **Related Commands:** [SENSe:]SWEep:DELay, [SENSe:]SWEep:TIME:(CENTer and STARt).
- ***RST Condition:** Defaults to 500 µsec.

SWEep:TIME:STOP?

[SENSe:]SWEep:TIME:STOP? returns a number representing the current time interval between the trigger event and the stop of the currently specified range/span. The value (in ± seconds) is sent to the output buffer.

Example

Querying current stop time value

```
SWE:TIME:STOP  10E-6   Stop to 10µsec
SWE:TIME:STOP?         Query instrument to return stop time
                       value in seconds
enter  statement       Enter value into computer
```

VOLTage

[SENSe:]VOLTage<*number*> subsystem is used to select a specific inputs vertical range and offset. Inputs 1, 2, 3, and 4 are independently programmable.

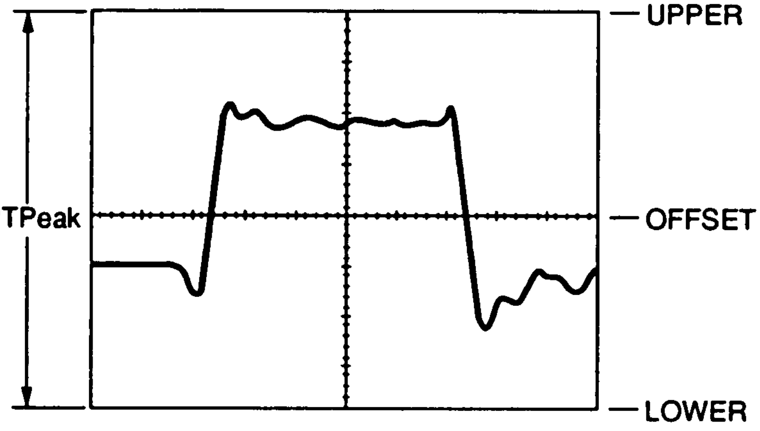
Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Comments

- **Entering Number:** The desired input *number* from 1 to 4 is specified for the input commands listed in this section.

For purposes of selecting the VOLTage commands PTPeak, LOWer, OFFSet, and UPPer are as shown below.



VOLTage:RANGe:LOWer

[SENSe:]VOLTage<*number*>:RANGe:LOWer <*lower*> sets the voltage that is represented at the lower end of the range for the selected input number.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
lower	numeric	Depends on VOLTage<n>:RANGe[:PTPeak]	V

Example

Set input 2 lower to 1 volt

```
VOLT2:RANG:LOW  1
```
Lower limit to 1 volt

Comments

- **Entering Lower:** The range of acceptable LOWer values are dependent on the current VOLTage<*n*>:RANGe[:PTPeak] setting LOWer can be calculated using the values given below and the following formula:

  LOWer = OFFSet ± (PTPeak/2)

VOLTage<n>:RANGe[:PTPeak]	VOLTage<n>:OFFSet limits
8 mV to 400 mV	±2 V ± (PTPeak/2)
>400 mV to 2.0 V	±10 V ± (PTPeak/2)
>2.0 V to 10.0 V	±50 V ± (PTPeak/2)
>10.0 V to 40.0 V	±250 V ± (PTPeak/2)

  For example, if PTPeak is set to 400 mV, then

  Minimum LOWer= –2 V – (400 mV/2) = –2 V – 200 mV = –2.2 V

  If LOWer is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.

- **Effects on Other VOLTage Selections:** Lower changes will cause the following commands to change their current parameters:

  VOLTage<n>:RANGe:UPPer: New upper value is calculated and entered as follows:

  upper = lower + PTPeak

  VOLTage<n>:RANGe:OFFSet: New OFFSet value is calculated and entered as follows:

  offset = center of upper and lower

- **Entering VOLTage<n>:RANGe[:PTPeak]:** PTPeak CANNOT be entered or changed using the VOLTage<n>:RANGe:UPPer/:LOWer commands.
- **Related Commands:** [SENSe:]VOLTage<n>:RANGe ([:PTPeak], OFFSet, and UPPer).
- ***RST Condition:** Defaults to –2 volts.

VOLTage:RANGe:LOWer?

[SENSe:]VOLTage<*number*>:RANGe:LOWer? returns the current lower value for the input number specified. The value (in ± volts) is sent to the output buffer.

Example **Querying input 2 current lower value**

`VOLT2:RANG:LOW 1`	*Lower limit to 1 volt*
`VOLT2:RANG:LOW?`	*Query instrument to return lower range limit value in volts*
`enter statement`	*Enter value into computer*

VOLTage:RANGe:OFFSet

[SENSe:]VOLTage*<number>*:RANGe:OFFSet *<value>* sets the voltage that is represented at the center of the current range for the selected input number.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number *value*	numeric numeric	1 to 4 Depends on VOLTage<n>:RANGe[:PTPeak]	none V

Example **Set input 2 offset to 10 V**

```
VOLT2:RANG:OFFS  10     Input 2 offset to 10 volts
```

Comments

- **Entering Offset:** The range of acceptable OFFSet values is dependent on the current VOLTage<*n*>:RANGe[:PTPeak] setting as follows:

VOLTage<n>:RANGe[:PTPeak]	VOLTage<n>:OFFSet limits
8 mV to 400 mV	±2 V
>400 mV to 2.0 V	±10 V
>2.0 V to 10.0 V	±50 V
>10.0 V to 40.0 V	±250 V

  If OFFSet is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.

- **Probe Attenuation:** Changing CORRection<n>:AFACtor settings after selecting VOLTage<n>:RANGe:OFFSet will cause the offset parameter to change.
- **Related Commands:** [SENSe:]VOLTage<n>:RANGe[:PTPeak], CORRection<n>:AFACtor.
- ***RST Condition:** Defaults to 0 volts.

VOLTage:RANGe:OFFSet?

[SENSe:]VOLTage*<number>*:RANGe:OFFSet? returns the current offset value for the input number specified. The value (in volts) is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example **Querying input 2 offset value**

```
VOLT2:RANG:OFFS  10     Input 2 offset to 10 volts
VOLT2:RANG:OFFS?        Query instrument to return input 2
                        offset value in volts
enter  statement        Enter value into computer
```

VOLTage:RANGe[:PTPeak]

[SENSe:]VOLTage*<number>*:RANGe[:PTPeak] *<range>* is used to define the full scale vertical axis, or "Y-axis" of the input specified.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none
range	numeric	8 MV to 40.0 V	V

Example **Set input 2 range to 10 volts (full scale)**

```
VOLT2:RANG:PTP  10      Input 2 range to 10 volts
```

Comments

- **PTPeak versus CORRection:** PTPeak values can be set from 0.008 to 40.0 when CORRection<n>:AFACtor is set to 1:1. If the CORRection<n>:AFACtor value is changed, the VOLTage<n>:RANGe[:PTPeak] value is multiplied by the probe attenuation factor.
- **Effects on Other VOLTage Selections:** PTPeak changes may cause the following commands to change their current parameters:

  VOLTage<n>:RANGe:LOWer: New lower voltage is calculated and entered as follows:

  LOWer = (PTPeak/2) below current OFFSet setting

  VOLTage<n>:RANGe:UPPer: New upper voltage is calculated and entered as follows:

  UPPer = (PTPeak/2) above current OFFSet setting
- **PTPeak versus TRIGger:** Changing the PTPeak value may effect the TRIGger:LEVel currently selected.
- **PTPeak versus OFFSet:** Changing the PTPeak value does NOT change the current OFFSet value.
- **Related Commands:** CORRection<n>:AFACtor, [SENSe:]VOLTage<n>:RANGe:(OFFSet, UPPer, and LOWer).
- ***RST Condition:** Defaults to 4 volts on all inputs.

VOLTage:RANGe[:PTPeak]?

[SENSe:]VOLTage*<number>*:RANGe[:PTPeak]? returns the current full scale vertical axis setting for the input specified. The value (in volts) is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Example **Querying input 2 full scale range setting**

```
VOLT2:RANG:PTP  10      Input 2 range to 10 volts
VOLT2:RANG:PTP?         Query instrument to return input 2
                        full scale range setting
enter  statement        Enter value into computer
```

VOLTage:RANGe:UPPer

[SENSe:]VOLTage*<number>*:RANGe:UPPer *<upper>* sets the voltage that is represented at the upper screen for the selected input number.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
upper	numeric	Dependent on VOLTage<n>:RANGe[:PTPeak]	volts

Example **Set input 3 upper range limit to 10 volts**

```
VOLT3:RANG:UPP  10      Upper limit to 10 volts
```

Comments

- **Entering Upper:** The range of acceptable UPPer values is dependent on the current VOLTage<*n*>:RANGe[:PTPeak] setting UPPer can be calculated using the values given below and the following formula:

  UPPer = OFFSet ± (PTPeak/2)

VOLTage<n>:RANGe[:PTPeak]	VOLTage<n>:OFFSet limits
8 mV to 400 mV	±2 V ± (PTPeak/2)
>400 mV to 2.0 V	±10 V ± (PTPeak/2)
>2.0 V to 10.0 V	±50 V ± (PTPeak/2)
>10.0 V to 40.0 V	±250 V ± (PTPeak/2)

  For example, if PTPeak is set to 400 mV, then

  Maximum UPPer = +2 V + (400 mV/2) = +2 V + 200 mV = +2.2 V

  If UPPer is set to a value outside the allowable range, it will automatically be set to the nearest acceptable value without generating an error.

- **Effects on Other VOLTage Selections:** Upper changes will cause the following commands to change their current parameters:

  VOLTage<n>:RANGe:LOWer: New lower value is calculated and entered as follows:

  lower = upper – PTPeak

  VOLTage<n>:RANGe:OFFSet: New OFFSet value is calculated and entered as follows:

  offset = center of upper and lower

- **Entering VOLTage<n>:RANGe[:PTPeak]:** PTPeak CANNOT be entered or changed using the VOLTage<n>:RANGe:UPPer/:LOWer commands.

- **Related Commands:** [SENSe:]VOLTage<n>:RANGe([:PTPeak], OFFSet, and LOWer).

- ***RST Condition:** Defaults to 2 volts.

VOLTage:RANGe:UPPer?

[SENSe:]VOLTage<*number*>:RANGe:UPPer? returns the current lower value for the input number specified. The value (in ± volts) is sent to the output buffer.

Example **Querying input 3 current upper value**

`VOLT3:RANG:UPP 10`	*Upper limit to 10 volts*
`VOLT3:RANG:UPP?`	*Query instrument to return upper range limit value in volts*
`enter statement`	*Enter value into computer*

STATus

The STATus command subsystem enables you to examine the status of the Oscilloscope trigger, calibration, and self test results by monitoring (reading the bit value) the various register groups. Figure 6-1 shows the four STATus Registers in the Oscilloscope.

Standard Event Status Register (*ESE). Operates under IEEE 488.2 control. Refer to the appropriate mainframe or command module users manual for more information on this register.

Status Byte Register (*STB?). Operates under IEEE 488.2 control. Refer to the appropriate mainframe or command module users manual for more information on this register.

Standard Operation Status Register. Operates under Oscilloscope control. The Operation Status Register (figure 6-1) is discussed in this section.

Questionable Data/Signal Register. Operates under Oscilloscope control. The Questionable Data/Signal Register (figure 6-2) is discussed in this section.

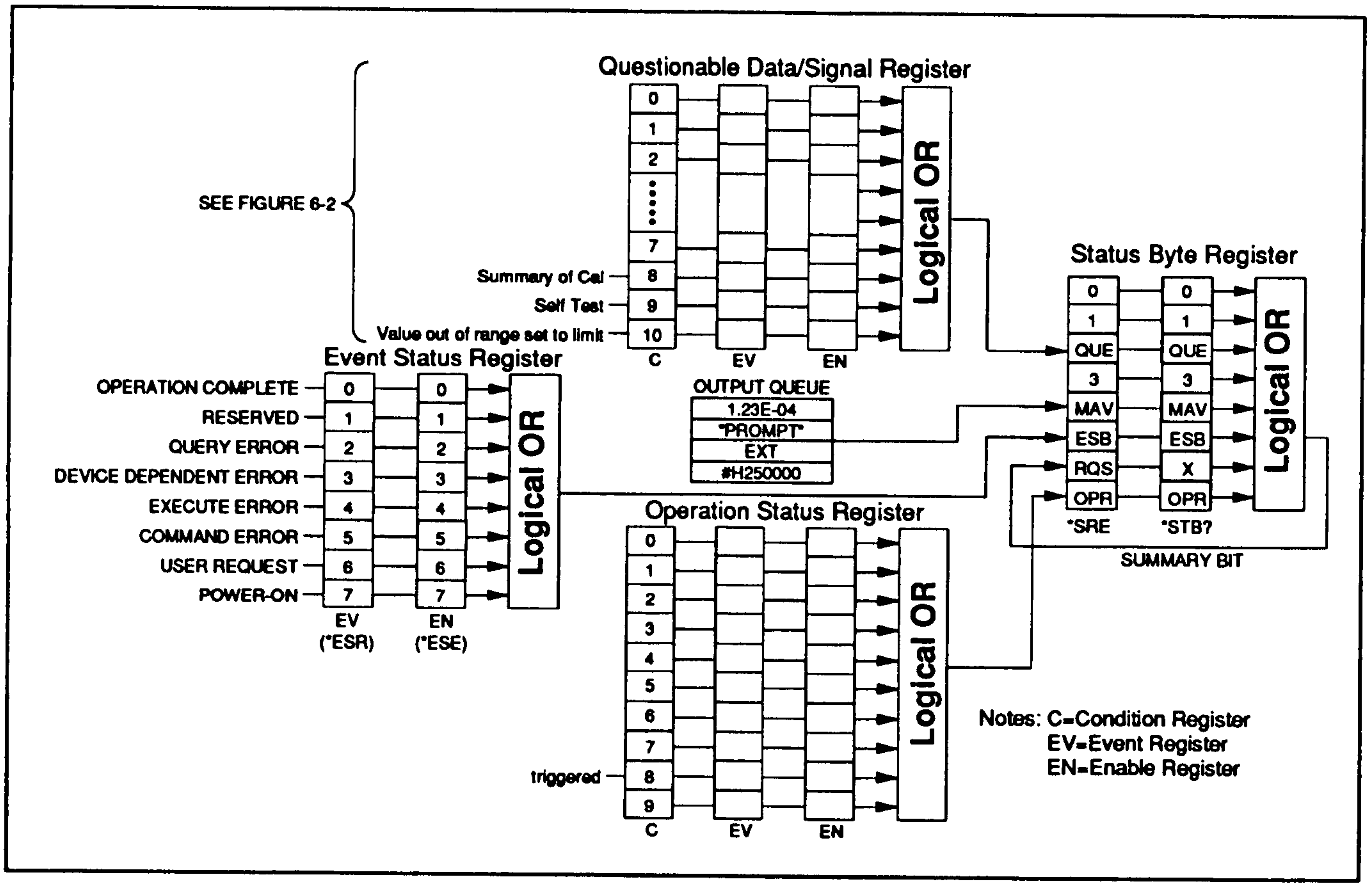


Figure 6-1. Oscilloscope STATus Registers

Subsystem Syntax

```
STATus
    :OPERation
            :CONDition?
            :ENABle
            [:EVENt]?
    :PRESet
    :QUEStionable
        :CONDition?
        :ENABle
        [:EVENt]?
        :CALibration
            :CONDition?
            :ENABle
            [:EVENt]?
            :INPut<number>
                :CONDition?
                :ENABle
                [:EVENt]?
                :AD
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :DELay
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :GAIN
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :HYSTeresis
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :LTRigger
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :OFFSet
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :TNULl
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :TRIGger
                    :CONDition?
                    :ENABle
                    [:EVENt]?
            :DCALibration
                :CONDition?
                :ENABle
                [:EVENt]?
```

Subsystem Syntax

```
STATus — Continued
    :QUEStionable — Continued
            :PROBe
                :CONDition?
                :ENABle
                [:EVENt]?
        :TEST
            :CONDition?
            :ENABle
            [:EVENt]?
            :ACQuisition
                :CONDition?
                :ENABle
                [:EVENt]?
                :AD
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :ATRigger
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :DA
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :LTRigger
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :TIMebase
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                    :INTerpolator
                        :CONDition?
                        :ENABle
                        [:EVENt]?
            :RAM
                :CONDition?
                :ENABle
                [:EVENt]?
                :ACQuisition
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :DISPlay
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :NVOLatile
                    :CONDition?
                    :ENABle
                    [:EVENt]?
```

Subsystem Syntax

```
STATus — Continued
  :QUEStionable — Continued
    :TEST — Continued
      :RAM — Continued
        :SYSTem
          :CONDition?
          :ENABle
          [:EVENt]?
      :ROM
        :CONDition?
        :ENABle
        [:EVENt]?
        :NPRotect
          :CONDition?
          :ENABle
          [:EVENt]?
        :SYSTem
          :CONDition?
          :ENABle
          [:EVENt]?
```

:OPERation:CONDition?

STATus:OPERation:CONDition? returns a decimal weighted value from 0 to 65535 indicating which bits are set true in the Operation Status Register's condition register. The contents of all the CONDition Registers are always set to "0".

Example **Read the condition register**

```
STAT:OPER:COND?                Queries the condition register,
                               without clearing the contents
```

:OPERation:ENABle

STATus:OPERation:ENABle ***<number>*** sets the enable mask, which allows true conditions (transitions) in the OPERation[:EVENt] register to be reported.

Example **Set enable register bit 8 to true**

```
STAT:OPER:ENAB  256            Sets bit 8 true
```

- **Bits Used:** Bit 8 (decimal 256) is the only bit used in the Operation Status Register for this instrument.

:OPERation:ENABle?

STATus:OPERation:ENABle? returns the bit value of the Operation Status Register's enable register. Returns a decimal weighted value from 0 to 65535 indicating which bits are set true. Reading the ENABle register does not clear its contents.

Example **Read the enable register**

```
STAT:OPER:ENAB?
```

Queries the enable register without clearing the contents

- **Bits Used:** Bit 8 (decimal 256) is the only bit used in the Operation Status Register for this instrument. Bit 8 is set true (1) when a trigger has occurred.

:OPERation[:EVENt]?

STATus:OPERation:[:EVENt]? queries the status of the Operation Status Register's event register. The event register latches only low to high events from the Operation Status Register's condition register. Returns a decimal weighted value from 0 to 65535 indicating which bits are set true. Reading the event register by a query will clear its contents.

Example **Read the event register**

```
STAT:OPER?
```

Queries the event register, and clears the contents

- **Bits Used:** Bit 8 (decimal 256) is the only bit used in the Operation Status Register for this instrument. Bit 8 is set true (1) when a trigger has occurred.

:PRESet

STATus:PRESet sets the contents of the Oscilloscope enable registers to a known state. When executed, the PRESet command affects all 51 QUEStionable ENABle registers, and sets all bits true (1).

Example **Preset the oscilloscope enable register**

```
STAT:PRES
```

All Enable register bits to true

Comments

- **Other Registers:** PRESet does not affect the Status Byte or Event Status registers.
- **Questionable Enable Register:** PRESet sets the questionable enable register to 0.
- **Event Registers:** PRESet does not clear any of the QUEStionable EVENt registers. Use the *CLS command is used to clear all event registers.

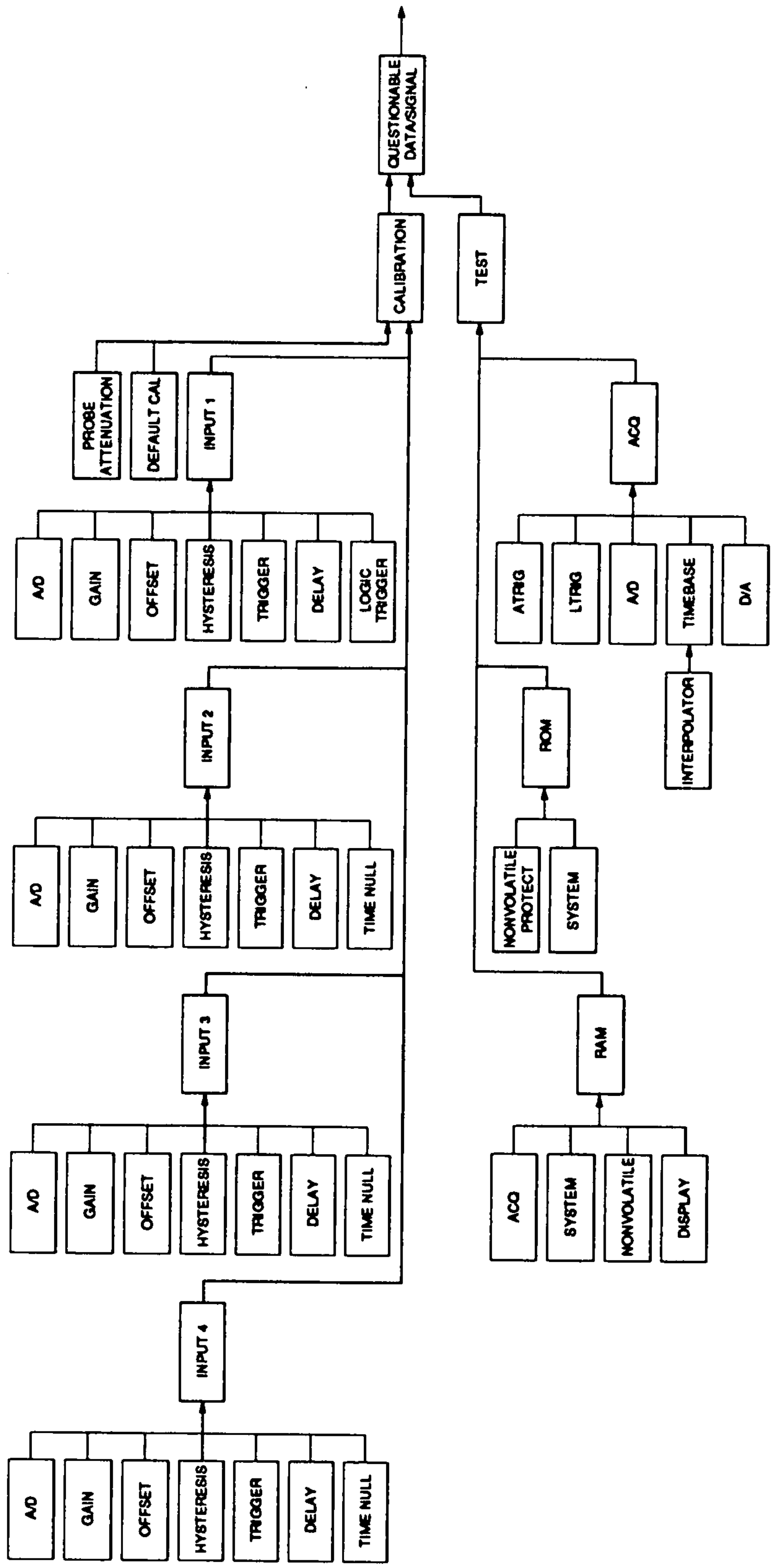
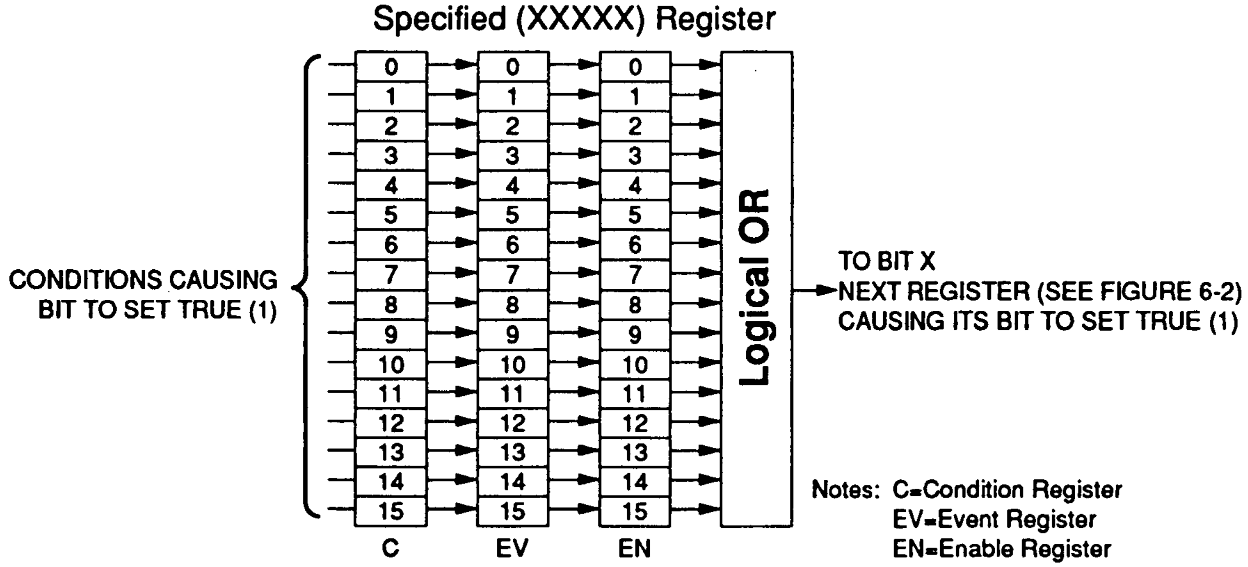


Figure 6-2. Oscilloscope Status Questionable Data/Signal Register Subsystem

:QUEStionable

STATus:QUEStionable subsystem contains 52 separate registers that, through summing registers, eventually report to the QUEStionable Data/Signal register. See figure 6-2 and the Subsystem Syntax at the beginning of this section for a list of all the registers that set the QUEStionable Data/Signal Register.

A diagram is provided for each register in the QUEStionable Data/Signal Register system as shown in figure 6-3. The following description for using the CONDition?, [:EVENt]?, and ENABle commands/queries apply to all registers within the Oscilloscope.


Specified (XXXXX) Register
CONDITIONS CAUSING BIT TO SET TRUE (1)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
C
EV
EN
Logical OR
TO BIT X
NEXT REGISTER (SEE FIGURE 6-2)
CAUSING ITS BIT TO SET TRUE (1)
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

These registers are set and queried using decimal weighted bit values. The decimal equivalent for bits 0 to 15 is shown below. As an example, sending a decimal value of 4608 will set bits 9 and 12 true (1).

Bit Number to Decimal Value

Bit Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Decimal Value	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	reserved

Figure 6-3. Register Diagram

Each individual register (XXXXX) in the Oscilloscope is made up of three separate registers:

:CONDition register

:EVENt register

:ENABle register

:CONDition?

STATus:QUEStionable:XXXXX:CONDition? queries the current contents of the specified (XXXXX):CONDition Register. The contents of all the CONDition Registers are always set to "0".

Example **Read the condition register**

```
STAT:QUES:XXXXX:COND?
```

Queries the specified (XXXXX) Condition Register.

:ENABle

STATus:QUEStionable:XXXXX:ENABle ***<number>*** sets the enable mask, which allows true conditions (transitions) in the specified (XXXXX):EVENt Register to be reported.

Example **Set enable register bits 9 through 12 to true**

```
STAT:QUES:XXXXX:ENAB  7680
```

Sets bits 9 to 12 true

:ENABle?

STATus:QUEStionable:XXXXX:ENABle? returns the bit value of the specified (XXXXX):ENABle Register. Returns a decimal weighted value from 0 to 65535 indicating which bits are set true. Reading the ENABle register does not clear its contents.

Example **Query the enable register**

```
STAT:QUES:XXXXX:ENAB?
```

Queries the specified (XXXXX) Enable register, without clearing the contents

[:EVENt]?

STATus:QUEStionable:XXXXX[:EVENt]? Queries the status of the specified (XXXXX):EVENt Register. The EVENt Register latches only low to high events from the specified (XXXXX):CONDition Register. Returns a decimal weighted value from 0 to 65535 indicating which bits are set true. Reading the specified EVENt Register by a query will clear its contents.

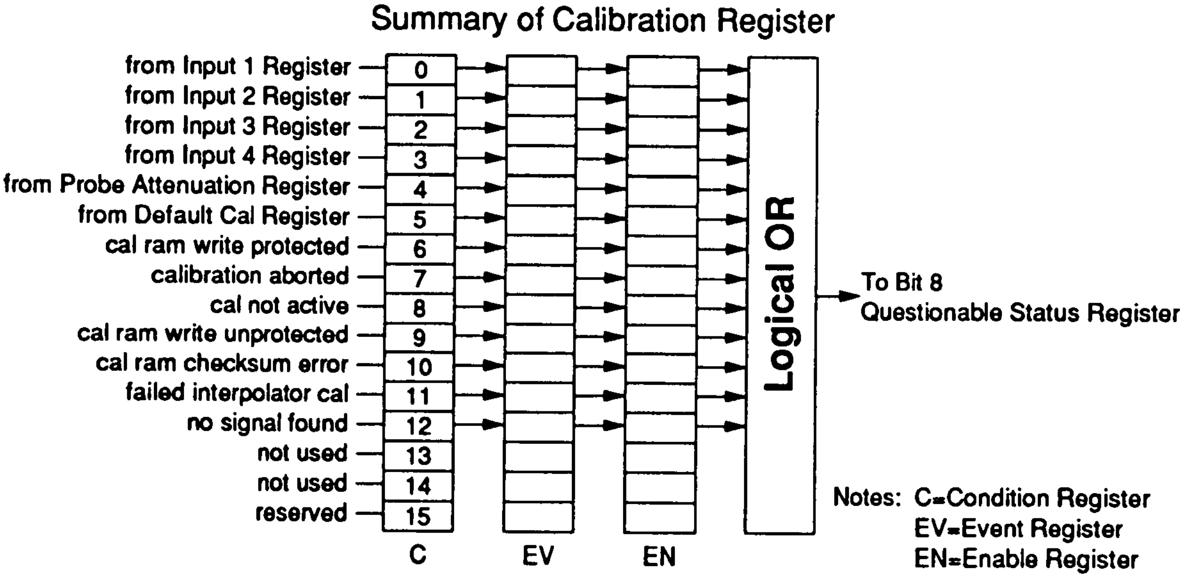
Example **Read the event register**

```
STAT:QUES:XXXXX?
```

Queries the specified (XXXXX) Event Register and clears the contents

:QUEStionable:CALibration

STATus:QUEStionable:CALibration register reports a summary of calibration results and status for all inputs to the Questionable Data/Signal Register. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.



Example **Query calibration event register**

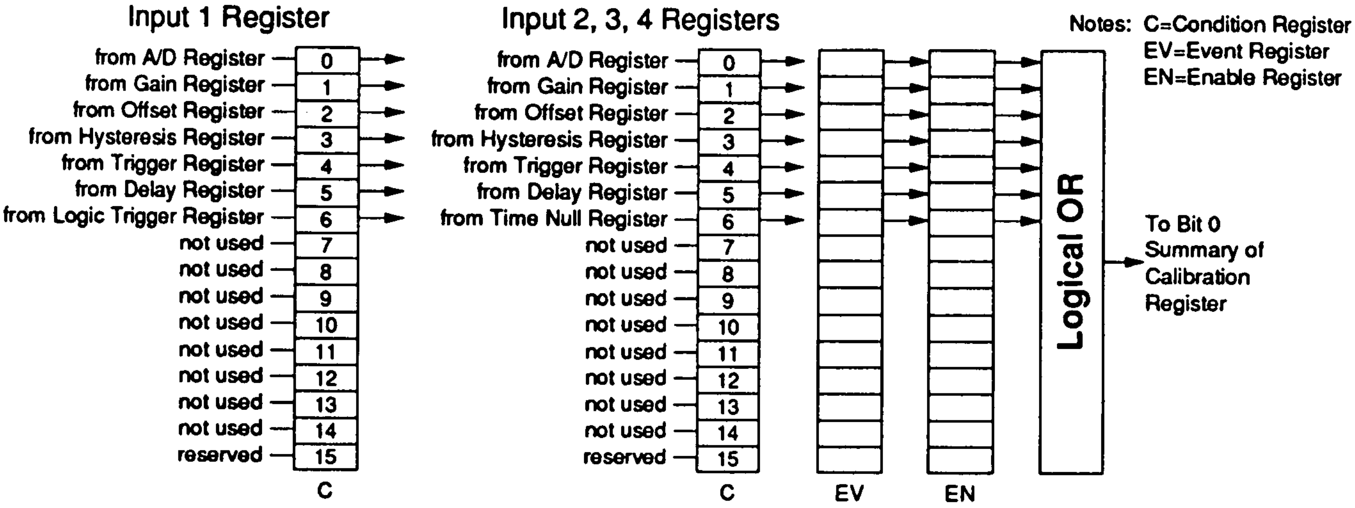
`STAT:QUES:CAL?` *Query instrument to return register contents*

:QUEStionable:CALibration:INPut

STATus:QUEStionable:CALibration:INPut<*number*> register reports the status of calibration data for the input specified. *number* (1 to 4) specified the desired input. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none



Example **Query input 1 event register**

`STAT:QUES:CAL:INP1?` *Query instrument to return register contents*

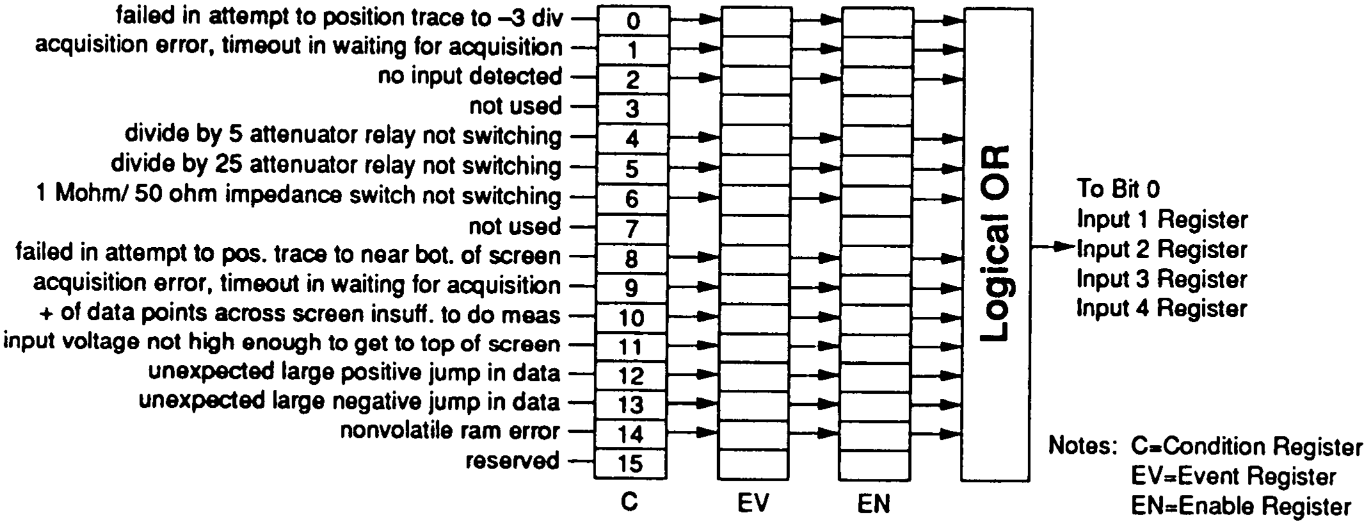
:QUEStionable:CALibration:INPut:AD

STATus:QUEStionable:CALibration:INPut<*number*>:AD register reports the status of the A/D calibration data for the input specified. *number* (1 to 4) specifies the desired input. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

A/D Register


failed in attempt to position trace to –3 div
acquisition error, timeout in waiting for acquisition
no input detected
not used
divide by 5 attenuator relay not switching
divide by 25 attenuator relay not switching
1 Mohm/ 50 ohm impedance switch not switching
not used
failed in attempt to pos. trace to near bot. of screen
acquisition error, timeout in waiting for acquisition
+ of data points across screen insuff. to do meas
input voltage not high enough to get to top of screen
unexpected large positive jump in data
unexpected large negative jump in data
nonvolatile ram error
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 0
Input 1 Register
Input 2 Register
Input 3 Register
Input 4 Register

Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query input 2 A/D event register**

```
STAT:QUES:CAL:INP2:AD?
```

Query instrument to return register contents

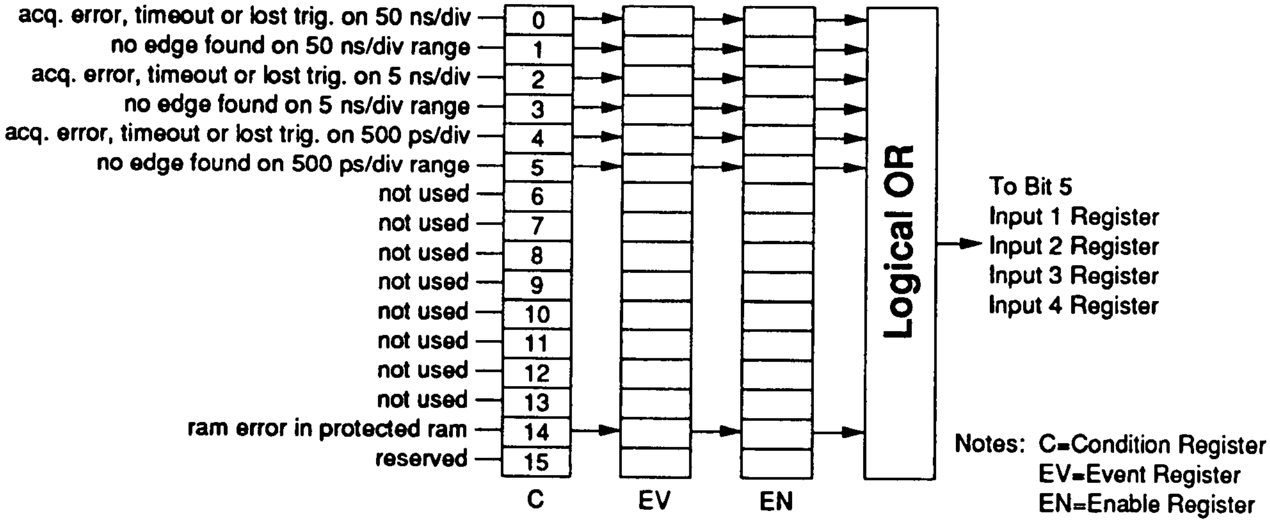
:QUEStionable:CALibration:INPut:DELay

STATus:QUEStionable:CALibration:INPut<*number*>:DELay register reports the status of delay calibration data for the input specified. *number* (1 to 4) specifies the desired input. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none

Delay Register


acq. error, timeout or lost trig. on 50 ns/div
no edge found on 50 ns/div range
acq. error, timeout or lost trig. on 5 ns/div
no edge found on 5 ns/div range
acq. error, timeout or lost trig. on 500 ps/div
no edge found on 500 ps/div range
not used
not used
not used
not used
not used
not used
not used
not used
ram error in protected ram
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 5
Input 1 Register
Input 2 Register
Input 3 Register
Input 4 Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query input 3 delay event register**

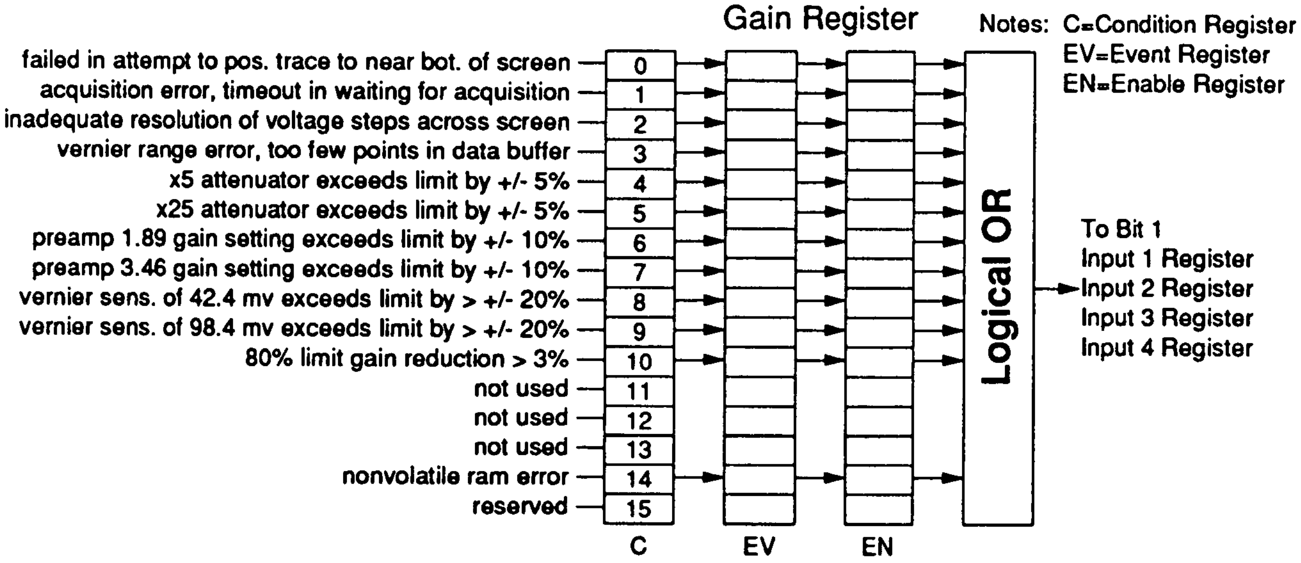
`STAT:QUES:CAL:INP3:DEL?` *Query instrument to return register contents*

:QUEStionable:CALibration:INPut:GAIN

STATus:QUEStionable:CALibration:INPut*<number>*:GAIN register reports the status of gain calibration data for the input specified. *number* (1 to 4) specifies the desired input. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none


Gain Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register
failed in attempt to pos. trace to near bot. of screen
0
acquisition error, timeout in waiting for acquisition
1
inadequate resolution of voltage steps across screen
2
vernier range error, too few points in data buffer
3
x5 attenuator exceeds limit by +/- 5%
4
x25 attenuator exceeds limit by +/- 5%
5
preamp 1.89 gain setting exceeds limit by +/- 10%
6
preamp 3.46 gain setting exceeds limit by +/- 10%
7
vernier sens. of 42.4 mv exceeds limit by > +/- 20%
8
vernier sens. of 98.4 mv exceeds limit by > +/- 20%
9
80% limit gain reduction > 3%
10
not used
11
not used
12
not used
13
nonvolatile ram error
14
reserved
15
C
EV
EN
Logical OR
To Bit 1
Input 1 Register
Input 2 Register
Input 3 Register
Input 4 Register

Example **Query input 4 gain event register**

```
STAT:QUES:CAL:INP4:GAIN?
```

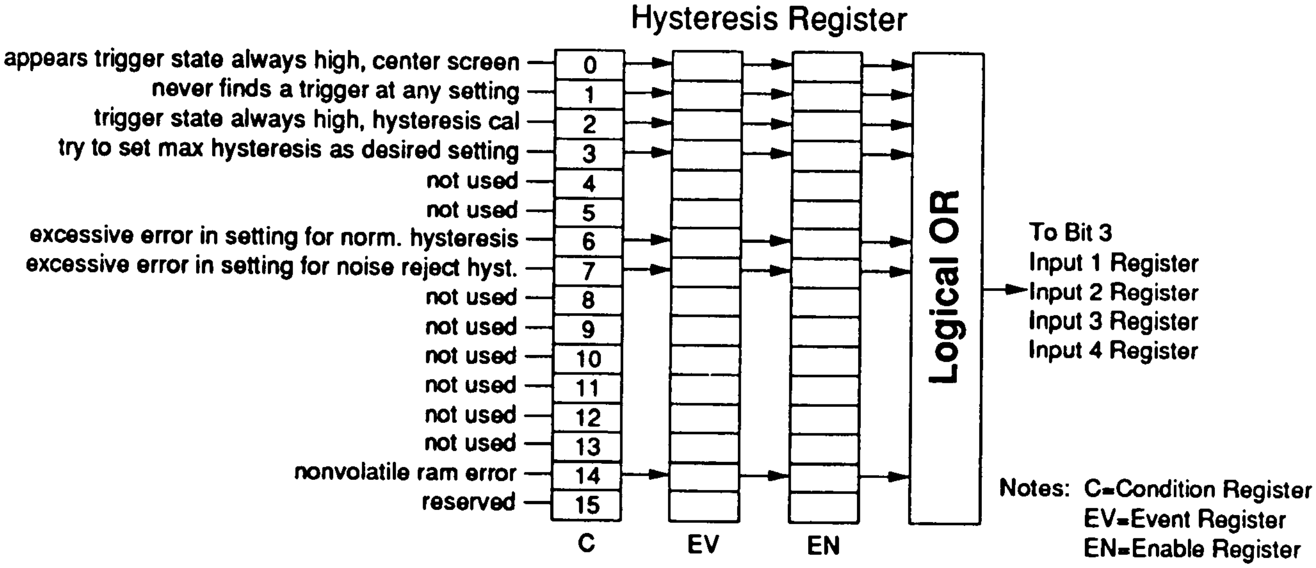
Query instrument to return register contents

:QUEStionable:CALibration:INPut:HYSTeresis

STATus:QUEStionable:CALibration:INPut<*number*>:HYSTeresis register reports the status of hysteresis calibration data for the input specified. *number* (1 to 4) specifies the desired input. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none


Hysteresis Register
appears trigger state always high, center screen
never finds a trigger at any setting
trigger state always high, hysteresis cal
try to set max hysteresis as desired setting
not used
not used
excessive error in setting for norm. hysteresis
excessive error in setting for noise reject hyst.
not used
not used
not used
not used
not used
not used
nonvolatile ram error
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 3
Input 1 Register
Input 2 Register
Input 3 Register
Input 4 Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example

Query input 1 hysteresis event register

```
STAT:QUES:CAL:INP1:HYST?
```

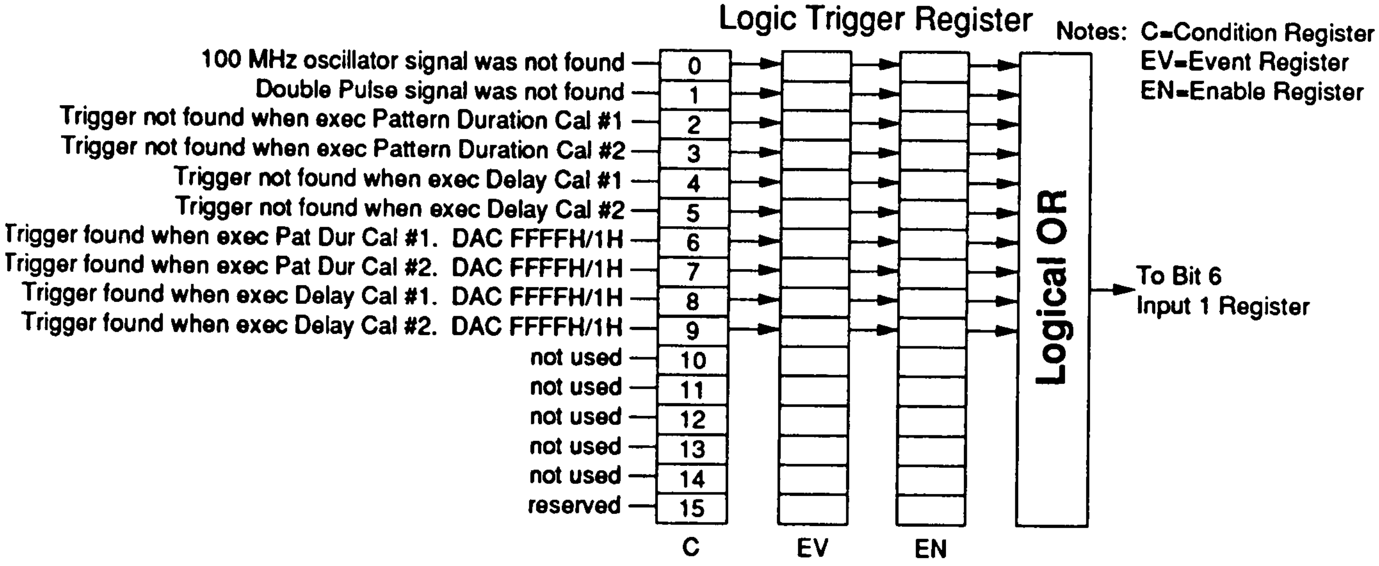
Query instrument to return register contents

:QUEStionable:CALibration:INPut:LTRigger

STATus:QUEStionable:CALibration:INPut*<number>*:LTRigger register reports the status of logic trigger calibration data for input 1. Only input 1 contains the LTRigger register. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1	none


Logic Trigger Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register
100 MHz oscillator signal was not found
Double Pulse signal was not found
Trigger not found when exec Pattern Duration Cal #1
Trigger not found when exec Pattern Duration Cal #2
Trigger not found when exec Delay Cal #1
Trigger not found when exec Delay Cal #2
Trigger found when exec Pat Dur Cal #1. DAC FFFFH/1H
Trigger found when exec Pat Dur Cal #2. DAC FFFFH/1H
Trigger found when exec Delay Cal #1. DAC FFFFH/1H
Trigger found when exec Delay Cal #2. DAC FFFFH/1H
not used
not used
not used
not used
not used
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 6
Input 1 Register

Example

Query input 1 logic trigger event register

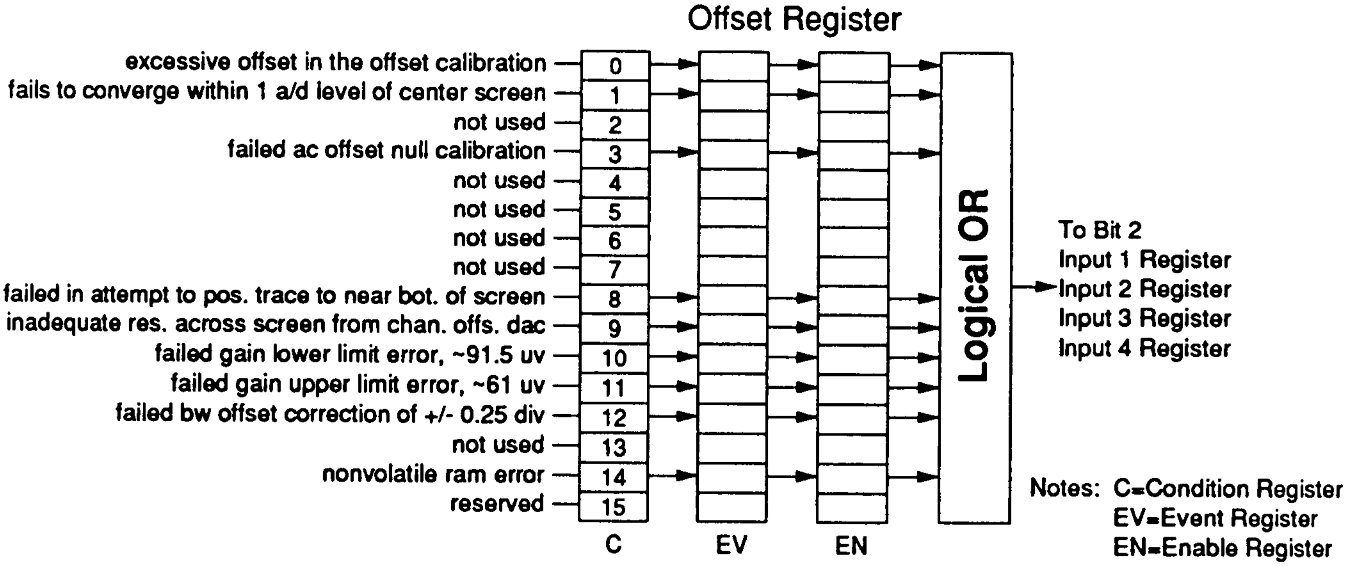
`STAT:QUES:CAL:INP1:LTR?` *Query instrument to return register contents*

:QUEStionable:CALibration:INPut:OFFSet

STATus:QUEStionable:CALibration:INPut<*number*>:OFFSet register reports the status of offset calibration data for the input specified. *number* (1 to 4) specifies the desired input. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none


Offset Register
excessive offset in the offset calibration — 0
fails to converge within 1 a/d level of center screen — 1
not used — 2
failed ac offset null calibration — 3
not used — 4
not used — 5
not used — 6
not used — 7
failed in attempt to pos. trace to near bot. of screen — 8
inadequate res. across screen from chan. offs. dac — 9
failed gain lower limit error, ~91.5 uv — 10
failed gain upper limit error, ~61 uv — 11
failed bw offset correction of +/- 0.25 div — 12
not used — 13
nonvolatile ram error — 14
reserved — 15
C
EV
EN
Logical OR
To Bit 2
Input 1 Register
Input 2 Register
Input 3 Register
Input 4 Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query input 2 offset event register**

```
STAT:QUES:CAL:INP2:OFFS?
```

Query instrument to return register contents

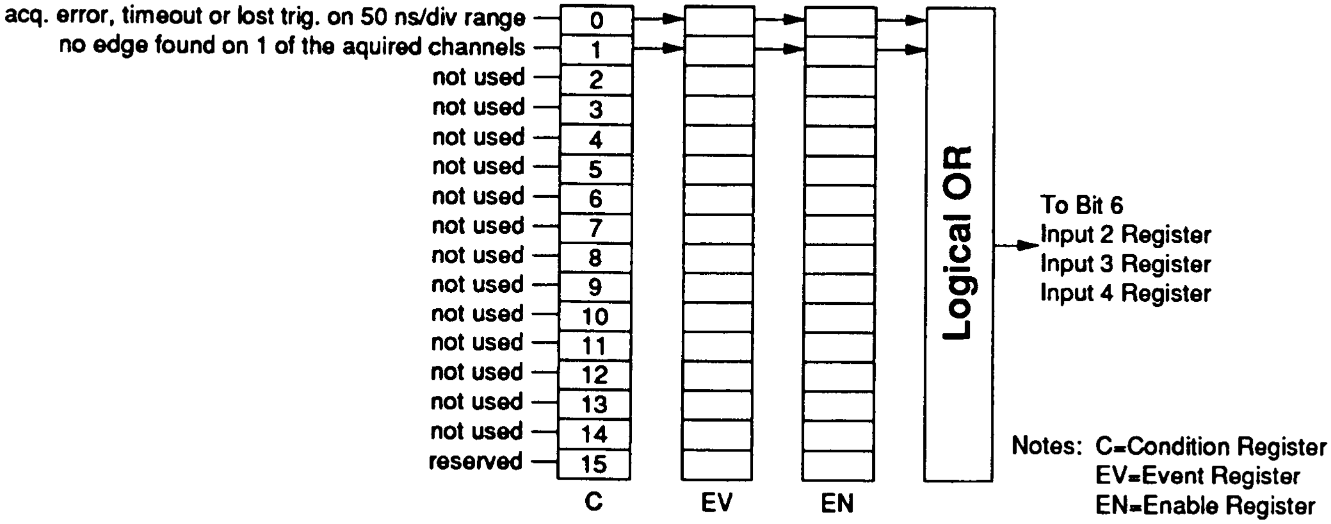
:QUEStionable:CALibration:INPut:TNULl

STATus:QUEStionable:CALibration:INPut*<number>*TNULl register reports the status of time null calibration data for the input specified. *number* (2 to 4) specifies the desired input. Input 1 does not contain a time null register. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	2 to 4	none

Time Null Register



Example **Query input 3 time null event register**

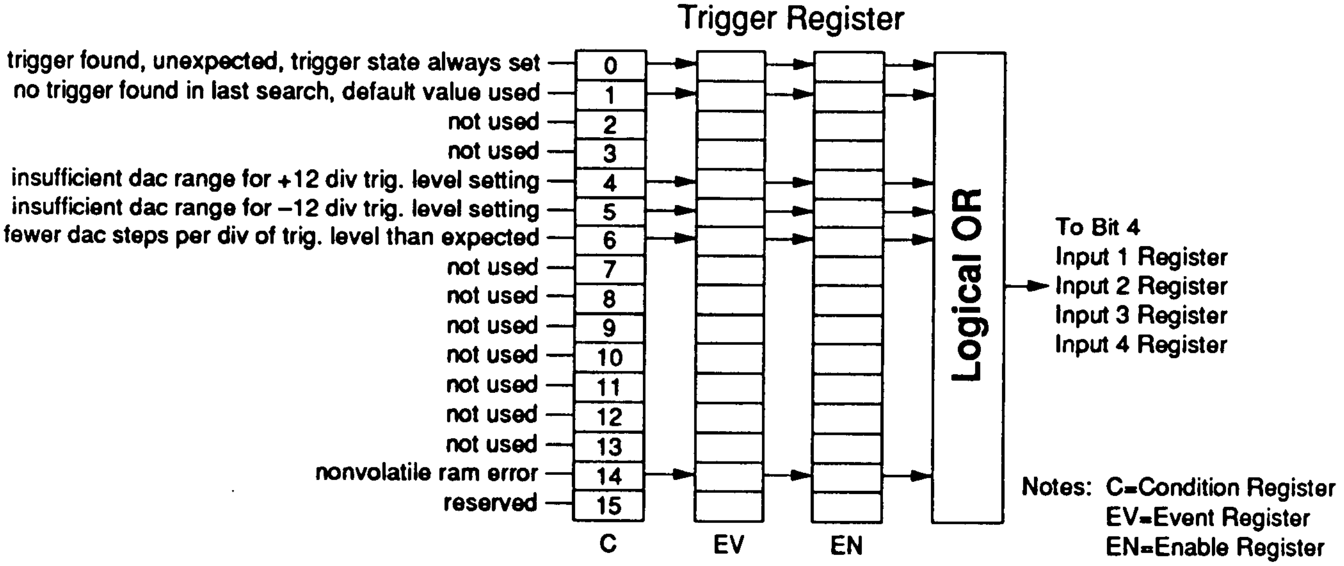
```
STAT:QUES:CAL:INP3:TNUL?
```
Query instrument to return register contents

:QUEStionable:CALibration:INPut:TRIGger

STATus:QUEStionable:CALibration:INPut<*number*>:TRIGger register reports the status of trigger calibration data for the input specified. *number* (1 to 4) specifies the desired input. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
number	numeric	1 to 4	none


Trigger Register
trigger found, unexpected, trigger state always set
no trigger found in last search, default value used
not used
not used
insufficient dac range for +12 div trig. level setting
insufficient dac range for −12 div trig. level setting
fewer dac steps per div of trig. level than expected
not used
not used
not used
not used
not used
not used
not used
nonvolatile ram error
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 4
Input 1 Register
Input 2 Register
Input 3 Register
Input 4 Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

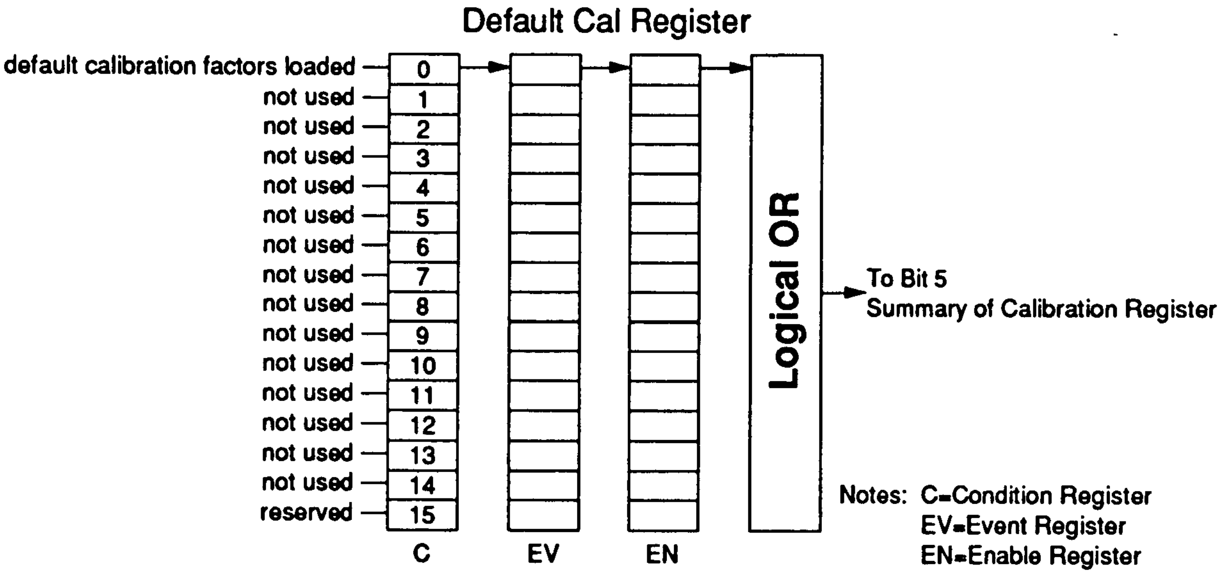
Example **Query input 4 trigger event register**

```
STAT:QUES:CAL:INP4:TRIG?
```

Query instrument to return register contents

:QUEStionable:CALibration:DCALibration

STATus:QUEStionable:CALibration:DCALibration register reports default calibration factor status load. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.



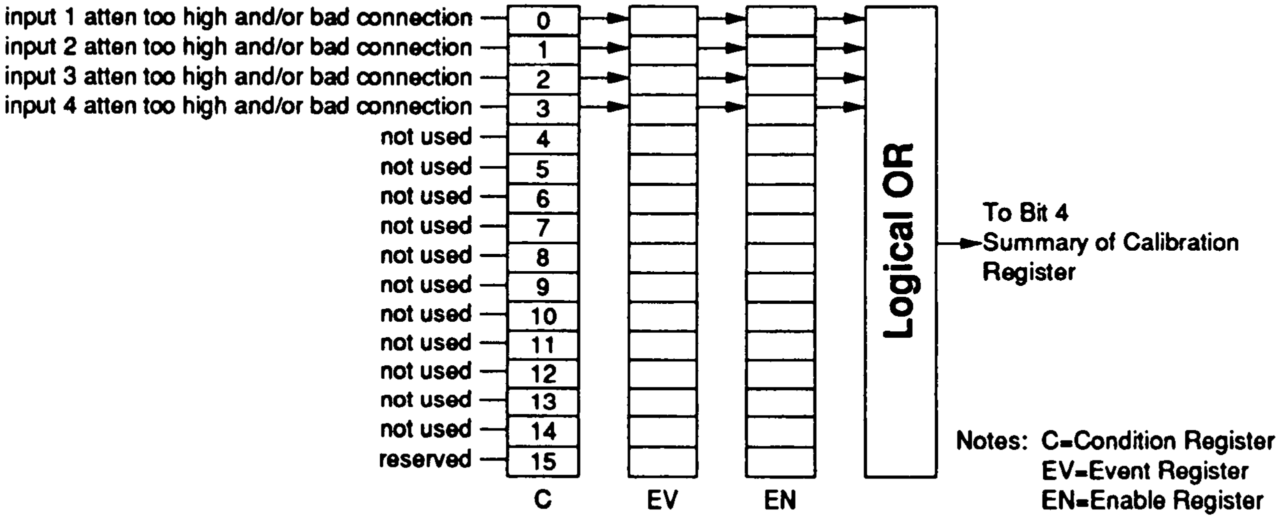
Example **Query default calibration event register**

`STAT:QUES:CAL:DCAL?`	*Query instrument to return register contents*

:QUEStionable:CALibration:PROBe

STATus:QUEStionable:CALibration:PROBe register reports probe calibration attenuation results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Probe Attenuation Register



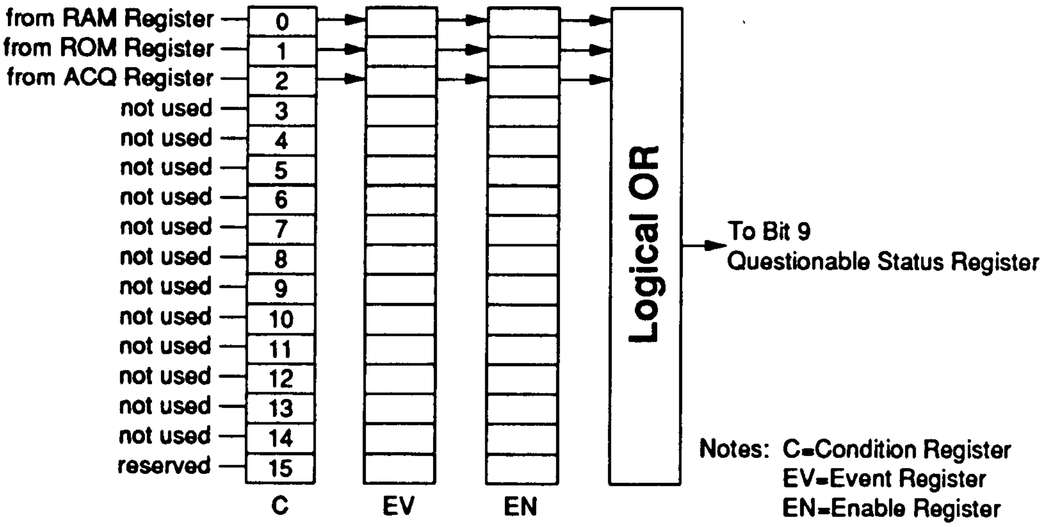
Example **Query probe calibration attenuation event register**

`STAT:QUES:CAL:PROB?` *Query instrument to return register contents*

:QUEStionable:TEST

STATus:QUEStionable:TEST register reports diagnostic test results or self test status. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Self Test Status Register



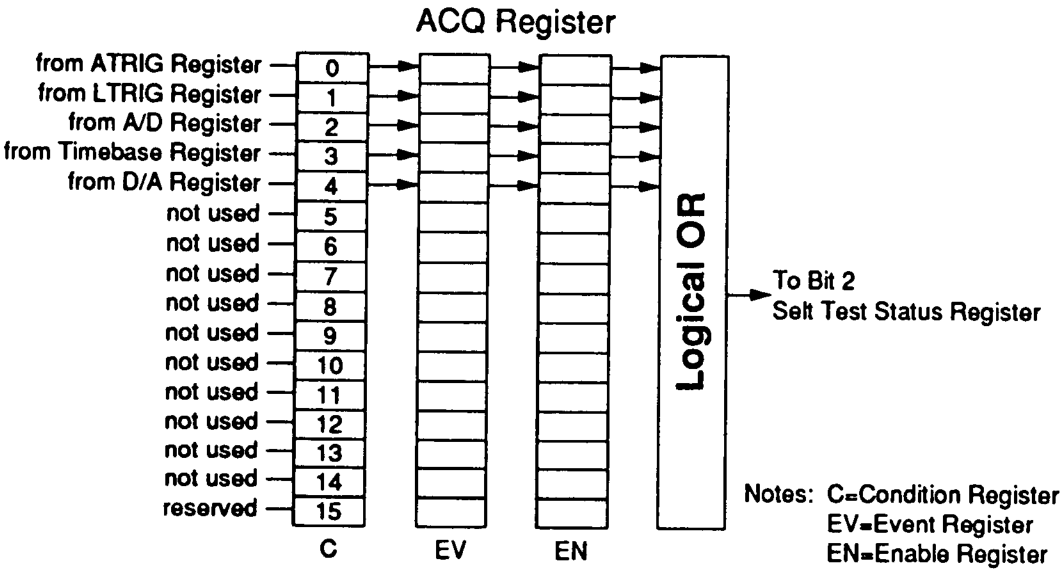
Example **Query test event register**

```
STAT:QUES:TEST?
```

Query instrument to return register contents

:QUEStlonable:TEST:ACQulsltlon

STATus:QUEStionable:TEST:ACQuisition register reports acquisition diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

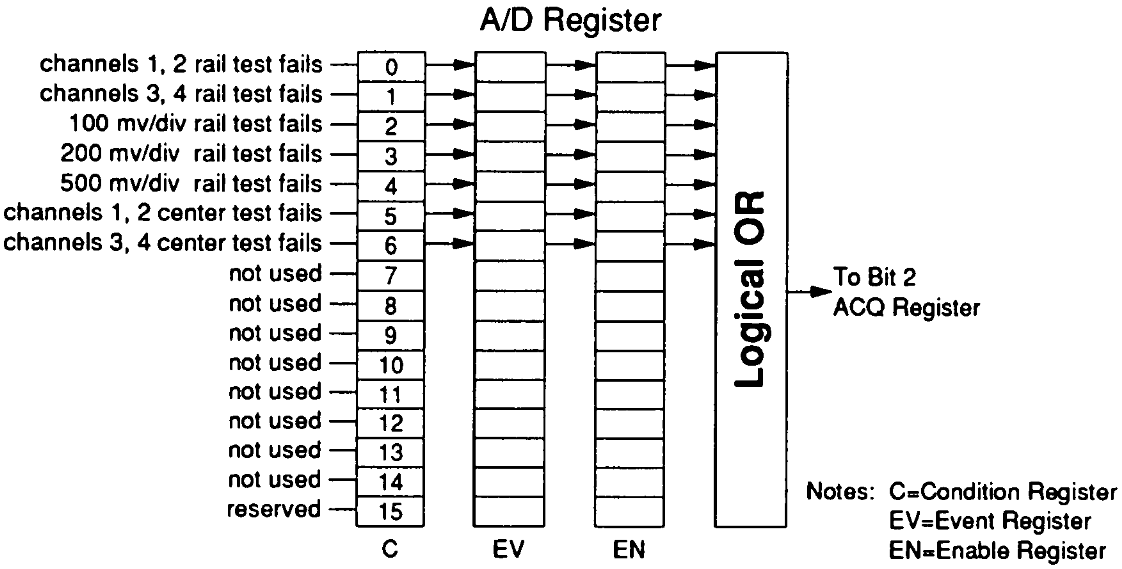


Example **Query acquisition event register**

`STAT:QUES:TEST:ACQ?` *Query instrument to return register contents*

:QUEStionable:TEST:ACQuisition:AD

STATus:QUEStionable:TEST:ACQuisition:AD register reports acquisition A/D diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.


A/D Register
channels 1, 2 rail test fails
channels 3, 4 rail test fails
100 mv/div rail test fails
200 mv/div rail test fails
500 mv/div rail test fails
channels 1, 2 center test fails
channels 3, 4 center test fails
not used
not used
not used
not used
not used
not used
not used
not used
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 2
ACQ Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

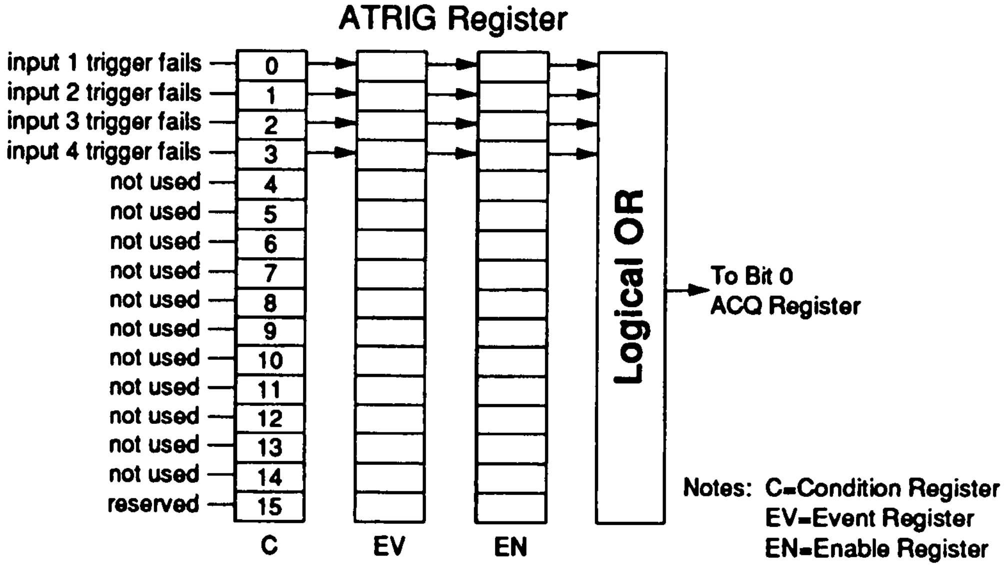
Example **Query acquisition A/D event register**

```
STAT:QUES:TEST:ACQ:AD?
```

Query instrument to return register contents

:QUEStionable:TEST:ACQuisition:ATRigger

STATus:QUEStionable:TEST:ACQuisition:ATRigger register reports acquisition analog trigger diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.



Example **Query acquisition analog trigger event register**

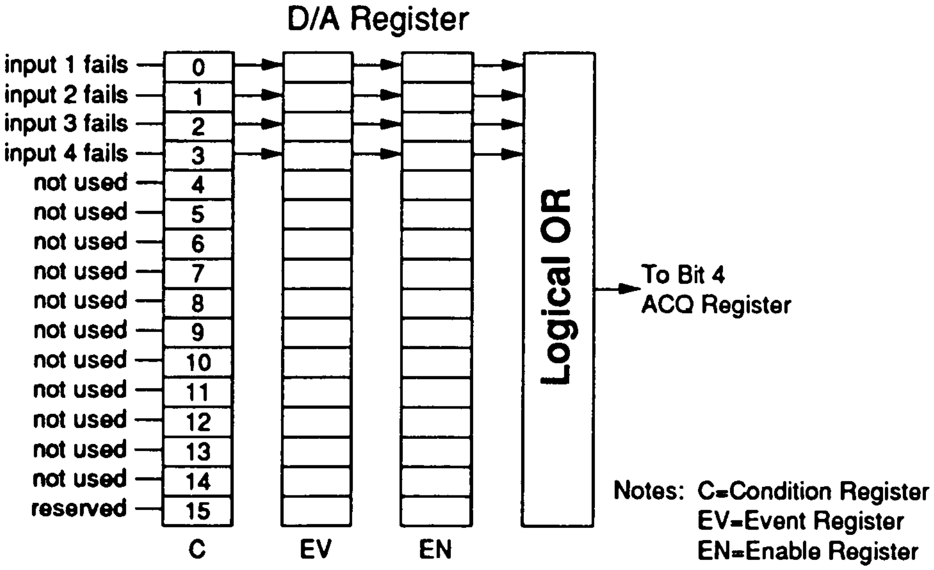
```
STAT:QUES:TEST:ACQ:ATR?
```

Query instrument to return register contents

:QUEStionable:TEST:ACQuisition:DA

STATus:QUEStionable:TEST:ACQuisition:DA register reports acquisition D/A diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

D/A Register



Notes: C=Condition Register
EV=Event Register
EN=Enable Register

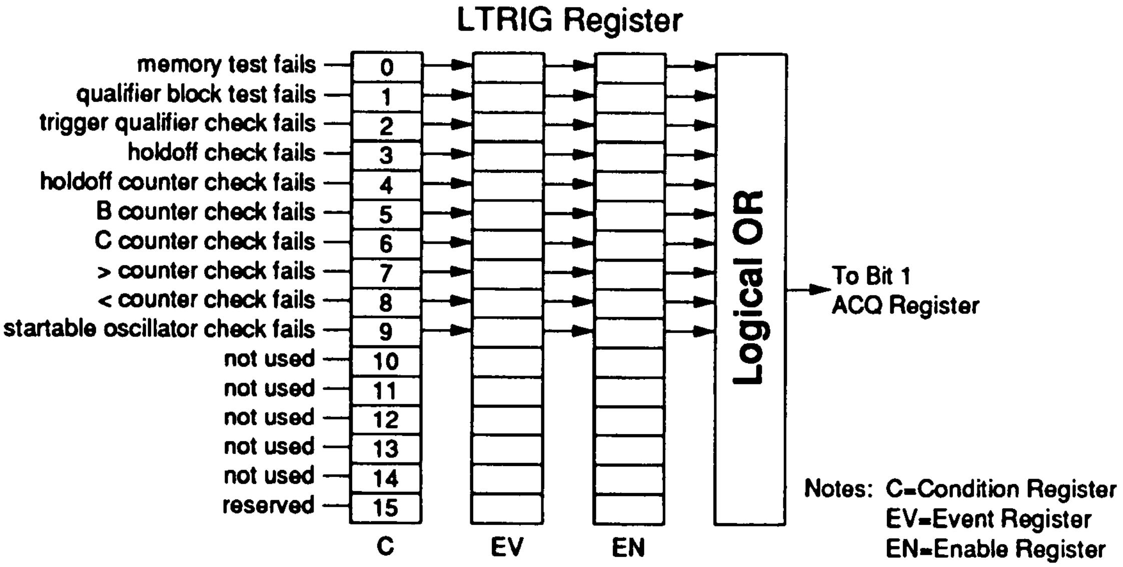
Example **Query acquisition D/A event register**

```
STAT:QUES:TEST:ACQ:DA?
```

Query instrument to return register contents

:QUEStionable:TEST:ACQuisition:LTRigger

STATus:QUEStionable:TEST:ACQuisition:LTRigger register reports acquisition logic trigger diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

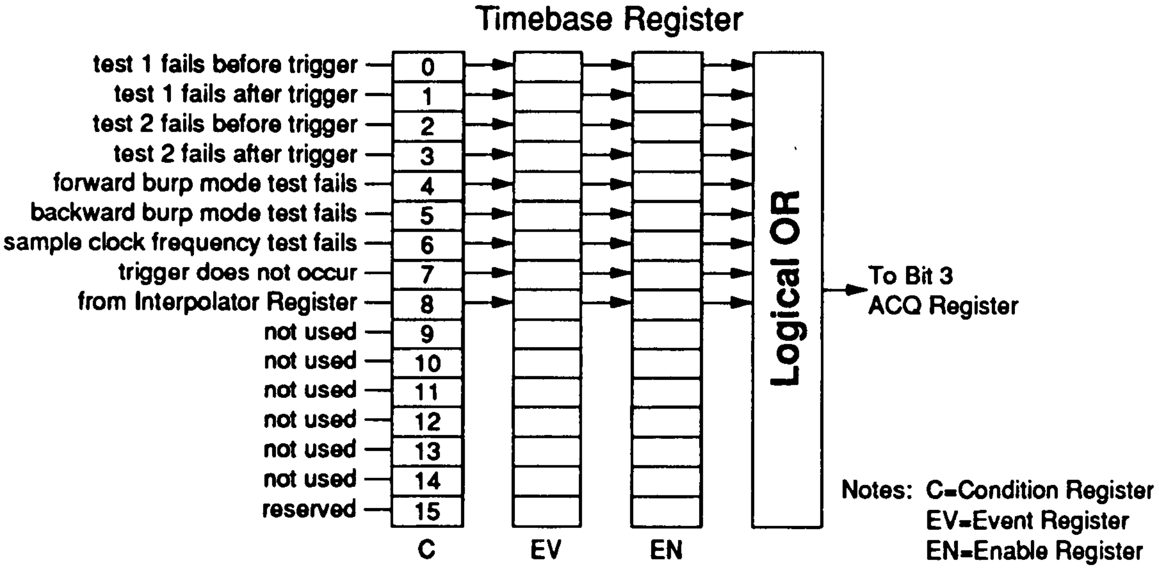


Example **Query acquisition logic trigger event register**

`STAT:QUES:TEST:ACQ:LTR?` *Query instrument to return register contents*

:QUEStionable:TEST:ACQuisition:TIMebase

STATus:QUEStionable:TEST:ACQuisition:TIMebase register reports acquisition time base diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.


Timebase Register
test 1 fails before trigger
test 1 fails after trigger
test 2 fails before trigger
test 2 fails after trigger
forward burp mode test fails
backward burp mode test fails
sample clock frequency test fails
trigger does not occur
from Interpolator Register
not used
not used
not used
not used
not used
not used
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 3
ACQ Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

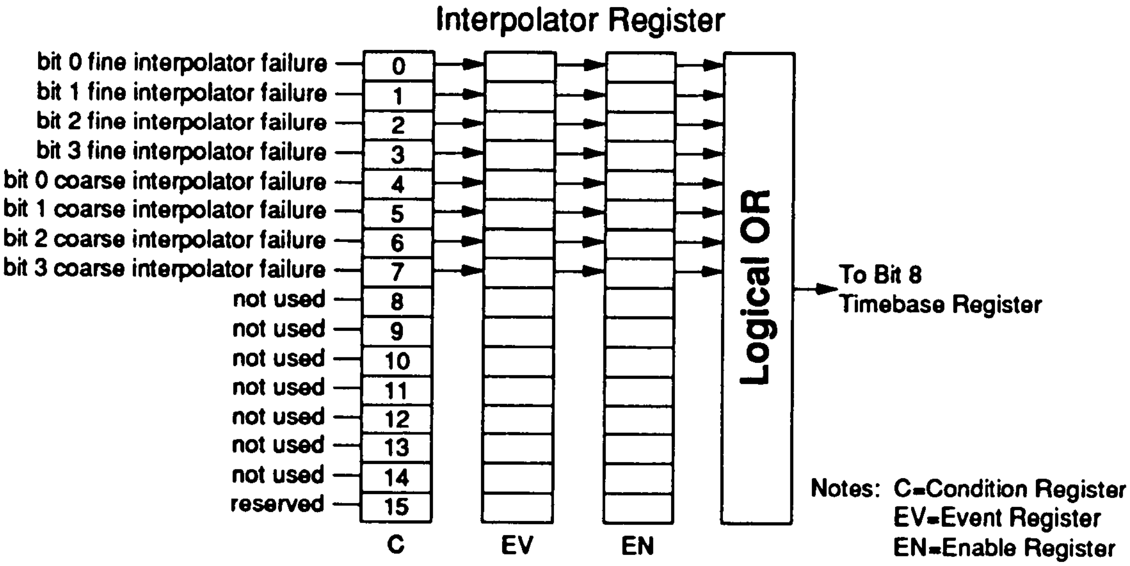
Example **Query acquisition time base event register**

```
STAT:QUES:TEST:ACQ:TIM?
```

Query instrument to return register contents

:QUEStionable:TEST:ACQuisition:TIMebase:INTerpolator

STATus:QUEStionable:TEST:ACQuisition:TIMebase:INTerpolator register reports acquisition time base interpolator diagnostics. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.


Interpolator Register
bit 0 fine interpolator failure
bit 1 fine interpolator failure
bit 2 fine interpolator failure
bit 3 fine interpolator failure
bit 0 coarse interpolator failure
bit 1 coarse interpolator failure
bit 2 coarse interpolator failure
bit 3 coarse interpolator failure
not used
not used
not used
not used
not used
not used
not used
reserved
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
C
EV
EN
Logical OR
To Bit 8
Timebase Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

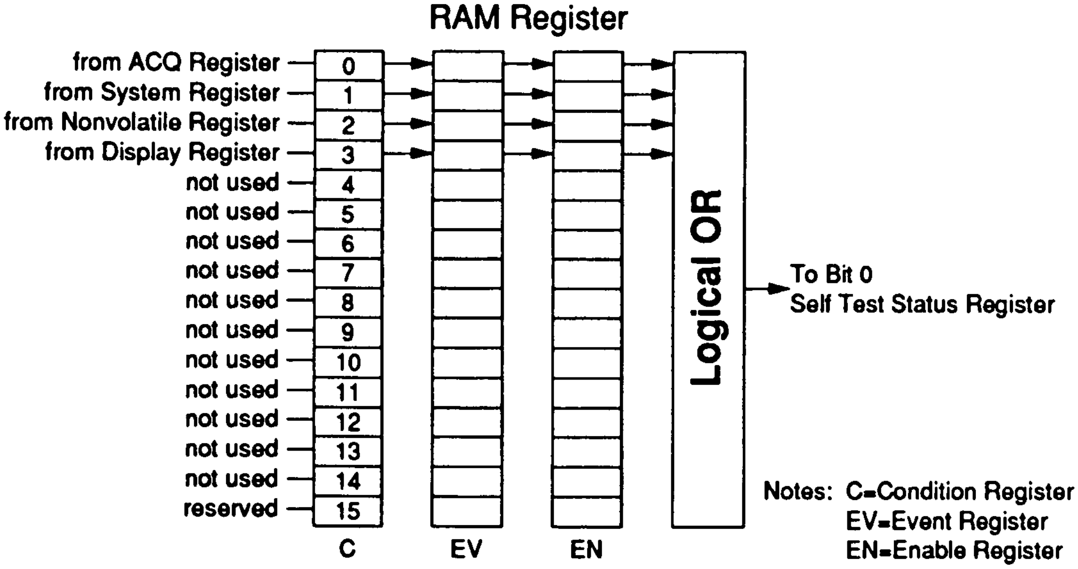
Example **Query acquisition time base interpolator event register**

```
STAT:QUES:TEST:ACQ:TIM:INT?
```

Query instrument to return register contents

:QUEStionable:TEST:RAM

STATus:QUEStionable:TEST:RAM register reports random access memory diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.



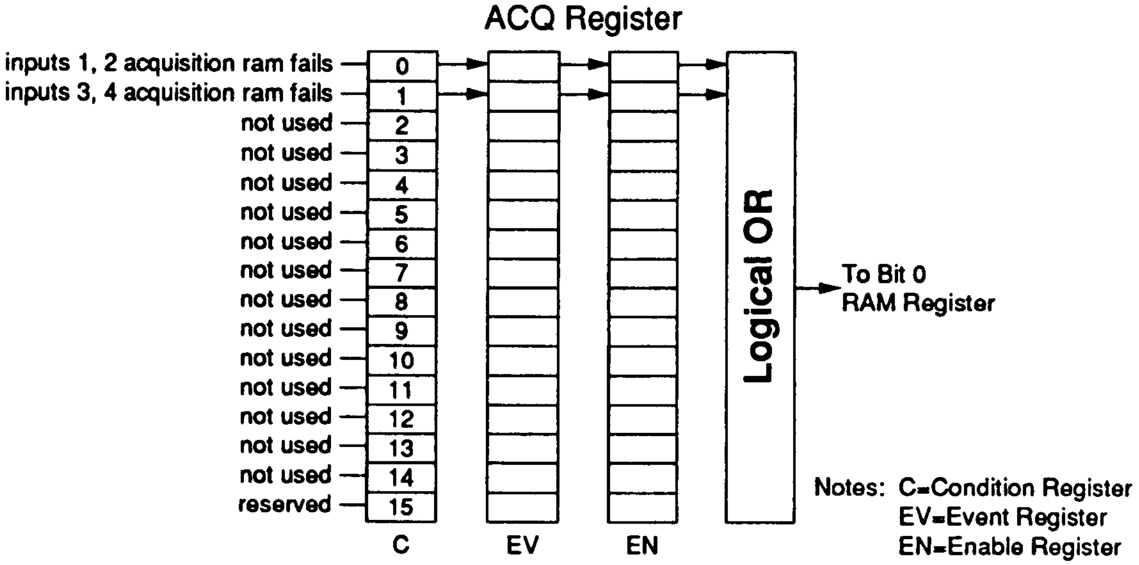
Example **Query RAM event register**

```
STAT:QUES:TEST:RAM?
```

Query instrument to return register contents

:QUEStionable:TEST:RAM:ACQuisition

STATus:QUEStionable:TEST:RAM:ACQuisition register reports acquisition random access memory diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

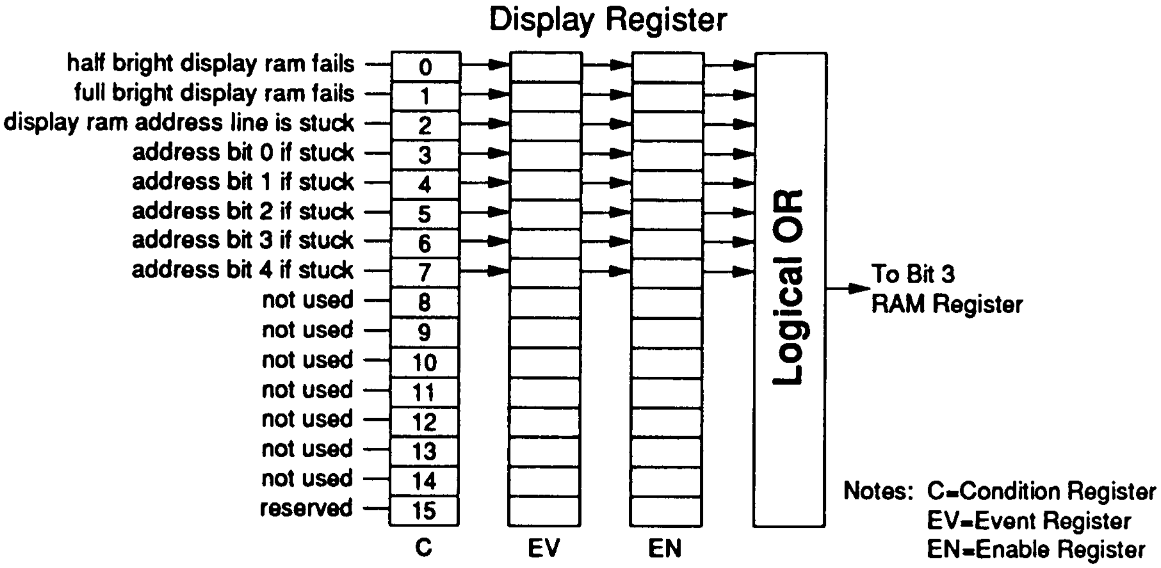


Example **Query acquisition RAM event register**

`STAT:QUES:TEST:RAM:ACQ?` *Query instrument to return register contents*

:QUEStionable:TEST:RAM:DISPlay

STATus:QUEStionable:TEST:RAM:DISPlay register reports display random access memory test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

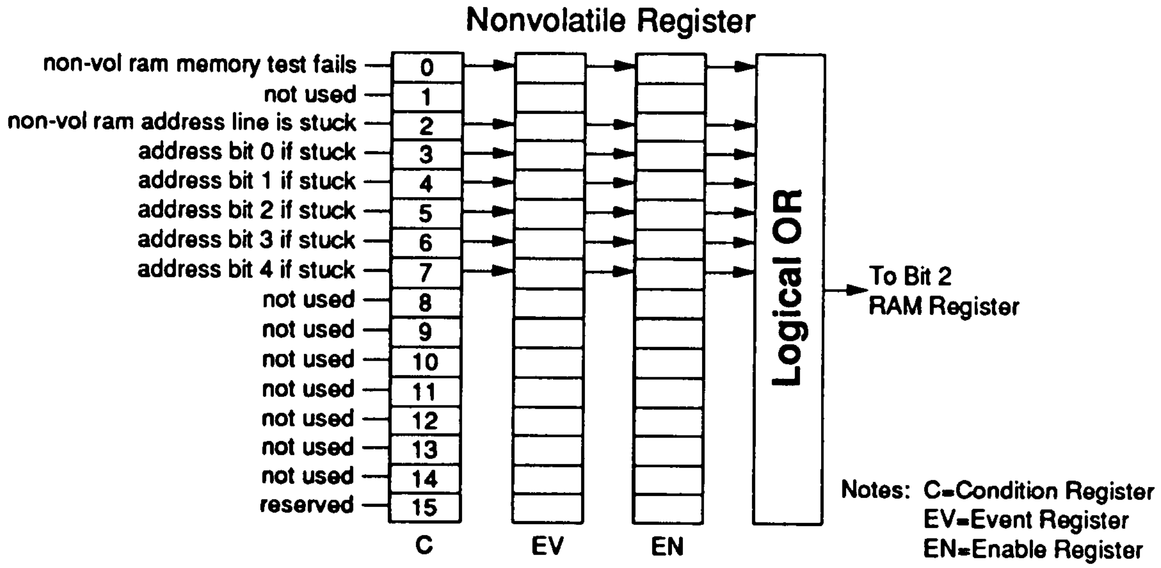


Example **Query display RAM event register**

`STAT:QUES:TEST:RAM:DISP?` *Query instrument to return register contents*

:QUEStionable:TEST:RAM:NVOLatile

STATus:QUEStionable:TEST:RAM:NVOLatile register reports nonvolatile random access memory diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.



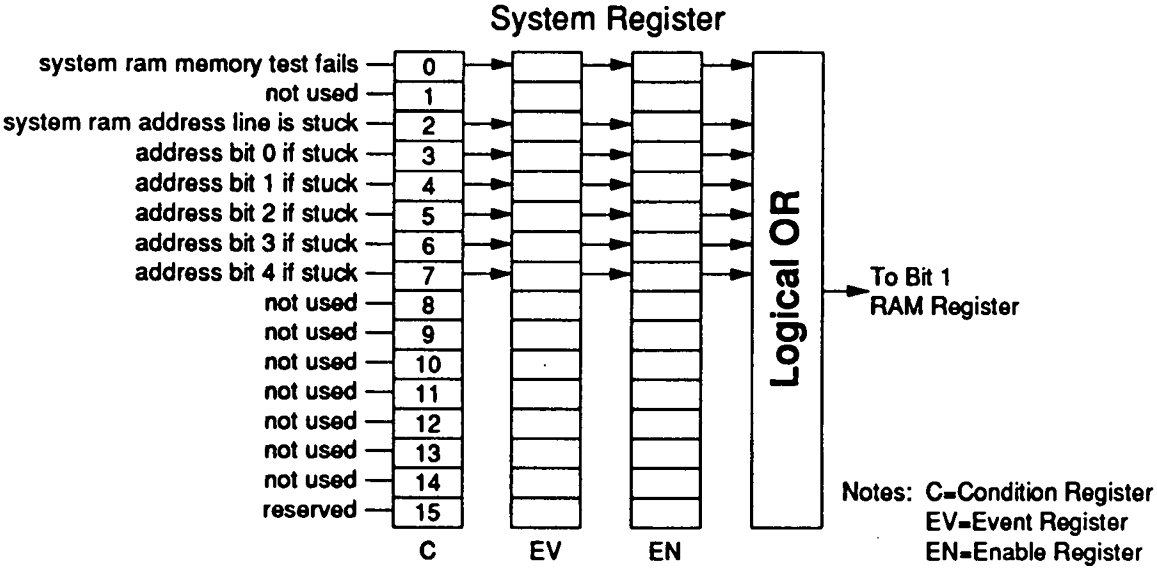
Example **Query nonvolatile RAM event register**

```
STAT:QUES:TEST:RAM:NVOL?
```

Query instrument to return register contents

:QUEStionable:TEST:RAM:SYSTem

STATus:QUEStionable:TEST:RAM:SYSTem register reports system random access memory diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

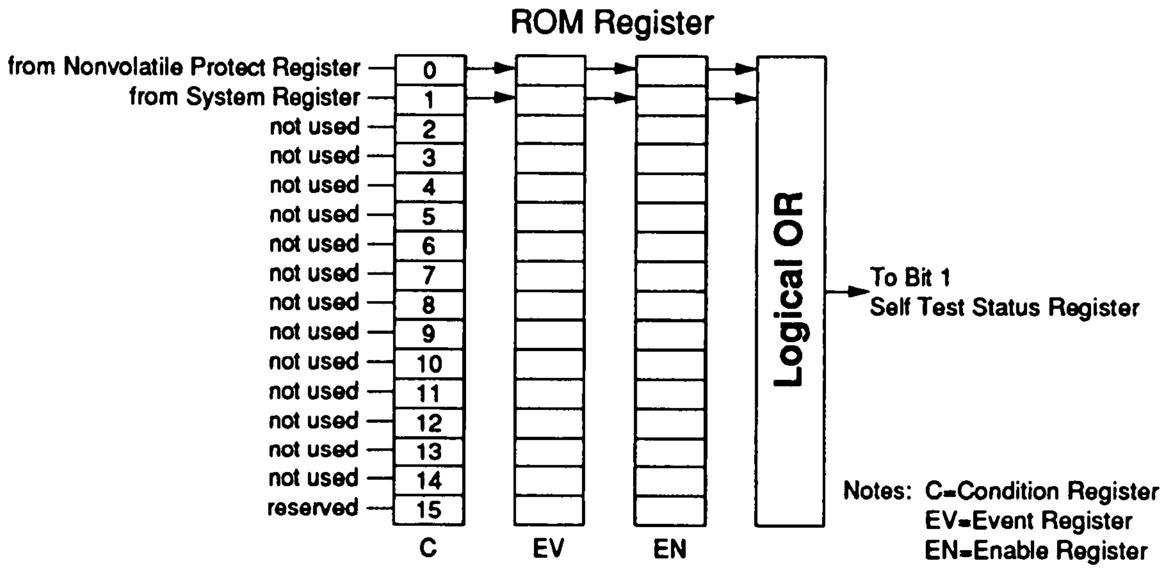


Example **Query system RAM event register**

`STAT:QUES:TEST:RAM:SYST?` *Query instrument to return register contents*

:QUEStionable:TEST:ROM

STATus:QUEStionable:TEST:ROM register reports read only memory diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.



Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query ROM event register**

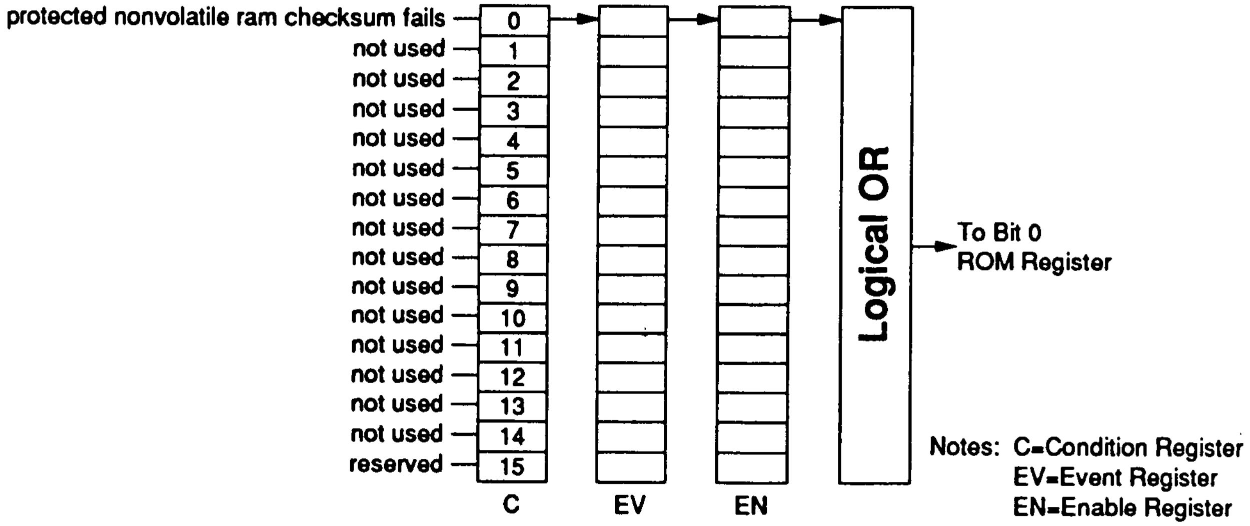
```
STAT:QUES:TEST:ROM?
```

Query instrument to return register contents

:QUEStionable:TEST:ROM:NPRotect

STATus:QUEStionable:TEST:ROM:NPRotect register reports non-volatile protected random access memory diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.

Nonvolatile Protect Register

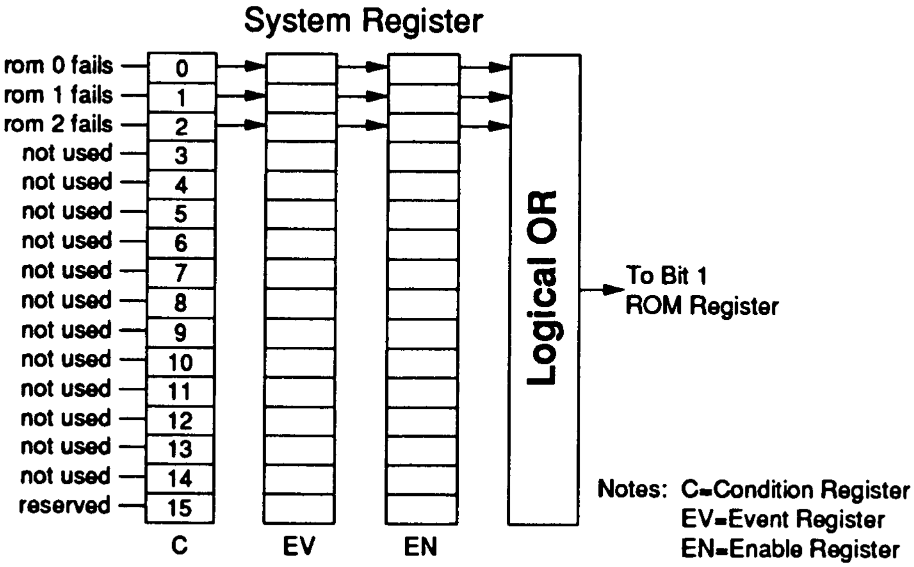


Example **Query non-volatile protected ROM event register**

`STAT:QUES:TEST:ROM:NPR?`	*Query instrument to return register contents*

:QUEStionable:TEST:ROM:SYSTem

STATus:QUEStionable:TEST:ROM:SYSTem register reports system read only memory diagnostic test results. Use the diagram to interpret returned results. See figure 6-2 and 6-3 for additional information on using the CONDition?, ENABle, ENABle?, and [:EVENt]? commands/queries.



Example **Query system ROM event register**

`STAT:QUES:TEST:ROM:SYST?` *Query instrument to return register contents*

SYSTem

The SYSTem command subsystem is used to define the programming language used, enable reading and writing to the advisory line of the instrument, read the SCPI version, and perform an autoscale function.

Subsystem Syntax

```
SYSTem
    :AUToscale
    :ERRor?
    :LANGuage <command>
    :LANGuage?
    :SERial <string>
    :SET <setup>
    :SET?
    :VERSion?
```

:AUToscale

The AUToscale command causes the instrument to evaluate all input signals, and then set the correct conditions to present the signals.

Example

Execute an autoscale

```
AUT                         Initiate an autoscale
```

Comments

- **Controls Affected:** The following controls are set to present the input signals:

  [SENSe:]INPut<n>:RANGe all (range and offset) as required
  CALCulate:MATH<n> to OFF
  INITiate:CONTinuous to OFF
  [SENSe:]SWEep:TIME as required
  TRIGger:ECOunt as required
  TRIGger:LEVel as required
  TRIGger:SLOPe as required

- **More than One Input Signal:** If signals are present on more than one input, the sweep will be triggered on the signal closest to input 1. If a signal is not present on input 1, then the instrument will be triggered on input 2. If a signal is not present on input 2, then the instrument will be triggered on input 3, etc.
- **No Input Signal:** If no signals are found on any input, the instrument will be returned to its former state.
- **Channel Coupling:** If a large offset is present on the input signal, coupling may change from DC to AC.

:ERRor?

SYSTem:ERRor? returns the next error number and corresponding error message in the error queue. See Appendix B for a listing of error numbers and messages.

Example **Read the next error number in the error queue**

`dimension statement`	*Dimension a string*
`SYST:ERR?`	*Query instrument to return the next error number and message*
`enter statement`	*Enter value into computer*

Comments

- **Error Numbers/Messages in the Error Queue.** Each error generated by the instrument stores an error number and corresponding error message in the error queue. The error message can be up to 255 characters long.
- **Clearing the Error Queue:** An error number/message is removed from the queue each time the SYSTem:ERRor? query is sent. The errors are cleared first-in, first-out. When the queue is empty, each following SYSTem:ERRor? query returns 0, "No error". To clear all error numbers/messages in the queue, execute the *CLS command.
- **Maximum Error Numbers/Messages in the Error Queue:** The queue holds a maximum of 30 error numbers/messages. If the queue overflows, the last error number/message in the queue is replaced by –350, "Too many errors". The least recent error numbers/messages remain in the queue and the most recent are discarded.
- ***RST Condition:** *RST does not clear the error queue.

:LANGuage

SYSTem:LANGuage *<command>* used to select the programming language. COMPatible selects the HP 54503A Compatible Language, and SCPI selects the Standard Commands for Programmable Instruments programming Language.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
command	discrete	COMPatible|SCPI	none

Example

Select SCPI to instruct the Oscilloscope

```
SYST:LANG  SCPI          Select SCPI language
```

Comments

- **Selecting Command:** When the HP 54503A Compatible Language (COMPatible) is selected, use the instructions found in Chapters 3 and 4 to program the Oscilloscope. When Standard Commands for Programmable Instruments (SCPI) is selected, use the instructions found in Chapters 5 and 6 to program the Oscilloscope.
- **Switching Languages:** Switching languages while programming is permissible, however:

  Allow 1 second after changing for the Oscilloscope to configure to the new language.

  After switching languages, a *RST is automatically performed to place the instrument in a known state.
- **Programming the Wrong Language:** If the Oscilloscope is configured to operate using one language, and a command from the other language is executed (with different syntax), an error will be generated.
- ***RST Condition:** *RST does not change SYSTem:LANGuage selected.

:LANGuage?

SYSTem:LANGuage? returns the current programming language selected. Returns COMPatible if the HP 54503A Compatible Language is selected, and SCPI if the Standard Commands for Programmable Instruments programming Language is selected. The data is sent to the output buffer.

Example

Querying the current programming language selected

```
dimension  statement   Dimension a string
SYST:LANG?             Query instrument to return current
                       programming language
enter  statement       Enter value into computer
```

:SERial

SYSTem:SERial *<string>* used to enter a serial number in the instrument. As the instrument serial number is entered at the factory, do not use this command unless there is a need to serialize the instrument for a different application.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
string	numeric	Alpha-numeric, no special	none

Example

Enter a different serial number

```
SER  "1234A56789"        Different serial number
```

Comments

- **Entering Serial Number *string*:** Serial number consists of 10 alpha-numeric digits enclosed in quotes (").
- **Protect Switch:** The calibration protection switch must be set to the **UNPROTECTED** position to write a new serial number to the protected non-volatile ram within the instrument.
- **Serial Number versus *IDN?:** The serial number is part of the string returned for the *IDN? query.

:SET

SYSTem:SET *<setup>* is used to set the Oscilloscope to a condition defined by a previously returned learn string. The learn string contains all the commands and parameters necessary to setup the instrument in one 1024 byte string.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
setup	block	binary block data in # format	none

Example See SYSTem:SET? query for example

Comments

- **Using SET:** The logical order for using this instruction is to send the query first to retrieve setup data, store the data until needed, then send the learn string to the instrument using this command.
- **SET versus *SAV/*RCL:** The SYSTem:SET command performs the same function as the save and recall commands, except:

  Data can be saved at any location the user desires.

  No limit to the number of setups that can be saved/recalled.

:SET?

SYSTem:SET? returns the current learn string from the Oscilloscope. The learn string contains all the commands and parameters the Oscilloscope is currently setup to in one 1024 byte string and a header.

Example **Query setup learn string**

For this example, the learn string is "#41024...."

`Dimension statement`	*Dimension string for data*
`Setup statements`	*Setup Oscilloscope as desired using commands described in this chapter*
`SYST:SET?`	*Query instrument to return setup learn string*
`enter statement`	*Enter value into computer*
`store statement`	*Save data*
`change instrument`	*Set the Oscilloscope to perform a different function.*
`recall statement`	*Recall data*
`SYST:SET #41024....`	*Send data to the Oscilloscope (recalls previous setup)*

Comments

- Related Commands: *LRN?, *SAV, *RCL.

:VERSIon?

SYSTem:VERSion? returns the current SCPI version number the instrument complies with. The data is sent to the output buffer.

Example

Return the instruments SCPI version number

`SYST:VERS?`	*Query instrument to return version number*
`enter statement`	*Enter value into computer*

Comments

- **Returned Format:** Return data is in the form YYYY.V, where YYYY is the year-version, and the V is the revision number for that year. V=0 if no approved revisions are claimed.
- ***RST Conditions:** *RST does not effect revision number.

TEST

The TEST command subsystem is used to perform internal diagnostics. These diagnostics are provided to give a high confidence level of instrument functionality. Before performing any of the diagnostics, execute a *RST to set critical parameters to a known state, and a STATus:PRESet to enable the STATus:QUEStionable registers.

Subsystem Syntax

```
TEST
    :ACQ [<test>]
    :RAM [<test>]
    :ROM [<test>]
    :TALL
```

:ACQ

TEST:ACQ [<*test*>] is used to perform up to five acquisition tests. When selected, the Oscilloscope performs an Analog Trigger test, Logic Trigger test, an A/D test, a Time base test, and/or a D/A test. If the *test* parameter is not sent, all five tests are performed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
test	discrete	ATRigger|LTRigger| AD|TIMebase|DA	none

Example **Perform the acquisition time base test**

```
TEST:ACQ  TIM
```
Perform acquisition time base test

Comments

- **Test Results:** Found by querying the STATus:QUEStionable:TEST:ACQ register.
- **Test Failure:** If any of the five acquisition tests fail, perform Firmware Calibration procedures provided in Chapter 5, and repeat the test. If the test fails again, refer to instructions found in the Service Manual.
- **Related Commands:** STATus:QUEStionable:TEST.

:RAM

TEST:RAM [<*test*>] is used to perform up to four random access memory tests. When selected, the Oscilloscope performs a Display RAM test, System RAM test, a Non-volatile RAM test, and/or an Acquisition RAM test. If the *test* parameter is not sent, all four tests are performed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
test	discrete	DISPlay|ACQuisition| SYSTem|NVOLatile	none

Example **Perform all four RAM tests**

```
TEST:RAM                    Perform RAM test
```

Comments

- **Test Results:** Found by querying the STATus:QUEStionable:TEST:RAM register.
- **Test Failure:** If any of the four RAM tests fail, perform Firmware Calibration procedures provided in Chapter 5, and repeat the test. If the test fails again, refer to instructions found in the Service Manual.
- **Related Commands:** STATus:QUEStionable:TEST.

:ROM

TEST:ROM [<*test*>] is used to perform one read only memory test and one non-volatile protected random access memory test. When selected, the Oscilloscope performs a System ROM test, and/or a Protected Non-volatile RAM test. If the *test* parameter is not sent, both tests are performed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
test	discrete	SYSTem|NVPRotect	none

Example **Perform the system ROM test**

```
TEST:ROM  SYST              Perform system ROM test
```

Comments

- **Test Results:** Found by querying the STATus:QUEStionable:TEST:ROM register.
- **Test Failure:** If any of the two ROM tests fail, perform Firmware Calibration procedures provided in Chapter 5, and repeat the test. If the test fails again, refer to instructions found in the Service Manual.
- **Related Commands:** STATus:QUEStionable:TEST.

:TALL

TEST:TALL is used to perform the RAM, ROM, and ACQ tests. When selected, the Oscilloscope performs all the individual tests.

Example

Perform the RAM, ROM, and ACQ tests

```
TEST:TALL                    Perform all tests
```

Comments

- **User Connection:** Disconnect all inputs prior to performing self tests.
- **Test Results:** Found by querying the STATus:QUEStionable:TEST register.
- **Test Failure:** If any of the tests fail, perform Firmware Calibration procedures provided in Chapter 5, and repeat the test. If the test fails again, refer to instructions found in the Service Manual.
- **Related Commands:** STATus:QUEStionable:TEST.

TRACe

The TRACe command subsystem is used to transfer waveform data. The transfer can take place internally (memories within the instrument) or externally (bus and controller).

Internal: Transfer of data between pixel memories, waveform memories, and inputs. Destination and source are specified using one command (DATA), and data is transferred.

External: Transfer of data between the bus and the instrument's waveform or pixel memories. The waveform record is actually contained in two portions, the waveform data and the preamble.

> The waveform data is the actual data acquired for each point in the specified input.
>
> The preamble contains the information for interpreting the waveform data. This includes the number of points acquired, format of acquired data, and type of acquired data. The preamble also contains the X and Y increments, origins, and references for the acquired data, so that the returned data can be translated to time and voltage values.

The waveform data and preamble must be read (by the controller) or sent (to the instrument) with two separate commands, DATA (?) and PREamble (?).

Note

The term "Time Buckets" is defined as — the time range divided into a specific number of horizontal time points (as defined by [SENSe:]AVERage:POINts), each with an equal and fixed time associated with it.

Subsystem Syntax

```
TRACe:
    [:DATA] <destination>,<source>
    [:DATA]? <source>
    :POINts? <source>
    :PREamble <destination>,<data>
    :PREamble? <source>
```

[:DATA]

TRACe[:DATA] *<destination>,<source>* is used to transfer waveform data. By specifying different *source* and *destination* parameters, the command can be used to:

- Erase pixel memory.
- Merge pixel memory.
- Store waveform data in waveform memory.
- Send the instrument a waveform data record over the bus and store it in the previously specified memory.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Unit
destination	discrete	PMEMoryn (*n*=1 to 2)| WMEMory*n*(*n*=1 to 4) See comments before selecting	none
source	discrete	0|<*data*>|PMEMory0| INPut*n*(*n*=1 to 4)| WMEMory*n*(*n*=1 to 4)| MATH*n*(*n*=1 to 2) See comments before selecting	none
data	block	binary block data in # format	none

Example

Send the Oscilloscope waveform data and place in pixel memory 1

For the example, waveform data is "#41024...."

```
TRAC  PMEM1,#41024....      Send waveform data to
                            Oscilloscope and place in
                            PMEM1
```

Comments

- **Selecting Source and Destination:** Depending on the desired action, make selections of source and destination as follows:

  Erase Pixel Memory: To erase a pixel memory location, enter *destination* of the location to be erased (PMEMory 1 or 2), and *source* to 0. Once the command is executed, there is no way to retrieve the original data. PMEMory0 cannot be erased.

  Merge Pixel Memory: To merge the current contents of pixel memory 0 (active waveform) with the current contents of pixel memory 1 or 2, set *destination* to PMEMory 1 or 2, and *source* to PMEMory0 (active waveform). Once the command is executed, there is no way to retrieve the original PMEMory1 or 2 data.

Store Data in Waveform Memory (from internal location): To store waveform data in one of four Waveform Memories, set *destination* to where data will be stored (WMEMory1 to 4), and *source* to where data is currently (WMEMory1 to 4, INPut1-4, MATH1-2). Pixel Memories cannot be stored.

Store Data in Waveform Memory (from external location): To store waveform data in one of four Waveform Memories, set *destination* to where data will be stored (WMEMory1 to 4). *source* is the actual waveform *data* (binary block data in # format) received over the bus. The format of the waveform data being sent must match the format previously specified by the waveforms preamble for the destination memory. See FORMat[:DATA] command for more information. When specifying the source, use the following guidelines:

> ENVelope mode requires Waveform MEMory pairs (WMEM1 and 3, or WMEM 2 and 4) to transfer data. Specify WMEMory1 for WMEM1 and 3, or WMEMory2 for WMEM2 and 4. The data is transferred as two arrays. For example, if WMEM1 is specified as the source, the first array is transferred into WMEMory 1 and the second array is transferred into WMEMory 3. The data type is then changed to normal for each of the waveform memories.
>
> SCALar and AVERage modes are transferred to the selected Waveform MEMory (WMEM1, 2, 3, or 4).

Store Data in Pixel Memory (from external location): To store waveform data in one of two Pixel Memories, set *destination* to where data will be stored (PMEMory1 to 2). *source* is the actual waveform *data* (binary block data in # format) received over the bus.

- **Waveform Data Format:** The format of the waveform data being sent must match the format previously specified by the waveforms preamble for the destination memory. See FORMat[:DATA] command for more information.
- **Related Commands:** FORMat[:DATA], TRACe[:DATA]?, TRACe:PREamble.

[:DATA]?

TRACe[:DATA]? <*source*> is used to output the waveform data record stored in the instruments Waveform MEMory, Pixel MEMory, MATH function, or input buffer over the bus.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Unit
source	discrete	PMEMory*n* (*n*=0 to 2)| INPut*n* (*n*=1 to 4)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2) See comments before selecting	none

Example 1

Send waveform data from the oscilloscope over the bus

The following example illustrates the use of the TRACe[:DATA]? query only. Chapter 5 contains an example of performing a complete digitizing operation, and also the procedure used to read the block length then re-define a string to hold the data.

`Dimension statement`	*String to hold data*
`TRAC? INP3`	*Enter value from input 3 input buffer into computer (see Chapter 5, Digitize example, for the procedure to read block lengths*
`enter statement`	*Enter data into computer*

Comments

- **Waveform Data Source:** Waveform MEMories (WMEM1-4), Pixel MEMories (PMEM0-2), MATH functions (MATH1-2), or input buffers (INP1-4) may have waveform data sent from them. Select the desired location using the *source* parameter. When PMEM0-2 is selected, the current FORMat[:DATA] and [SENSe:]AVERage[:STATe] and TYPE selections are disregarded.
- **Waveform Data Format:** The format of the waveform data being sent is specified using the FORMat[:DATA] command. FORMat does not apply to pixel data (PMEM0-2).

- **Selecting Type:** The type of waveform acquisition is selected by the [SENSe:]AVERage:TYPE command. Acquisition TYPE does not apply to pixel data (PMEM0-2).

  SCALar: SCALar data consists of the last data point (hit) in each time bucket. This data is transmitted over the bus in a sequential fashion starting with time bucket 0 and going through time bucket *n*-1, where *n* is the number returned by the TRACe:POINts? query. Time buckets that don't have data in them return –1. Only the magnitude values of each data point are transmitted, the time values correspond to the position in the data array. The first voltage value corresponds to the first time bucket on the left of the active waveform and the last value corresponds to the next to last time bucket on the right of the active waveform.

  AVERage: Average data consists of the average of the first *n* hits in a time bucket, where *n* is the value returned by the [SENSe:]AVERage:COUNt? query. Time buckets that have fewer than *n* hits return the average of what data they do have. If the [SENSe:]SWEep:POINts:COMPlete parameter is set to 100%, then each time bucket must contain the number of data hits specified with the [SENSe:]AVERage:COUNt command. Again, if a time bucket doesn't have any data in it, it will return –1. This data is transmitted over the bus in linear fashion starting with time bucket 0 and proceeding through time bucket *n*-1, where *n* is the number returned by the TRACe:POINts? query. The first value corresponds to a point at the left of the active waveform and the last value is one point away from the right of the active waveform.

  ENVelope: Envelope data consists of two arrays of data, one containing the minimum of the first *n* hits in each time bucket and the other containing the maximum of the first *n* hits in each time bucket, where *n* is the value returned by the [SENSe:]AVERage:COUNt? query. If a time bucket does not have any hits in it, then –1 is returned for both the minimum and maximum values. The two arrays are transmitted one at a time over the bus linearly, starting with time bucket 0 (on the left of the active waveform) and proceeding through time bucket *n*–1, where *n* is the value returned by the TRACe:POINts? query. The array with the minimum values is sent first. The first value of each array corresponds to the data point on the left of the of the active waveform. The last value is one data point away from the right of the active waveform.

- **Interpreting Waveform Data:** In order to obtain useful information from the waveform data, the returned data must be scaled. The information necessary for scaling (X/Y) the waveform is contained in the preamble data.

Conversion from Data Value to Voltage: The formula to convert a data value from the specified source to a voltage value is:

voltage = [(data value-yreference)•yincrement]+yorigin

Conversion from Data Value to Time: The time value of a data point can be determined by the position of the data point. The formula to convert a data point from the specified source to a time value is:

time = [(data point number-xreference)•xincrement]+xorigin

As an example, the third data point sent with xorigin = 16 ns, xreference = 0, and xincrement = 2 ns, would result in the following calculation:

time = [(3 − 0) • 2 ns] + 16 ns = 22 ns

- **Related Commands:** TRACe:PREamble, FORMat[:DATA], [SENSe:]AVERage:TYPE.

:POINts?

TRACe:POINts? ***<source>*** returns the points value in the currently selected (*source*) waveform preamble. The points value is the number of time buckets contained in the waveform selected with the TRACe[:DATA]? query. The value is sent to the output buffer.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Unit
source	discrete	INPut*n* (*n*=1 to 4) WMEMory*n*(*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Query current points value of waveform memory 1

`TRAC:POIN? WMEM1`	*Query instrument to return points value of waveform memory 1*
`enter statement`	*Enter data into computer*

Comments

- **Returned Results:** In most cases the number of time buckets actually acquired will be the number of points set with the [SENSe:]AVERage:POINts command. There are some [SENSe:]SWEep:TIME settings where the actual number of points will be less than requested, as shown below.

  SWEep:TIME to 2 nsec - the number of points actually acquired will be 32, 64, or 100.

  SWEep:TIME to 20 nsec - the number of points actually acquired will be 32, 64, 128, or 200.

  SWEep:TIME to 50 nsec - the number of points actually acquired will be 32, 64, 128, 256, 500, 512, or 1000.

- **Related Commands:** [SENSe:]SWEep:TIME, [SENSe:]AVERage:POINts.

:PREamble

TRACe:PREamble *<destination>,<data>* is used to send the instrument waveform preamble over the bus and store it in the previously specified waveform memory.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Unit
destination *data*	discrete numeric	WMEMory*n*(*n*=1 to 4) See comments below	none none

Example

Send the Oscilloscope waveform preamble to waveform memory 1

For the example, waveform preamble is "xxxx...."

```
TRAC:PRE  WMEM1,xxxx....     Send waveform preamble to
                             waveform memory 1
```

Comments

- **Storing Waveform Preamble:** Only Waveform MEMories (WMEM1-4) may have a waveform preamble sent to them. The desired location is selected using *source*.
- **Waveform Preamble Format:** The format of the preamble data is as follows.

  <data> = <format NR1 >,<type NR1>,<points NR1>,<count NR1>, <xincrement NR3>, <xorigin NR3>, <xreference NR3>, <yincrement NR3>, <yorigin NR3>, <yreference NR3>
  Where:

<format> =	1 for 8 bit format 2 for 16 bit format
<type> =	1 for SCALar type 2 for AVERage type 3 for ENVelope type
<points > =	See TRACe:POINts? query.
<count>=	See [SENSe:]AVERage:COUNt? query.
<xincrement>=	See TRACe[:DATA]? query.
<xorigin>=	See TRACe[:DATA]? query.
<xreference>=	See TRACe[:DATA]? query.
<yincrement>=	See TRACe[:DATA]? query.
<yorigin>=	See TRACe[:DATA]? query.
<yreference> =	See TRACe[:DATA]? query.

- **Related Commands:** TRACe:PREamble?.

:PREamble?

TRACe:PREamble? ***<source>*** sends a waveform preamble stored in the specified Waveform MEMory, MATH function, or input buffer over the bus.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Unit
source	discrete	INPut*n* (*n*=1 to 4) WMEMory*n*(*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Send waveform preamble from input 1 buffer the bus

The following example illustrates the use of the TRACe:PREamble? query only. Chapter 5 contains an example on performing a complete digitizing operation.

`Dimension statement`	*Dimension string or array*
`TRAC:PRE? INP1`	*Send waveform preamble over the bus*
`enter statement`	*Enter data into computer*

Comments

- **Waveform Preamble:** Waveform MEMories (WMEM1-4), MATH functions (MATH1-2), or input buffers (INP1-4) may have waveform preamble sent from them, as selected by *source*.

  Preamble Data: The values set in the preamble are determined when the INITiate[:IMMediate] command is executed. The Preamble values are based on the settings of variables in the [SENSe:] subsystem. Although the preamble values can be changed with a controller, the way the data was acquired cannot be changed. Changing the preamble values cannot change the type of data that was actually acquired, the number of points actually acquired, etc. Therefore, extreme caution must be used when changing any waveform preamble values to ensure the data will still be useful. For example, setting POINts in the preamble to a value different from the actual number of points in the waveform will result in inaccurate data.

- **Waveform Preamble Returned Format:** The returned information can be read into a numeric string or an array. The format of the waveform preamble is explained in the TRACe:PREamble command.

- **Related Commands:** TRACe:PREamble.

TRIGger

The TRIGger command subsystem is used to define the conditions for a trigger. The Hewlett-Packard 54503A Compatible Language (COMP) offers a choice of complex trigger modes. See Chapter 4, TRIGger for more information.

Auto or triggered mode is selected with the INITiate:CONTinuous command.

Subsystem Syntax

```
TRIGger
    :ECOunt <events>
    :ECOunt?
    :HYSTeresis <mode>
    :HYSTeresis?
    :LEVel <level>
    :LEVel?
    :SLOPe <polarity>
    :SLOPe?
    :SOURce <source>
    :SOURce?
```

:ECOunt

TRIGger:ECOunt *<events>* is used to enable the trigger circuit on a specified trigger event.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
events	numeric	1, 3 to 16,000,000	none

ECOunt 1 sets holdoff-by-time mode with time set to 40 ns. ECOunt 3 through 16,000,000 enables trigger event mode.

Example **Enable the trigger circuit on the 8th trigger event**

```
TRIG:ECO  8                Trigger on the 8th edge
```

Comments

- **Event Source:** The source for counted *events* is specified using the TRIGger:SOURce command.
- **Related Commands:** TRIGger:SOURce.
- ***RST Conditions:** Defaults to 1

:ECOunt?

TRIGger:ECOunt? is used to return the currently selected number of counts (1 or from 3 to 16,000,000). The data is sent to the output buffer. See TRIGger:ECOunt command for more information.

Example **Query the current count selection**

```
TRIG:ECO  8                Trigger on 8th edge
TRIG:ECO?                  Query instrument to return count
                           setting
enter  statement           Enter data into computer
```

:HYSTeresis

TRIGger:HYSTeresis *<mode>* enables or disables use of the noise reject function. Used to avoid false triggering on noisy signals. OFF corresponds to noise reject off and ON corresponds to noise reject on.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	ON|1|OFF|0	none

Example

Set noise reject to on

```
TRIG:HYST  ON                Noise reject on
```

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **Specifying Source:** Select the source using the TRIGger:SOURce command.
- **Related Commands:** TRIGger:LEVel, TRIGger:SOURce.
- ***RST Conditions:** Defaults to OFF.

:HYSTeresis?

TRIGger:HYSTeresis? returns the currently selected noise reject function state (LOW if ON, or HIGH if OFF) for the input specified. The value is sent to the output buffer.

Example

Query the noise reject selection

```
dimension  statement     String for data
TRIG:HYST  ON            Noise reject on
TRIG:HYST?               Query instrument to return hysteresis
                         setting
enter  statement         Enter data into computer
```

:LEVel

TRIGger:LEVel *<level>* is used to set the trigger level voltage of the active trigger.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
level	numeric	See below	V

Example

Set trigger level to 1 volt

```
TRIG:LEV  1                Trigger level to 1 volt
```

Comments

- **Selecting Level:** *level* can be entered to a value that is ±0.75 of the current VOLTage<n>:RANGe[:PTPeak] setting from the current VOLTage<n>:RANGe:OFFSet setting.
- **Trigger Level Source:** The trigger *level* source is selected using the TRIGger:SOURce command.
- **Related Commands:** TRIGger:SOURce.
- ***RST Conditions:** Defaults to 0 volts.

:LEVel?

TRIGger:LEVel? returns the currently selected trigger level (in volts). The value is sent to the output buffer.

Example

Query the current trigger level

```
TRIG:LEV  1                Trigger level to 1 volt
TRIG:LEV?                  Query instrument to return trigger
                           level
enter  statement           Enter data into computer
```

:SLOPe

TRIGger:SLOPe ***<polarity>*** is used to select the edge for the trigger.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
polarity	discrete	POSitive|NEGative	none

Example

Set trigger edge to rising

```
TRIG:SLOP  POS            Slope to positive
```

Comments

- **Selecting Slope:** Enter POSitive to select the rising edge, and NEGative to select the falling edge.
- **Related Commands:** TRIGger:SOURce.
- ***RST Condition:** Defaults to POSitive.

:SLOPe?

TRIGger:SLOPe? returns the selected trigger edge for the currently selected trigger source. The data is sent to the output buffer. Returns POSitive if the rising edge is selected, or NEGative if falling edge is selected.

Example

Query the current slope selection

```
dimension  statement  String to hold data
TRIG:SLOP  POS        Slope to positive
TRIG:SLOP?            Query instrument to return slope
                      setting
enter  statement      Enter data into computer
```

:SOURce

TRIGger:SOURce ***<source>*** **is used to select the source that will actually produce the trigger.**

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	INPut*n* (*n*=1 to 4)| TTLTrg*n* (*n*=0 to 7)| ECLTrg*n* (*n*=0 to 1)	none

Example

Set trigger source to TTL trigger bus line 5

```
TRIG:SOUR  TTLT5          Source to TTL trigger bus line 5
```

Comments

- **Selecting Source:** *source* is specified depending on the desired source as follows:

  INPut1-4: Source is input connectors 1-4 on the instrument panel.

  TTLTrg0-7: Source is backplane TTL trigger bus lines 0-7. TRIGger:ECOunt, LEVel, and SLOPe do not effect TTL trigger sources.

  ECLTrg0-1: Source is backplane ECL trigger bus lines 0-1. TRIGger:ECOunt, LEVel, and SLOPe do not effect ECL trigger sources.

- **SYSTem:AUToscale:** Autoscale selects the trigger source from INPut 1-4 only. TTLTrg and ECLTrg lines cannot be used for an autoscale.

- ***RST Condition:** Defaults to INPut1.

:SOURce?

TRIGger:SOURce? returns the selected trigger source (INPut1-4, TTLTrg0-7, or ECLTrg0-1). The data is sent to the output buffer.

Example

Query the current trigger source selection

```
dimension  statement   String to hold data
TRIG:SOUR  TTLT5       Source to TTL trigger bus line 5
TRIG:SOUR?             Query instrument to return trigger
                       source setting
enter  statement       Enter data into computer
```

Command Cross Reference to COMP Commands

The following table is provided as a quick cross reference of all applicable Standard Commands for Programmable Instruments (SCPI) commands to the similar Hewlett-Packard 54503A Compatible Language (COMP) command(s).

SCPI Command	COMP Command	Description
ABORt	STOP	Command operates the same in both languages.
CALCulate		
:MATH<n>		
[:EXPRession]	FUNCtion<n>:ADD FUNCtion<n>:MULTiply FUNCtion<n>:SUBTract	Add, subtract, and multiply command operates the same in both languages.
CALibration	CALibrate	The SCPI CALibration and COMP CALibrate subsystems operate the same in both languages.
FORMat		
[:DATA]	WAVeform:FORMat	In COMP, BYTE is INT 8 and WORD is INT 16.
[:DATA]?	WAVeform:FORMat?	In COMP, BYTE is returned for 8 and WORD is returned for 16.
INITiate		
:CONTinuous ON	RUN and TIMebase:MODe TRIGgered	Command operates the same in both languages.
:CONTinuous AUTO	RUN and TIMebase:MODe AUTo	Command operates the same in both languages.
[:IMMediate]	DIGitize	Command operates the same in both languages.
MEASure		
[:SCALar]		
:VOLTage		
:[DC]? [(@)]	MEASure:VAVerage?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:AC? [(@)]	MEASure:VACrms? MEASure:VRMS?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:AMPLitude? [(@)]	MEASure:VAMPlitude?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:DCYCle? [ref] [(@)]	MEASure:DUTYcycle?	Command operates the same in both languages, except source, and the reference level are specified with the MEASure:SOURce, UPPer, and LOWer commands in COMP.

Command Cross Reference to COMP Commands — Continued

SCPI Command	COMP Command	Description
MEASure [:SCALar] :VOLTage :FALL		
:OVERshoot? [(@)]	MEASure:OVERshoot?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:PREShoot? [(@)]	MEASure:PREshoot?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:TIME? [limits] [(@)]	MEASure:FALLtime?	Command operates the same in both languages, except source, and the upper and lower thresholds are specified with the MEASure:SOURce, UPPer, and LOWer commands in COMP.
:FREQuency? [(@)]	MEASure:FREQuency?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:FTIMe? [limits][(@)]	MEASure:FALLtime?	Command operates the same in both languages, except source, and the upper and lower thresholds are specified with the MEASure:SOURce, UPPer, and LOWer commands in COMP.
:HIGH? [(@)]	MEASure:VTOP?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:LOW? [(@)]	MEASure:VBASe?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:MAXimum? [(@)]	MEASure:VMAX?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:MINimum? [(@)]	MEASure:VMIN?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:NWIDth? [(@)]	MEASure:NWIDth?	Command operates the same in both languages, except source, and the reference level are specified with the MEASure:SOURce, UPPer, and LOWer commands in COMP.
:PERiod? [(@)]	MEASure:PERiod?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:PDUTycycle? [<ref>][(@)]	MEASure:DUTYcycle?	Command operates the same in both languages, except source, and the reference level are specified with the MEASure:SOURce, UPPer, and LOWer commands in COMP.
:PWIDth? [<ref>][(@)]	MEASure:PWIDth?	Command operates the same in both languages, except source, and the reference level are specified with the MEASure:SOURce, UPPer, and LOWer commands in COMP.

Command Cross Reference to COMP Commands — Continued

SCPI Command	COMP Command	Description
MEASure		
[:SCALar]		
:VOLTage		
:RISE		
:OVERshoot? [(@)]	MEASure:OVERshoot?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:PREShoot? [(@)]	MEASure:PREshoot?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:TIME? [limits][(@)]	MEASure:RISEtime?	Command operates the same in both languages, except source, and the upper and lower thresholds are specified with the MEASure:SOURce, UPPer, and LOWer commands in COMP.
:RTIMe? [limits][(@)]	MEASure:RISEtime?	Command operates the same in both languages, except source, and the upper and lower thresholds are specified with the MEASure:SOURce, UPPer, and LOWer commands in COMP.
:TMAXimum? [(@)]	MEASure:TMAX?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
:TMINimum? [(@)]	MEASure:TMIN?	Command operates the same in both languages, except source is specified using the MEASure:SOURce command in COMP.
OUTPut	OUTPut	The Compatible SCPI OUTPut subsystems operate the same in both languages.
[SENSe:]		
AVERage		
:COUNt	ACQuire:COUNt	Command operates the same in both languages.
:COUNt?	ACQuire:COUNt?	Command operates the same in both languages.
[:STATe]	ACQuire:TYPe	In COMP, NORMal is the equivalent of scalar, and mode selected using the ACQuire:TYPe command only.
[:STATe]?	ACQuire:TYPe?	See AVERage[:STATe] for an explanation.
:TYPE	ACQuire:TYPe	In COMP, NORMal is the equivalent of scalar, and mode selected using the ACQuire:TYPe command only.
:TYPE?	ACQuire:TYPe?	See AVERage:TYPE for an explanation.
CORRection<n>		
:AFACtor	CHANnel<n>:PROBe	Command operates the same in both languages.
:AFACtor?	CHANnel<n>:PROBe?	Command operates the same in both languages.
INPut<n>		
:COUPling	CHANnel<n>:COUPling	In COMP, AC is the equivalent of selecting AC and 1MΩ, DC is the equivalent of selecting DC and 1MΩ, and DCFifty is the equivalent of selecting DC and 50Ω.
:COUPling?	CHANnel<n>:COUPling?	See INPut<n>:COUPling for an explanation.

Command Cross Reference to COMP Commands — Continued

SCPI Command	COMP Command	Description
[SENSe:]		
INPut<n>		
:FILTer		
:HPASs		
[:STATe]	CHANnel<n>:LFReject	Command operates the same in both languages.
[:STATe]?	CHANnel<n>:LFReject?	Command operates the same in both languages.
[:LPASs]		
[:STATe]	CHANnel<n>:HFReject	Command operates the same in both languages.
[:STATe]?	CHANnel<n>:HFReject?	Command operates the same in both languages.
:IMPedance	CHANnel<n>:COUPling	In COMP, AC is the equivalent of selecting AC and 1MΩ, DC is the equivalent of selecting DC and 1MΩ, and DCFifty is the equivalent of selecting DC and 50Ω.
:IMPedance?	CHANnel<n>:COUPling?	See INPut<n>:IMPedance for an explanation.
[:STATe] OFF	BLANk	COMP command allows channels, functions, and memories to be turned off.
[:STATe] ON	VIEW	COMP command allows channels, functions, and memories to be turned on.
[:STATe]?	STATus?	COMP command allows channels, functions, and memories to be turned queried.
SWEep		
:POINts	ACQuire:POINts	Command operates the same in both languages.
:POINts	ACQuire:POINts?	Command operates the same in both languages.
:COMPlete	ACQuire:COMPlete	Command operates the same in both languages.
:COMPlete?	ACQuire:COMPlete?	Command operates the same in both languages.
:TIME		
:DELay	TIMebase:DELay	Command operates the same in both languages.
:DELay?	TIMebase:DELay?	Command operates the same in both languages.
:LINK	TIMebase:REFerence	Command operates the same in both languages.
:LINK?	TIMebase:REFerence?	Command operates the same in both languages.
:RANGe	TIMebase:RANGe	Command operates the same in both languages.
:RANGe?	TIMebase:RANGe?	Command operates the same in both languages.
VOLTage<n>		
:RANGe		
:OFFSet	CHANnel<n>:OFFSet	Command operates the same in both languages.
:OFFSet?	CHANnel<n>:OFFSet?	Command operates the same in both languages.
[:PTPeak]	CHANnel<n>:RANGe	Command operates the same in both languages.
[:PTPeak]?	CHANnel<n>:RANGe?	Command operates the same in both languages.
STATus		
:OPERation	TER?	In COMP, trigger register separate for trigger bit.
:PRESet	SUMMary:PRESet	Command operates the same in both languages.
:QUEStionable	SUMMary:QUEStionable	The COMP and SCPI QUEStionable registers operate the same in both languages.

Command Cross Reference to COMP Commands — Continued

SCPI Command	COMP Command	Description
SYSTem		
:AUToscale	AUToscale	Command operates the same in both languages.
:ERRor?	SYSTem:ERRor?	Compatible command allows the message to be blanked.
:LANGuage	SYSTem:LANGuage	Command operates the same in both languages.
:LANGuage?	SYSTem:LANGuage?	Command operates the same in both languages.
:SERial	SERial	Command operates the same in both languages.
:SET	SYSTem:SETup	Command operates the same in both languages.
:SET?	SYSTem:SETup?	Command operates the same in both languages.
TEST	TEST	The Compatible SCPI TEST subsystems operate the same in both languages.
TRACe		
[:DATA]	DISPlay:DATa	In COMP, source is specified using the DISPlay:SOURce command.
	MERGe	In COMP, only the merge to location is specified.
	ERASe	In COMP, only the source being erased is specified.
	STORe	Command operates the same in both languages.
	WAVeform:DATa	In COMP, source is specified using the WAVeform:SOURce command.
[:DATA]?	WAVeform:DATa?	In COMP, source is specified using the WAVeform:SOURce command.
	DISPlay:DATa?	In COMP, source is specified using the DISPlay:SOURce command.
:POINts?	WAVeform:POINts?	Command operates the same in both languages.
:PREamble	WAVeform:PREamble	In COMP, source is specified using the WAVeform:SOURce command.
:PREamble?	WAVeform:PREamble?	In COMP, source is specified using the WAVeform:SOURce command.
TRIGger		
:ECOunt	TRIGger:HOLDoff	In COMP, holdoff is defined by time or events.
:ECOunt?	TRIGger:HOLDoff?	In COMP, holdoff is returned in time or events.
:HYSTeresis	TRIGger:SENSitivity	Command operates the same in both languages.
:HYSTeresis?	TRIGger:SENSitivity?	Command operates the same in both languages.
:LEVel	TRIGger:LEVel	Command operates the same in both languages.
:LEVel?	TRIGger:LEVel?	Command operates the same in both languages.
:SLOPe	TRIGger:SLOPe	Command operates the same in both languages.
:SLOPe?	TRIGger:SLOPe?	Command operates the same in both languages.
:SOURce	TRIGger:SOURce	Command operates the same in both languages.
:SOURce?	TRIGger:SOURce?	Command operates the same in both languages.

Common Command Reference

The following table lists the IEEE 488.2 Common (*) Commands that can be executed by the Oscilloscope module. The operation of some of these commands is described in Chapter 5 of this manual. For more information on Common Commands, refer to the HP E1405 Command Module User's Guide or the ANSI/IEEE Standard 488.2-1987.

Common Command Reference

Command	Title	Description
*CLS	Clear status register	Clears all Event Registers, the Request for OPC flag, and all Queues (except output queue).
*ESE <mask> *ESE?	Event status enable Event status enable query	Used to set the bits in the Event Status Enable Register. Queries the current contents in the Event Status Enable Register.
*ESR?	Event status register query	Queries & clears contents in the Standard Event Status Register.
*IDN?	Identification query	Returns identification string of the Oscilloscope.
*LRN?	Learn query	Returns a string that contains the current Oscilloscope setup.
*OPC *OPC?	Operation complete Operation complete query	Sets the Request for OPC flag when all pending operations have completed. Also sets OPC bit in the Event Status register. Returns a 1 to the output queue when all pending operations have completed.
*RCL <n>	Recall saved state	Recalls previously stored Oscilloscope Module configuration. <n> (1 to 4) is the location in memory where the desired (previously stored) set-up is located.
*RST	Reset	Resets the Oscilloscope hardware and firmware to a specific state (refer to Table 3-4 for a list of reset conditions).
*SAV <n>	Save state	Stores the present Oscilloscope Module configuration in memory. Stores all Oscilloscope and Measurement Set-up parameters. <n> (1 to 4) is the location in memory where the current set-up is to be stored.
*SRE <mask> *SRE?	Service request enable Service request enable query	Used to set the Service Request Enable Register bits to generate a service request. Queries the current contents in the Service Request Enable Register.
*STB?	Read status byte query	Queries the current contents in the Status Byte Register.
*TRG?	Trigger	Used to generate a trigger event.
*TST?	Self-Test query	Returns 0 unless self test fails. Disconnect all inputs prior to performing self tests.
*WAI	Wait to Continue	Halts execution of commands and queries until the No Operation Pending message is true.

— NOTES —

Command Quick Reference

The following tables summarize Standard Commands for Programmable Instruments (SCPI) commands and IEEE 488.2 Common (*) commands for the HP E1426A Oscilloscope module.

Subsystem	Command/Parameter	Description
ABORt		Stops acquiring data for the active waveform.
CALCulate	:DATA?	Returns the calculated results of the last math operation performed.
	:MATH<number>	Selects the math number (1-2).
	[:EXPRession] <*function*>	Adds, subtracts, or multiplies two specified sources and retains the result in the math number specified.
	:STATe <*mode*>	Enables or disables math operations for the selected math number.
	:STATe?	Returns the current state of the math number specified.
	:STATe <*mode*>	Enables or disables the calculate subsystem.
	:STATe?	Returns the current state of the math subsystem.
CALibration	:PCALibration	Probe calibration related commands.
	:ATTenuation	Probe attenuation calibration related commands.
	:BCALibration	Performs an attenuation calibration.
	:INPut<*number*>	Selects the input (1-4) for the attenuation calibration.
	:TNULl:INP1TO<*number*> ,<*time*>	Enters a time null value for a specified input pair (1-2, 1-3, 1-4).
	:REPort? <*input*>	Returns a report of calibration results for the input specified.
	:SCALibration	Self calibration related commands.
	:BCALibration	Performs a configured calibration, or loads default data.
	:DCALibration	Configures for a default calibration routine.
	:DELay <*input*>	Configures for a delay calibration on the input specified (1-4).
	:DOUTput <*level*>	Sets the DC Calibrator output to 0 volts or 5 volts.
	:LTCalibration	Configures for a logic trigger calibration on the input specified (1-4).
	:TNULl <*input_skew*>	Configures for a time null calibration on the input pair specified (1-2, 1-3, 1-4).
	:VERTical	Configures for a vertical calibration on all inputs.
	:TNULl <*value1*>,<*value2*>,<*value3*>	Enters time null values for input pairs 1-2, 1-3, and 1-4.
	:TNULl?	Returns the current time null values for input pairs 1-2, 1-3, and 1-4.
CONFigure?		Returns the last configured measurement.
CONFigure	[:SCALar]	
	:VOLTage	
	:AC [<*chan_list*>]	Configures the source specified by *chan_list* for an AC voltage measurement.
	:AMPLitude [<*chan_list*>]	Configures the source specified by *chan_list* for an amplitude voltage measurement.
	:[DC] [<*chan_list*>]	Configures the source specified by *chan_list* for a DC voltage measurement.
	:DCYCle [<*ref*>][<*chan_list*>]	Configures the source specified by *chan_list* for a duty cycle measurement. Reference level can be specified, or the default value can be used (if left blank).
	:FALL	Falling edge measurements.
	:OVERshoot [<*chan_list*>]	Configures the source specified by *chan_list* for an overshoot measurement on the falling edge.
	:PREShoot [<*chan_list*>]	Configures the source specified by *chan_list* for a preshoot measurement on the falling edge.

SCPI Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
CONFigure	[:SCALar] :VOLTage	
	:FALL	Falling edge measurements.
	:TIME [<*lower_limit*> [,<*upper_limit*>]] [<*chan_list*>]	Configures the source specified by *chan_list* for a fall time measurement. Upper and lower threshold limits can be specified, or the default values can be used (if left blank).
	:FREQuency [<*chan_list*>]	Configures the source specified by *chan_list* for a frequency measurement.
	:FTIMe [<*lower_limit*> [,<*upper_limit*>]] [<*chan_list*>]	Configures the source specified by *chan_list* for a fall time measurement. Upper and lower threshold limits can be specified, or the default values can be used (if left blank).
	:HIGH [<*chan_list*>]	Configures the source specified by *chan_list* for a high voltage measurement.
	:LOW [<chan_list>]	Configures the source specified by *chan_list* for a low voltage measurement.
	:MAXimum [<*chan_list*>]	Configures the source specified by *chan_list* for a maximum voltage measurement.
	:MINimum [<*chan_list*>]	Configures the source specified by *chan_list* for a minimum voltage measurement.
	:NDUTycycle [<*ref*>][<*chan_list*>]	Configures the source specified by *chan_list* for a negative duty cycle measurement. Reference level can be specified, or the default value can be used (if left blank).
	:NWIDth [<*ref*>][<*chan_list*>]	Configures the source specified by *chan_list* for a negative pulse width measurement. Reference level can be specified, or the default value can be used (if left blank).
	:PDUTycycle [<*ref*>][<*chan_list*>]	Configures the source specified by *chan_list* for a positive duty cycle measurement. Reference level can be specified, or the default value can be used (if left blank).
	:PERiod [<*chan_list*>]	Configures the source specified by *chan_list* for a period measurement.
	:PWIDth [<*ref*>][<*chan_list*>]	Configures the source specified by *chan_list* for a positive pulse width measurement. Reference level can be specified, or the default value can be used (if left blank).
	:RISE	Rising edge measurements.
	:OVERshoot [<*chan_list*>]	Configures the source specified by *chan_list* for an overshoot measurement on the rising edge.
	:PREShoot [<*chan_list*>]	Configures the source specified by *chan_list* for a preshoot measurement on the rising edge.
	:TIME [<*lower_limit*> [,<*upper_limit*>]] [<*chan_list*>]	Configures the source specified by *chan_list* for a rise time measurement. Upper and lower threshold limits can be specified, or the default values can be used (if left blank).
	:RTIMe [<*lower_limit*> [,<*upper_limit*>]] [<*chan_list*>]	Configures the source specified by *chan_list* for a rise time measurement. Upper and lower threshold limits can be specified, or the default values can be used (if left blank).
	:TMAXimum [<*chan_list*>]	Configures the source specified by *chan_list* for a time at first maximum voltage measurement.
	:TMINimum [<*chan_list*>]	Configures the source specified by *chan_list* for a time at first maximum voltage measurement.
FETCh	[[:SCALar] :VOLTage [:<*function*>]]?	Transfers the results of a previously executed measurement to the output buffer. When function is provided, transfers the results of the specified measurement. When function is blank, transfers the results of the last measurement executed.

SCPI Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
FORMat	[:DATA] *<type>*,*<format>*	Selects the waveform data format (8 or 16 bit).
	[:DATA]?	Returns the format currently selected.
INITiate	:CONTinuous *<mode>*	Enables or disables waveform data acquisition.
	:CONTinuous?	Returns the current data acquisition state.
	[:IMMediate]	Digitizes waveform data on the selected input (1-4), and starts a configured measurement.
MEASure	[:SCALar]	
	:VOLTage	
	:AC? [*<chan_list>*]	Performs an AC voltage measurement on the source specified by *chan_list*.
	:AMPLitude? [*<chan_list>*]	Performs an amplitude voltage measurement on the source specified by *chan_list*.
	:[DC]? [*<chan_list>*]	Performs a DC voltage measurement on the source specified by *chan_list*.
	:DCYCle? [*<ref>*][*<chan_list>*]	Performs a duty cycle measurement on the source specified by *chan_list*. Reference level can be specified, or the default value can be used (if left blank).
	:FALL	Falling edge measurement on the source specified by *chan_list*.
	:OVERshoot? [*<chan_list>*]	Performs an overshoot measurement on the source specified by *chan_list*.
	:PREShoot? [*<chan_list>*]	Performs a preshoot measurement on the source specified by *chan_list*.
	:TIME? [*<lower_limit>* [,*<upper_limit>*]] [*<chan_list>*]	Performs a fall time measurement on the source specified by *chan_list*. Upper and lower threshold limits can be specified, or the default values can be used (if left blank).
	:FREQuency? [*<chan_list>*]	Performs a frequency measurement on the source specified by *chan_list*.
	:FTIMe? [*<lower_limit>* [,*<upper_limit>*]] [*<chan_list>*]	Performs a fall time measurement on the source specified by *chan_list*. Upper and lower threshold limits can be specified, or the default values can be used (if left blank).
	:HIGH? [*<chan_list>*]	Performs a high voltage measurement on the source specified by *chan_list*.
	:LOW? [<chan_list>]	Performs a low voltage measurement on the source specified by *chan_list*.
	:MAXimum? [*<chan_list>*]	Performs a maximum voltage measurement on the source specified by *chan_list*.
	:MINimum? [*<chan_list>*]	Performs a minimum voltage measurement on the source specified by *chan_list*.
	:NDUTycycle? [*<ref>*][*<chan_list>*]	Performs a negative duty cycle measurement on the source specified by *chan_list*. Reference level can be specified, or the default value can be used (if left blank).
	:NWIDth? [*<ref>*][*<chan_list>*]	Performs a negative pulse width measurement on the source specified by *chan_list*. Reference level can be specified, or the default value can be used (if left blank).
	:PDUTycycle? [*<ref>*][*<chan_list>*]	Performs a positive duty cycle measurement on the source specified by *chan_list*. Reference level can be specified, or the default value can be used (if left blank).
	:PERiod? [*<chan_list>*]	Performs a period measurement on the source specified by *chan_list*.
	:PWIDth? [*<ref>*][*<chan_list>*]	Performs a positive pulse width measurement on the source specified by *chan_list*. Reference level can be specified, or the default value can be used (if left blank).

SCPI Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
MEASure	[:SCALar]	
	:VOLTage	
	:RISE	Rising edge measurement on the source specified by *chan_list.*
	:OVERshoot? [<*chan_list*>]	Performs an overshoot measurement on the source specified by *chan_list.*
	:PREShoot? [<*chan_list*>]	Performs a preshoot measurement on the source specified by *chan_list.*
	:TIME? [<*lower_limit*> [,<*upper_limit*>]] [<*chan_list*>]	Performs a rise time measurement on the source specified by *chan_list.* Upper and lower threshold limits can be specified, or the default values can be used (if left blank).
	:RTIMe? [<*lower_limit*> [,<*upper_limit*>]] [<*chan_list*>]	Performs a rise time measurement on the source specified by *chan_list.* Upper and lower threshold limits can be specified, or the default values can be used (if left blank).
	:TMAXimum? [<*chan_list*>]	Performs a time at first maximum voltage measurement on the source specified by *chan_list.*
	:TMINimum? [<*chan_list*>]	Performs a time at first maximum voltage measurement on the source specified by *chan_list.*
MEMory	:VME	VME Memory allocation for acquisition and measurement data.
	:ADDRess <*address*>	Selects an A24 memory address space for acquisition data.
	:ADDRess? [MIN|MAX]	Returns the present A24 memory address space selected if the parameter is blank. Returns MINimum, or MAXimum address available, if selected.
	:MEASure	Commands for measurement data.
	:ADDRess <*address*>	Selects an A24 memory address space for measurement data.
	:ADDRess? [MIN|MAX]	Returns the present A24 memory address space selected if the parameter is blank. Returns MINimum, or MAXimum address available, if selected.
	:STATe <*mode*>	Enables or disables VME memory space for measurement data.
	:STATe?	Returns the current VME memory measurement state.
	:SIZE <*bytes*>	Selects the size of the external VME memory card.
	:SIZE?	Returns the current size of the external VME memory card.
	:STATe <*mode*>	Enables or disables the VME memory subsystem (for acquisition and measurement data).
	:STATe?	Returns the current VME memory subsystem state.
OUTPut	:ECLTrg<*number*>	Selects ECL trigger bus lines 0-2.
	[:STATe] <*mode*>	Enables or disables the selected ECL trigger line.
	[:STATe]?	Returns the current ECL trigger state for the line selected.
	:EXTernal	Selects the TTL Trigger Output connector on the front panel.
	[:STATe] <*mode*>	Enables or disables the TTL Trigger Output connector.
	[:STATe]?	Returns the current TTL Trigger Output connector state.
	[:STATe]	Enables or disables the entire output subsystem. Must be enabled for any selected output to function.
	[:STATe]?	Returns the current output subsystem state.
	:TTLTrg<*number*>	Selects TTL trigger bus lines 0-7.
	[:STATe] <*mode*>	Enables or disables the selected TTL trigger line.
	[:STATe]?	Returns the current TTL state for the line selected.
READ	[[:SCALar]	
	:VOLTage	
	[:<*function*>]]?	Initiates a previously configured measurement and transfers the results to the output buffer. When function is provided, the specified measurement is performed. When function is blank, the last measurement executed is performed.

SCPI Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
[SENSe:]	AVERage	Average subsystem.
	:COUNt <count>	Enters the count used during average data acquisition mode.
	:COUNt?	Returns the current acquisition count value.
	[:STATe] <mode>	Enables or disables the average acquisition mode. When OFF, acquisition mode is determined by the AVERage:TYPE selected.
	[:STATe]?	Returns the current average acquisition mode state.
	:TYPE <mode>	Selects the type of acquisition that will occur (scalar or envelope) when AVERage[:STATe] is set to OFF.
	:TYPE?	Returns the acquisition type (scalar or envelope) currently selected.
	CORRection<number>	Correction subsystem.
	:AFACtor <atten>	Enters the specified input's (1-4) probe attenuation factor.
	:AFACtor?	Returns the current probe attenuation factor for the specified input.
	INPut<number>	Input subsystem.
	:COUPling <type>	Selects the specified input's (1-4) coupling (AC or DC).
	:COUPling?	Returns the specified input's (1-4) current coupling selection.
	:FILTer	Filter selections.
	[:LPASs][:STATe] <mode>	Enables or disables the specified input's (1-4) low pass filter.
	[:LPASs][:STATe]?	Returns the specified input's (1-4) current low pass filter state.
	:HPASs[:STATe] <mode>	Enables or disables the specified input's (1-4) high pass filter.
	:HPASs[:STATe]?	Returns the specified input's (1-4) current high pass filter state.
	:IMPedance <value>	Selects the specified input's (1-4) impedance (1MΩ or 50Ω).
	:IMPedance?	Returns the specified input's (1-4) current impedance selection.
	[:STATe] <mode>	Enables or disables the specified input (1-4).
	[:STATe]?	Returns the specified input's (1-4) current state.
	SWEep	Sweep subsystem.
	:POINts <points>	Selects the number of time buckets for a data acquisition.
	:POINts?	Returns the current points value.
	:COMPlete <complete>	Enters the completion criteria for a data acquisition.
	:COMPlete?	Returns the current acquisition complete value.
	:TIME:	Time selections.
	:CENTer <center_time>	Enters the time representing the center of the selected horizontal range.
	:CENTer?	Returns the current center sweep time value.
	:DELay <time>	Enters the desired time between the trigger and delay reference point.
	:DELay?	Returns current sweep delay value.
	:LINK <reference>	Sets the delay reference point to start, stop, or center of the active waveform.
	:LINK?	Returns the delay reference point currently selected.
	:RANGe <range>	Enters full scale horizontal range.
	:RANGe?	Returns current full scale horizontal range value.
	:SPAN <span>	Enters full scale horizontal span (range).
	:SPAN?	Returns current full scale horizontal span (range) value.
	:STARt <start_time>	Enters the time representing the start of the selected horizontal range.
	:STARt?	Returns the current start sweep time value.
	:STOP <stop_time>	Enters the time representing the stop of the selected horizontal range.
	:STOP?	Returns the current stop sweep time value.

SCPI Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
[SENSe:]	VOLTage*<number>*	Voltage subsystem.
	:RANGe	Range selections.
	:LOWer *<lower>*	Enters the voltage representing the bottom of the selected vertical range.
	:LOWer?	Returns the current lower range value.
	:OFFSet *<value>*	Enters the specified input's (1-4) offset.
	:OFFSet?	Returns the specified input's (1-4) current offset value.
	[:PTPeak] *<range>*	Enters the specified input's (1-4) full scale vertical range.
	[:PTPeak]?	Returns the specified input's (1-4) current full scale vertical range value.
	:UPPer *<upper>*	Enters the voltage representing the top of the selected vertical range.
	:UPPer?	Returns the current upper range value.
STATus	:OPERation	Reports when a trigger has occurred.
	:CONDition?	Always returns (0).
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:PRESet	Sets all the questionable enable registers to "1's".
	:QUEStionable	Reports the calibration and self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:CALibration	Reports the input 1-4, default, and probe attenuation calibration values and conditions.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:INPut*<number>*	Reports the specified input's (1-4) current calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:AD	Reports the specified input's (1-4) A/D calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:DELay	Reports the specified input's (1-4) delay calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.

SCPI Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
STATus	:QUEStionable	
	:CALibration	
	:INPut<*number*>	
	:GAIN	Reports the specified input's (1-4) gain calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:HYSTeresis	Reports the specified input's (1-4) hysteresis calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:LTRigger	Reports the input 1 logic trigger calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:OFFSet	Reports the specified input's (1-4) offset calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:TNULl	Reports the specified input's (2-4) time null calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:TRIGger	Reports the specified input's (1-4) trigger calibration data.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:DCALibration	Reports default calibration load status.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:PROBe	Reports the probe calibration attenuation results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.

SCPI Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
STATus	:QUEStionable	
	:TEST	Reports diagnostic or self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:ACQuisition	Reports the acquisition self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:AD	Reports the acquisition A/D self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:ATRigger	Reports the acquisition analog trigger self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:DA	Reports the acquisition D/A self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:LTRigger	Reports the acquisition logic trigger self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:TIMebase	Reports the acquisition time base self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:INTerpolator	Reports the acquisition time base interpolator self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.

SCPI Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
STATus	:QUEStionable	
	:TEST	
	:RAM	Reports the random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:ACQuisition	Reports the acquisition random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:DISPlay	Reports the display random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:NVOLatile	Reports the non-volatile random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:SYSTem	Reports the system random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:ROM	Reports the read only memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:NPRotect	Reports the non-protected random access memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.
	:SYSTem	Reports the system read only memory self test results.
	:CONDition?	Always returns 0.
	:ENABle	Allows true conditions (transitions) in the event register to be reported.
	[:EVENt]?	Returns a decimal weighted value indicating which bits have been set.

SCPI Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
SYSTem	:AUToscale	Evaluates all inputs, then sets conditions to present the signals.
	:ERRor?	Returns system error number and message.
	:LANGuage *<command>*	Selects the Oscilloscope programming language (COMP or SCPI).
	:LANGuage?	Returns the programming language currently selected.
	:SERial *<string>*	Enters the instrument serial number.
	:SET *<setup>*	Sends a previously saved learn string to the Oscilloscope (contains Oscilloscope setup information).
	:SET?	Returns the learn string (contains Oscilloscope setup information).
	:VERSion?	
TEST	:ACQ [*<test>*]	Performs an internal acquisition self test. All five tests (AD, analog trigger, logic trigger, time base, and DA), are performed unless an individual test is specified.
	:RAM [*<test>*]	Performs an internal random access memory self test. All four tests (display, acquisition, system, and non-volatile) are performed unless an individual test is specified.
	:ROM [*<test>*]	Performs an internal read only memory self test. Both tests (system ROM and non-volatile protected RAM) are performed unless an individual test is specified.
	:TALL	Performs all internal self tests (ACQ, RAM, and ROM).
TRACe:	[:DATA] *<destination>,<source>*	Writes a binary block of data to the specified destination, or moves the specified source to the selected destination.
	[:DATA]? *<source>*	Reads a binary block of data from the specified source.
	:POINts? *<source>*	Returns the data points value currently selected in the preamble of the specified source.
	:PREamble *<destination>,<data>*	Sends preamble data to the specified destination.
	:PREamble? *<source>*	Receives preamble data from the specified source.
TRIGger	:ECOunt *<events>*	Selects the number of events to holdoff the trigger event.
	:ECOunt?	Returns the current holdoff value.
	:HYSTeresis *<mode>*	Enables or disables noise rejection for the currently selected source.
	:HYSTeresis?	Returns the current noise rejection state.
	:LEVel *<level>*	Enters the trigger level.
	:LEVel?	Returns the current trigger level value.
	:SLOPe *<polarity>*	Selects the edge (rising or falling) for the trigger.
	:SLOPe?	Returns the trigger edge currently selected.
	:SOURce *<source>*	Selects the source that will produce the trigger (input 1-4, TTLTrg 0-7, ECLTrg 0-2).
	:SOURce?	Returns the trigger source currently selected.

IEEE 488.2 Common Commands Quick Reference

Command	Title	Description
*CLS	Clear status register	Clears all Event Registers, the Request for OPC flag, and all Queues (except output queue).
*ESE <mask> *ESE?	Event status enable Event status enable query	Used to set the bits in the Event Status Enable Register. Queries the current contents in the Event Status Enable Register.
*ESR?	Event status register query	Queries and clears contents in the Standard Event Status Register.
*IDN?	Identification query	Returns identification string of the Oscilloscope.
*LRN?	Learn query	Returns a string that contains the current Oscilloscope setup.
*OPC *OPC?	Operation complete Operation complete query	Sets the Request for OPC flag when all pending operations have completed. Also sets OPC bit in the Event Status register. Returns a 1 to the output queue when all pending operations have been completed.
*RCL <n>	Recall saved state	Recalls previously stored Oscilloscope Module configuration. <n> (1 to 4) is the location in memory where the desired (previously stored) set-up is located.
*RST	Reset	Resets the Oscilloscope hardware and firmware to a specific state (refer to Table 3-4 for a list of reset conditions).
*SAV <n>	Save state	Stores the present Oscilloscope Module configuration in memory. Stores all Oscilloscope and Measurement Set-up parameters. <n> (1 to 4) is the location in memory where the current set-up is to be stored.
*SRE <mask> *SRE?	Service request enable Service request enable query	Used to set the Service Request Enable Register bits to generate a service request. Queries the current contents in the Service Request Enable Register.
*STB?	Read status byte query	Queries the current contents in the Status Byte Register.
*TRG?	Trigger	Used to generate a trigger event.
*TST?	Self-Test query	Returns 0 unless self test fails. Disconnect all inputs prior to performing self tests.
*WAI	Wait to Continue	Halts execution of commands and queries until the No Operation Pending message is true.

A

Oscilloscope Specifications and Characteristics

Specifications

The following are performance specifications for the HP E1426A Digitizing Oscilloscope.

Vertical

Bandwidth (–3 dB, dc coupled)
Repetitive:[1] dc to 500 MHz
Single-shot: dc to 2 MHz (based on 10 points per period of input signal)

Rise Time:[2] 700 ps

Input R (selectable): 1 MΩ ±1% or 50Ω ±1%

Maximum Input Voltage[3]
1MΩ: ±250 V [dc + peak ac(<10 kHz)]
50Ω: 5 Vrms

Offset Accuracy:[4] ±(0.5% of ch. offset + 2% of voltage range)

Voltage Measurement Accuracy (dc)[4,5]
Dual Cursor: ±(1.25% of voltage range)
Single Cursor: ±(1.25% of voltage range + offset accuracy)

Horizontal

Time Base Reference Accuracy: 0.005%

Delta-t Accuracy
Real-time: ±(.2% x time base range + 0.005% x delta-t + 150 ps)

Trigger

Trigger Sensitivity

≥40 mV Voltage Range
dc to 100 MHz: 0.063 x voltage range
100 MHz to 500 MHz: 0.156 x voltage range

<40 mV Voltage Range
dc to 100 MHz: 2.5 mV
100 MHz to 500 MHz: 6 mV

Notes: Specifications valid for temperature range ±10°C from software calibration temperature with eight or more averages selected.

[1] Upper bandwidth reduces by 2.5 MHz for each °C above 35°C.

[2] Rise time figure is calculated from: tr = 0.35/Bandwidth.

[3] On voltage ranges ≤400 mV the maximum overdrive of the input must not exceed 125 times the voltage range.

[4] Expansion is used below 56 mV voltage range so vertical resolution and accuracies are correspondingly reduced.

[5] Accuracy decreases 0.08% per °C from software calibration temperature.

Characteristics

The following are performance characteristics of the HP E1426A Digitizing Oscilloscope.

Vertical

Switchable Bandwidth Limits
ac-coupled (lower -3 dB frequency): 90 Hz
LF reject (lower -3 dB frequency): 450 Hz
bandwidth limit (upper -3 dB frequency): dc to 30 MHz

Number of channels:[1] 4

Vertical Sensitivity Voltage Range (all channels): 8 mV to 40 V

Vertical Gain Accuracy (dc):[2,3] ±1.25%

Vertical Resolution:[3] ±0.4% (8-bit A/D)
±0.1% (10 bits via digitize with averaging)

Maximum Sample Rate: 20 MSa/s

Waveform Record Length[4,5]**:** Up to 1024 points

Input C: 7 pF nominal

Input coupling: ac, dc

Offset Range:	Voltage Range	Available Offset
	8 mV-400 mV	±2 V
	>400 mV-2 V	±10 V
	>2 V-10 V	±50 V
	>10 V-40 V	±250 V

Dynamic range (dc + peak ac): ±1.5 x voltage range from offset

Channel-to-channel Isolation: (with channels at equal sensitivity)
40 dB: dc to 100 MHz
30 dB: 100 to 500 MHz

Horizontal

Time Base Range: 2 ns to 50 s

Time Base Resolution: 20 ps

	Time Base Range	Available Delay
Delay Range	500 ms—50 s	4 x time base range
(post-trigger)	1 ms—200 ms	1 s
	2 ns—500 µs	1,000 x time base range
Delay Range	50µs—50 s	-3.996 x time base range
(pre-trigger)	100 ns—20µs	-99.9 µs
	2 ns—50 ns	-1,000 x time base range

Trigger

Trigger Pulse Width (minimum): 1.5 ns

Trigger Level Range: ±1.5 x voltage range from offset.

Backplane Trigger Delay: ≈40 ns (from oscilloscope input to backplane).

Notes: Specifications valid for temperature range ±10°C from software calibration temperature with eight or more averages selected.

1. Simultaneous acquisition on two channels. Channels 1 and 4 are acquired simultaneously. If four channels are used, data is acquired alternately by channels 1 and 4, then 2 and 3.
2. Accuracy decreases 0.08% per °C from software calibration temperature.
3. Expansion is used below 56 mV range so vertical resolution and accuracies are correspondingly reduced.

4. In repetitive mode:
   2 ns time base range, waveform record length is 100 points
   5 ns time base range, waveform record length is 250 points
   10 ns time base range, waveform record length is 500 points
   ≥20 ns time base range, waveform record length is 1000 points
5. For single shot via digitize, the waveform record length is dependent on the timebase range. Note: You may need to set ACQUIRE:COMPLETE to a value less than 100 (for 100%).

SCREEN WIDTH	SINGLE-SHOT POINTS/ACQUISITIONS
50s to 50µs	500
20µs	≈400
10µs	≈200
5µs	≈100
2µs	≈40
1µs	≈20
500ns	≈10
200ns	≈4
100ns	≈1
50ns to 2ns	0 - single-shot not available

A-2.2 Specifications

Operating Characteristics

Vertical

Deflection Factors: All channel's deflection factors are adjustable from 8 mV to 40 V.

Probe Attenuation Factors: Values from 0.9 to 1000 may be entered to scale the oscilloscope for external probes or attenuators attached to the channel inputs. When probe tip calibration is done, this value is calculated automatically.

Input Impedance: 1MΩ or 50 Ω selectable for each input.

Bandwidth limit (HF Reject): Can be selected for each input individually. Provides low pass filter with a –3 dB point at approximately 30 MHz for both triggering and signal acquisition.

LF Reject: Can be selected for each input individually. Provides high pass filter with a –3 dB point at approximately 450 Hz for both triggering and signal acquisition.

AC Coupling: Can be selected for each input individually. Provides high pass filter with a –3 dB point at approximately 90 Hz and a two-pole roll-off for both triggering and signal acquisition.

ECL/TTL Presets: Vertical deflection factor, offset, and trigger level can be preset independently on each channel for ECL and TTL levels.

Note

ECL and TTL presets are only available when using the COMPatible programming language.

Horizontal

Dual Time Base Windowing: Allows user to zoom in on portions of the waveform. The window time base can be set to provide as much as a 20:1 expansion ratio. Waveform measurements are performed on the dual time base window information when windowing is turned on.

Note

Dual Time Base Windowing is only available when using the COMPatible programming language.

Delay Between Channels: Difference in delay between channels can be nulled out to compensate for differences in input cables or probe length. See the Time Null (TNULl) Probe Calibration command in chapters 4 or 6 for more information.

Reference Location: The reference point can be located at the left edge, center, or right of the active waveform. The reference point is equal to trigger point plus the delay time.

Trigger Modes

Edge Trigger: Positive or negative edge can be selected for trigger on any of the four channel inputs.

Note

The remaining trigger modes are only available when using the COMPatible programming language.

Pattern Trigger: A pattern can be specified using all four of the inputs. Each of the inputs can be specified as a high, low, or don't care with respect to the level setting in the edge trigger mode. Trigger can be selected to occur on the last edge to enter the specified pattern or the first edge to exit the specified pattern.

Time Qualified Pattern Trigger: A trigger will occur on the first edge to exit a pattern only if it meets the specified time criteria. The available time qualified modes are:

- pattern present < [time]
- pattern present > [time]
- range–pattern present > [time1] and <[time2]

The time settings are adjustable from 20 ns to 160 ms (±3% ±2 ns). The time filter recovery time is ≤12 ns. In the "pattern present < [time]" mode, the pattern must be present > 1.5 ns for the trigger to respond.

Glitch Trigger: Use "pattern present < [time]" with [time] selected such that it is just less than the pulse width of the signal you are analyzing. The minimum glitch width is 1.5 ns.

State Trigger: A pattern is specified on any three of the four inputs, with the fourth input used as a clock. A trigger will occur on the rising or falling edge of the input specified as the clock when the pattern is present or not present. Setup time for the pattern with respect to the clock is less than or equal to 10 ns; hold time is zero.

Delayed Trigger

Event-delayed Mode: The trigger can be qualified by an edge, pattern, time qualified pattern or state. The delay can be specified as a number of occurrences of a rising or falling edge on any one of the four inputs. After the delay, an occurrence of a rising or falling edge of any one of the four inputs will generate the trigger. The occurrence value of the edge to trigger on is selectable from 1 to 16,000,000. The maximum edge counting rate is 70 MHz.

TV Trigger

60 Hz/525 Lines: Source is selected to be any one of the four inputs. Trigger level is adjustable for the selected source. Polarity is selected for positive or negative synchronizing pulses. A trigger occurs on the selected line and field of a 2/1 interlaced composite video signal. Line numbering is 1 to 263 for field 1 and 1 to 262 for field 2. This TV trigger mode is compatible with broadcast standard M.

50 Hz/625 Lines: Source is selected to be any one of the four inputs. Trigger level is adjustable for the selected source. Polarity is selected for positive or negative synchronizing pulses. A trigger occurs on the selected line and field of a 2/1 interlaced composite video signal. Line numbering is 1 to 313 for field 1 and 314 to 625 for field 2. This TV trigger mode is compatible with broadcast standards B, C, D, G, H, I, K, K1, L, and N.

User-defined Mode: Source is selected to be any one of the four inputs. Trigger level is adjustable for the selected source. The trigger is qualified with a high or low pulse that meets a selectable time range. The trigger is an occurrence of a rising or falling edge of the source after the qualifying pulse. The time settings for the qualifier are selectable from 20 ns to 160 ms (±3% ±2 ns). The trigger occurrence value is selectable from 1 to 16,000,000.

Note

All TV trigger modes require a clamped video signal for stable triggering. Use the HP 1133A TV pod to provide clamped video output that can be used in conjunction with the HP E1426A's TV triggering capabilities.

Trigger Holdoff: Trigger can be held off by either time or events over the ranges:

- Time: 40 ns to 320 ms
- Events: 2 to 16,000,000

An Event is defined as the specified trigger condition. A separate holdoff setting (time or events) is available for each trigger mode except delayed trigger, which is set to 40 ns.

Note

Holdoff by time is only available when using the COMPatible programming language.

Noise Reject Trigger: Improves triggering on noisy signals by increasing hysteresis.

Acquisition Modes

Minimum Persistence: The waveform is updated as new data is acquired for a particular time bucket, with one waveform data value per time bucket.

Note

The term "Time Bucket" is defined as — the time range divided into a specific number of horizontal time points (as defined by the :POINts command), each with an equal and fixed time associated with it.

Infinite Persistence: Each data point (existing and new) is always retained (never replaced).

Note

Persistence selections are only available when using the COMPatible programming language.

Averaging: The number of averages (n) can be specified in powers of 2, up to 2048. On each acquisition, 1/n times the new data is added to (n–1)/n of the previous value at each time coordinate. Averaging operates continuously, except for the digitize command, for which averaging terminates at the specified number of averages.

Envelope: Provides the running maximum and minimum voltage levels at each time bucket for a repetitive waveform.

Scroll Mode: In the Auto Mode at settings from 2 s to 50 s the E1426A automatically selects the scroll mode. It will also select scroll mode in the triggered mode with delay reference (= left and delay ≥0). Scroll mode updates each data point on the waveform as the data is acquired.

Delta t/Delta V

Markers: Dual voltage markers and dual time markers are available. Voltage markers can be independently assigned to channels, memories, or functions.

Note

Time and voltage markers are only available when using the COMPatible programming language.

Waveform Math

Two independent functions are provided for waveform math. The operators are +,–, X, invert, and only. The vertical channels and the waveform memories can be used as operands for the waveform math. Sensitivity and offset for these functions can be adjusted independently.

Note

Invert and only functions are only available when using the COMPatible programming language.

Waveform Save

Four non-volatile waveform memories and two volatile pixel memories are provided. Waveform memories store single-valued waveforms, such as an averaged waveform. If an envelope waveform is stored to a waveform memory, it will automatically be stored with the upper waveform in one waveform memory and the lower waveform in another. Pixel memories store an entire active waveform. They are very useful for storing multiple overlapping waveforms and infinite persistence waveforms.

Automatic Pulse Parameter Measurements

The E1426A offers 19 automatic pulse parameter measurements. The standard measurements are performed with 10%, 50%, and 90% voltage thresholds, as defined by IEEE standard 194-1977, "IEEE Standard Pulse Terms and Definitions."

Automatic measurements available on the HP E1426A

Rise Time	+ Pulse Width	Volts amp	Volts avg	Preshoot
Fall Time	– Pulse width	Volts base (low)	Volts max	Volts min
Frequency	Duty Cycle	Volts top (high)	Overshoot	Volts ACrms
Period	Delay	Volts p-p	Volts DCrms	

Note

Delay, peak-to-peak, and DCrms voltage measurements are only available when using the COMPatible programming language.

User-Definable Measurement Thresholds

The E1426A allows you to set your own thresholds for automatic measurements. Both the upper and lower thresholds can be set, either in % or Volts, as long as the upper threshold value is always ≥ the lower threshold. The mid threshold is always equal to mid-value between the upper and lower threshold.

Note

Only upper and lower thresholds for rise time and fall time can be defined by the user when using the SCPI programming language.

Continuous Measurements: Allows automatic measurements to be continuously updated. With continuous measurements off, the voltage and time markers are set on the waveform to indicate the position of the last measurement.

Measurement Statistics: The maximum, minimum and average of continuously updated measurements are calculated. Any three measurements can be selected for simultaneous calculation.

Measurement Limit Test: Maximum and minimum limits can be set for three of the automatic measurements. These continuously updated measurements are compared to the maximum and minimum limits. If the measurements are found to be outside the defined limits, the acquisition can be stopped and the waveform can be stored to a memory. In addition, the Service Request line can be set, to flag the controller. Measurement limit test can be set to stop after test limits have exceeded, or to continue testing.

Note

Continuous measurements, measurement statistics, and limit testing are only available when using the COMPatible programming language.

Setup Aids

Auto-Scale: The Auto-Scale feature automatically adjusts the vertical and horizontal factors, and the trigger level to values appropriate to the signals applied to the inputs. The Auto-Scale feature requires a signal with a duty cycle greater than 0.5% and a frequency greater than 50 Hz. Auto-Scale is operative only for relatively stable input signals.

Save/Recall: Four setups (1-4) may be saved in non-volatile memory.

Recall 0: If Auto-Scale, ECL or TTL preset, or recall setup are inadvertently selected, recall 0 restores the instrument to its last state prior to selection.

Front Panel Outputs

TTL Trigger Output: BNC female connector provides a TTL level output signal. The trailing (falling) edge should be used as the edge synchronous with the acquisition trigger. The trigger output uses the standard buffering scheme recommended in the VXI System specification.

Probe Compensation, AC Calibrator Output: BNC female connector provides a signal (≈1.5 kHz) for probe compensation. A probe-to-BNC adapter is used to connect the probe to the Front panel Probe Compensation BNC output. During calibration, this output is used to provide other calibration signals. This same BNC connector is used for trigger output. The leading (rising) edge, with amplitude from approximately –400 mV to 0 V (when a terminated in 50 Ω), is synchronous with the system trigger. The trailing (falling) edge of this pulse occurs approximately at the end of holdoff. The leading (rising) edge should be used as the edge synchronous with trigger.

Note

Selection of a trigger output from this connector is only available when using the COMPatible programming language.

DC Calibrator Output: BNC female connector provides the output used for vertical calibration of the E1426A.

Self Test Calibration

Built-in Self Test and Calibration Routines: Internal self-test capabilities provide a 90% confidence that the instrument is operating properly. External test procedures in the service manual provide 100% confidence. Self-calibration routines, ensure that the instrument is operating with its greatest accuracy and requires no external test equipment.

General Characteristics

Environmental Conditions

Temperature

Operating: 0°C to + 55°C
Non-operating: –40°C to + 70°C

Humidity

Operating: up to 95% relative humidity (non-condensing) at +40°C
Non-operating: up to 90% relative humidity at + 65°C.

Altitude

Operating: up to 4,600 meters (15,000 ft)
Non-operating: up to 15,300 meters (50,000 ft).

Vibration

Operating: Random vibration 5-500 Hz, 10 minutes per axis, 0.3 Grms.
Non-operating: Random vibration 5-500 Hz, 10 minute per axis, 2.41 Grms; Resonant search 5 to 500 Hz swept sine, 1 Octave/minute sweep rate, (0.75 G), 5 minute resonant dwell @ 4 resonances per axis.

Weight

Net: approximately 4 kg (9 lb.)

Shipping: approximately 8 kg (18 lb.)

Oscilloscope Error Messages

Table B-1 lists the error messages associated with the Oscilloscope module programmed using Hewlett-Packard 54503A Compatible Language (COMP). Table B-2 lists the error messages associated with the Oscilloscope module programmed using Standard Commands for Programmable Instruments (SCPI). See the appropriate mainframe manual for a complete list of error messages.

Table B-1. COMP Error Messages

No.	Description
0	No error
-100	Command error (unknown command)
-101	Invalid character received
-110	Command header error
-111	Header delimiter error
-120	Numeric argument error
-121	Wrong data type (numeric expected)
-123	Numeric overflow
-129	Missing numeric argument
-130	Non-numeric argument error
-131	Wrong data type (char expected)
-132	Wrong data type (string expected)
-133	Wrong data type (block expected)
-134	Data Overflow string or block to long
-139	Missing non-numeric argument
-142	Too many arguments
-143	Argument delimiter error
-144	Invalid message unit delimiter

Table B-1. COMP Error Messages — Continued

No.	Description
-200	No Can Do (generic execute error)
-201	Not executable in local mode
-202	Settings lost due to remote to local or power on
-203	Trigger ignored
-211	Legal command, but settings conflict
-212	Argument out of range
-221	Busy doing something else
-222	Insufficient capability or configuration
-232	Output buffer full or overflow
-300	Device failure
-301	Interrupt fault
-302	System error
-303	Time out
-310	RAM error
-311	RAM failure (hard error)
-312	RAM data loss (soft error)
-313	Calibration data loss
-320	ROM error
-321	ROM checksum
-322	Hardware and firmware incompatible
-330	Power on test failed
-340	Self test failed
-350	Too Many Errors (error queue overflow)
-400	Query Error (generic)
-410	Query INTERRUPTED
-420	Query UNTERMINATED
-421	Query received, Indefinite block response in progress
-422	Addressed to Talk, Nothing to Say
-430	Query DEADLOCKED

Table B-2. SCPI Error Messages

No.	Title	Potential Cause(s)
0	No error	No error has occurred.
70	Ram write protected	Calibration protect switch to PROTECTED.
−100	Command error	An unknown command error has occurred.
−101	Invalid character	Unrecognized character in specified parameter.
−102	Syntax error	Command missing a space or comma between parameters.
−103	Invalid separator	Command parameter is separated by a space rather than a comma.
−104	Data type error	The wrong data type (i.e., number, character, string, expression) was used when specifying a parameter.
−105	GET not allowed	A group execute trigger was received within a program message.
−108	Parameter not allowed	Parameter specified in a command that has no parameters.
−109	Missing parameter	No parameter specified in the command which has parameters.
−112	Program mnemonic too long	The command header contains more than 12 characters.
−113	Undefined header	Command header was incorrectly specified.
−121	Invalid character in number	An invalid character was used when specifying a parameter (i.e., alpha in decimal numeric).
−123	Numeric overflow	A parameter specifies a value greater than the command allows.
−124	Too many digits	More than 256 digits were specified for a parameter.
−128	Numeric data not allowed	Number specified for a parameter when a letter is required.
−130	Suffix error	An unknown suffix error has occurred.
−131	Invalid suffix	Parameter suffix incorrectly specified (i.e., K rather than KOHM).
−138	Suffix not allowed	Parameter suffix is specified when one is not allowed.
−140	Character data error	An unknown character data error has occurred.
−141	Invalid character data	The parameter type specified is not allowed.
−144	Character data too long	More than 12 characters were specified for a parameter.
−148	Character data not allowed	Character specified for a parameter when a number is required.
−150	String data error	An unknown string data error has occurred.
−151	Invalid string data	String data received was invalid.
−158	String data not allowed	String data encountered for a parameter when not expected.

Table B-2. SCPI Error Messages

No.	Title	Potential Cause(s)
–160	Block data error	An unknown block data error has occurred.
–161	Invalid block data	Block data received was invalid.
–168	Block data not allowed	Block data encountered for a parameter when not expected.
–170	Expression error	An unknown expression error has occurred.
–171	Invalid expression	The expression specified is not allowed.
–178	Expression data not allowed	A parameter other than a channel_list is enclosed in parentheses.
–200	Execution error	An unknown execution error has occurred.
–211	Trigger ignored	Trigger occurred from a source other than the specified source.
–213	INITiate ignored	An INITiate command was received while the oscilloscope was digitizing waveform data.
–221	Settings conflict	Parameters are set such that a measurement cannot be made.
–222	Data out or range	The parameter value specified is too large or too small.
–223	Too much data	The received block, string, or expression contained more data than was expected.
–310	System error	An unknown system error has occurred.
–350	Too many errors	The error queue is full as more than 30 errors have occurred.
–400	Query error	An unknown query error has occurred.
–410	Query interrupted	Data is not read from the output buffer before another command is executed.
–420	Query unterminated	Command which generates data not able to finish executing due to a configuration error.
–430	Query deadlocked	Command execution cannot continue since the mainframe's command input.
–440	Query unterminated after indefinite response	Command which generates data not able to execute due to a previous query error.

C

Optimizing Measurements

Using This Appendix

This appendix shows how to increase measurement speed using the HP E1426A Oscilloscope Module. Operation, measuring capabilities, and typical measurement times are all discussed to assist the user in performing measurements. This appendix contains the following sections:

Introduction

VXIbus is a relatively new systems architecture which provides many benefits including:

- Standardization
- Size Reduction
- Flexibility
- High Speed Measurement Throughput

While the standardization, size reduction, and flexibility benefits provided by the VXIbus are fairly obvious, measurement throughput is somewhat more complicated.

In order to obtain the maximum measurement throughput from any instrument, the user should have a basic understanding of its operation. This is especially true for the HP E1426A Digitizing Oscilloscope module. A good understanding of the how the oscilloscope operates, coupled with using the proper measurement setups, can increase throughput by nearly **10 times**.

Automatic Measurements

There are two basic steps required when performing an automatic measurement.

- Acquiring or digitizing the waveform on which the measurement will be performed.
- Measuring the parameter(s) using the digitized waveform.

Note

Once the waveform is digitized, perform as many measurements as possible on that one waveform. Do not digitize the waveform again for another measurement.

Sampling Techniques

All digitizing oscilloscopes digitize waveforms, however the method in which the waveform voltage is sampled can be divided into three basic categories:

- Real-Time
- Sequential
- Random Repetitive

The following discussion explains the basic operation of each sampling technique. Refer to Chapter 1, Figure 1-1 as required for a block diagram of the HP E1426A.

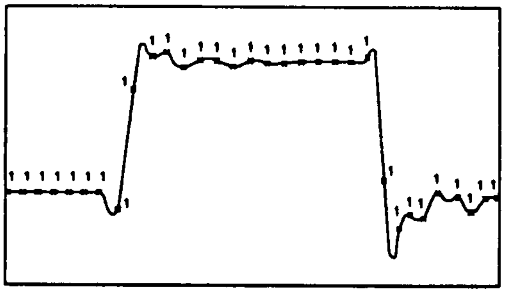
Real-Time

Real-time, or single-shot sampling, digitizes the input signal on its first occurrence.

Advantages: All signal data is acquired in one acquisition cycle. This is an important feature for capturing events that occur only one time, and also allows the capture of events prior to the trigger (pre-trigger).

Disadvantages: The bandwidth of the instrument is determined by the sampling rate of the A/D converter (2 MHz in the HP E1426A). Also, digital reconstruction to fill the points between samples is required. The HP E1426A does not use digital reconstruction.

The figure below illustrates real-time sampling. All points are acquired on one acquisition. Real-time sampling can be used to improve measurement throughput, since only one acquisition is needed to acquire a complete waveform.



ALL POINTS ON ONE ACQUISITION

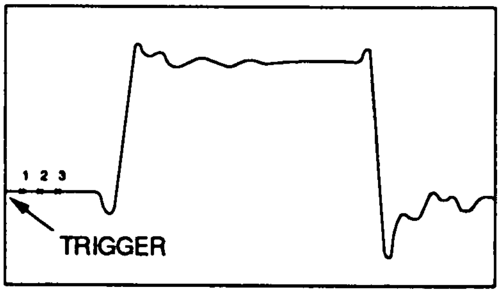
Sequential

Sequential sampling acquires one sample of the signal, a predetermined period of time **after** a trigger occurs. With each successive trigger, the sampling point is delayed further from the trigger point. After enough samples are acquired and digitized, the signal is reconstructed in waveform memory. Sequential sampling requires the signal to be repetitive (not periodic), and that a trigger occurs prior to each sample.

Advantages: The advantage of sequential sampling is that greater accuracy can be provided, since it can use a slower, higher resolution A/D converter.

Disadvantages: Because the acquisition takes place after the trigger, pre- trigger data cannot be sampled. At slower sweep speeds, it will take longer to acquire a waveform because only one point per trigger is sampled.

The figure shown below illustrates sequential sampling. Three acquisitions have been made, with one point acquired each time.



AFTER THREE ACQUISITIONS
(ONE POINT PER ACQUISITION)

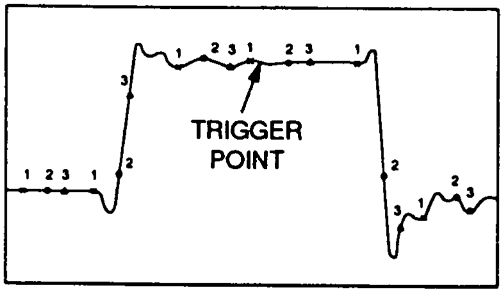
Random Repetitive

Random Repetitive sampling or equivalent-time sampling, is similar to sequential sampling, except that the signal is constantly sampled and digitized. The sampling rate is determined by the instrument's clock, not the trigger repetition rate. Depending upon the sweep speed setting, many points will be acquired on each trigger. Each sample will be separated by the sample period, causing the currently acquired points to be offset from the previously acquired points by a random time. Random Repetitive sampling requires the signal to be repetitive (not periodic), and that a trigger occur during each sample.

Advantages: The advantage of random repetitive sampling is that very precise time interval measurements can be made on very high bandwidth signals.

Disadvantages: Usually requires more than one acquisition to build up the waveform.

The figure shown below illustrates random repetitive sampling. Three acquisitions have been made, with six points acquired each time.



AFTER THREE ACQUISITIONS
(SIX POINTS PER ACQUISITION)

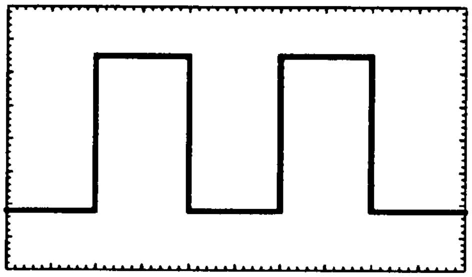
Random Repetitive Sampling

Random-Repetitive sampling takes advantage of two items:

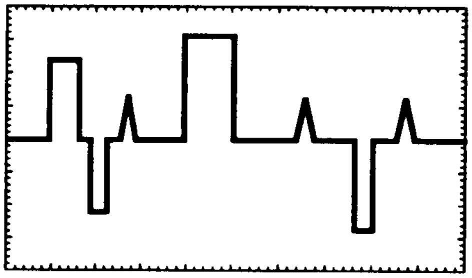
- There is a trigger which always has the same time relationship to the waveform being viewed.
- The majority of signals viewed by a user are repetitive in nature.

Repetitive Signals

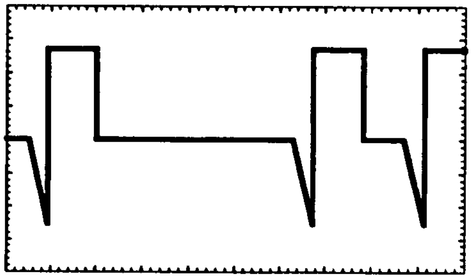
The signal does not have to be periodic to be repetitive, but must occur more than once. Examples of repetitive waveforms are shown below.



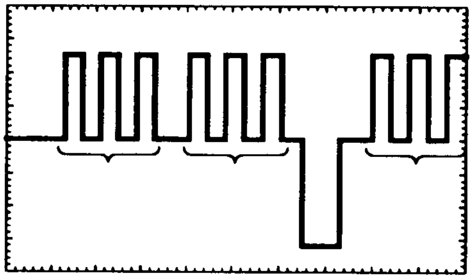
Repetitive square wave



Repetitive glitch



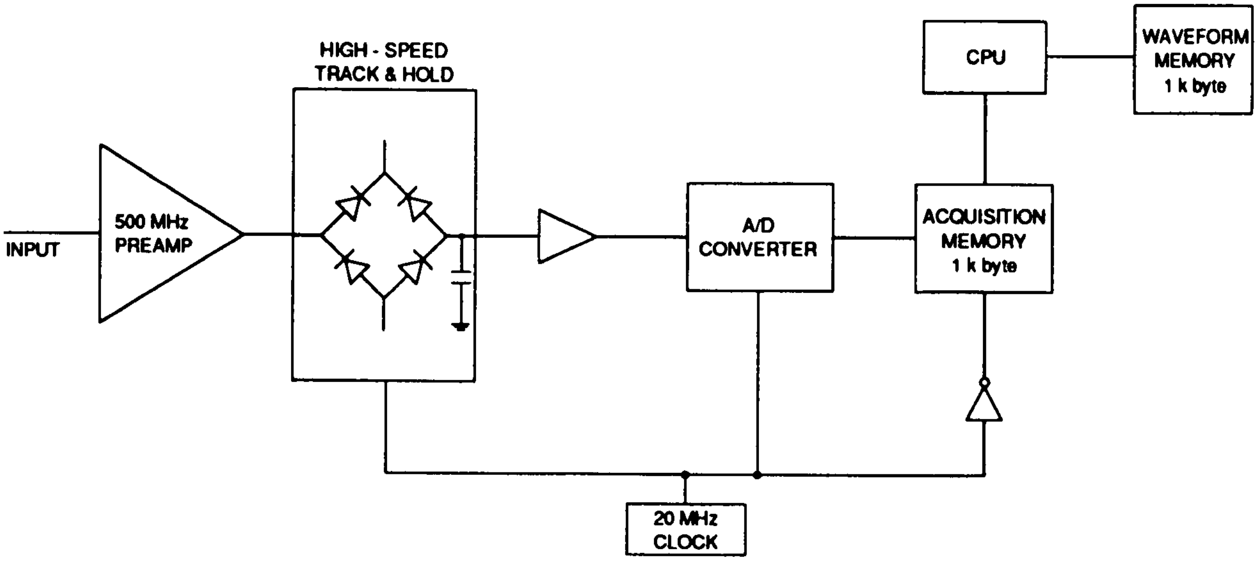
Repetitive glitch



Repetitive pulse train

Basic Operation

These two factors provide the entire basis upon which the random repetitive architecture depends. Knowing these two facts about the waveform, the E1426A can view signals at a bandwidth of 500 MHz, with 1.25% accuracy, and a time resolution of 20 ps. The method the HP E1426A accomplishes random repetitive sampling is illustrated and explained below.



The preamplifier amplifies or attenuates (depending upon the sensitivity setting) the input signal to provide the Track and Hold circuit with the proper voltage range.

Since the A/D Converter cannot digitize a signal with a 500 MHz bandwidth, a high-speed Track and Hold circuit is used. The Track and Hold circuit captures the signal voltage and holds it (on the capacitor) until the lower speed A/D converter can digitize it.

Once the Signal is digitized and stored in the acquisition memory, it is then processed by the CPU, and reconstructed in the waveform memory. The CPU determines where in the waveform memory each point belongs, and then re-constructs the waveform.

A microprocessor controls the acquisition hardware, which performs thousands of acquisitions per second. The sample rate of the digitizer has no affect on the bandwidth of the signal which can be acquired with the HP E1426A oscilloscope.

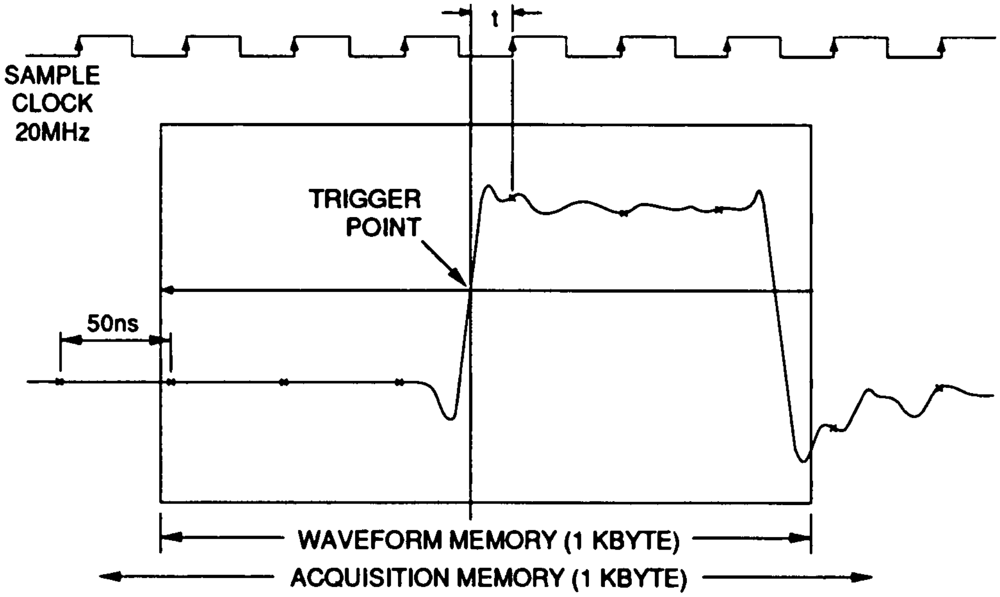
At very slow time base ranges the sample clock frequency is reduced so that the acquisition memory does not fill up with unneeded points. The time resolution will be reduced, but at slower time base ranges, most users do not require fine resolution.

Timing Relationship

The A/D converter runs at 20 Megasamples/second, causing a sample to be taken every 50 nsec. Whenever a trigger occurs, 1024 samples are taken, each spaced by 50 nsec. These points are placed in acquisition memory.

At the same time the waveform is being digitized, a Fine Interpolator circuit determines the exact timing relationship (t) between the trigger point and the sample clock. Since the Fine Interpolator has a resolution of 20 psec, and each sampled point is spaced precisely 50nsec apart, each point in the acquisition memory can be precisely placed in the waveform memory.

The waveform memory is 1 kbyte, and extends to the full time range. The acquisition memory is also 1 kbyte, but extends further than the waveform memory. This is because each point is spaced 50 nsec apart (when sampled at 20 Megasamples/second). This causes some acquired points not to be placed in the waveform memory.

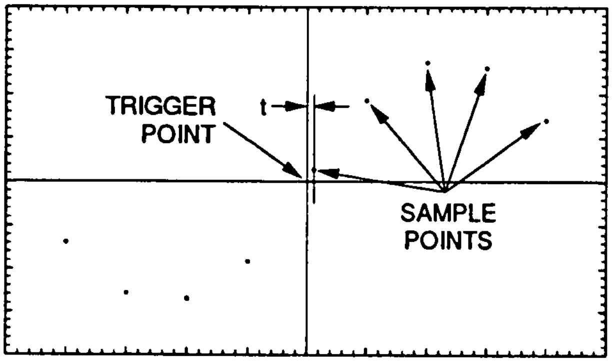


Acquiring a Waveform

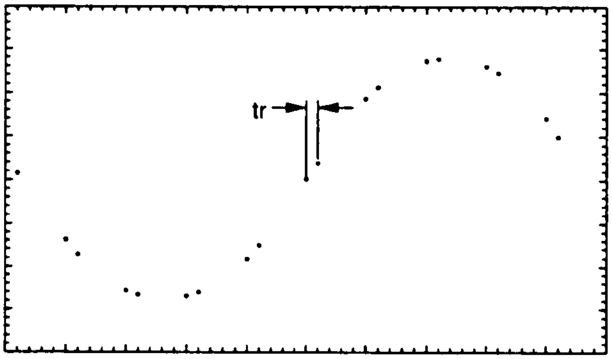
Because random repetitive sampling requires more than one acquisition to build up the waveform, the oscilloscope must make the most efficient use of each acquisition. To acquire data points at different locations on the waveform, each sample is separated by a random time. To accomplish this, the frequency of the sample clock is varied (slightly) to ensure that the trigger and the sample clock always have a random relationship to each other. This causes the currently acquired points to be offset from the previously acquired points by a random time.

The method the oscilloscope acquires a waveform is discussed below.

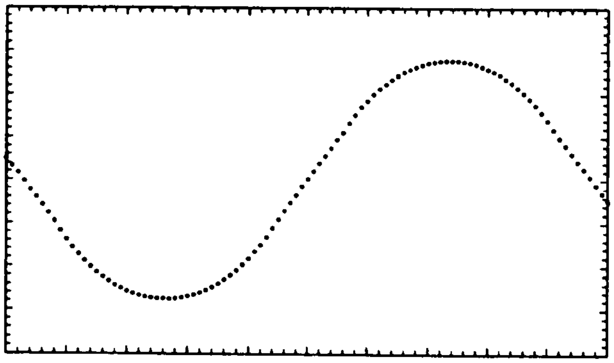
- The figure below shows the points acquired in the waveform memory after one acquisition. Each point is spaced 50 nsec apart. (t) represents the fine interpolator time, or the time between the sample clock and the trigger (discussed previously).



- At the next trigger, another set of points is acquired. Since there is a random relationship between the sample clock and the trigger, the newly acquired points will be offset from the original ones by a random amount of time (tr).



This cycle is repeated (as specified by the oscilloscope setup) until the waveform is acquired.



Digitizing Waveforms

The following section discusses some of the factors which affect the "digitization" process, or acquisition of a full waveform. With random repetitive sampling, several acquisitions may be required to digitize a waveform.

Note

When configuring the Oscilloscope to perform a waveform digitization, the user should set up the oscilloscope to perform the measurement required, NOT the most accurate measurement possible.

When the user instructs the oscilloscope to digitize a waveform, several factors determine how long the digitization will take. These factors include:

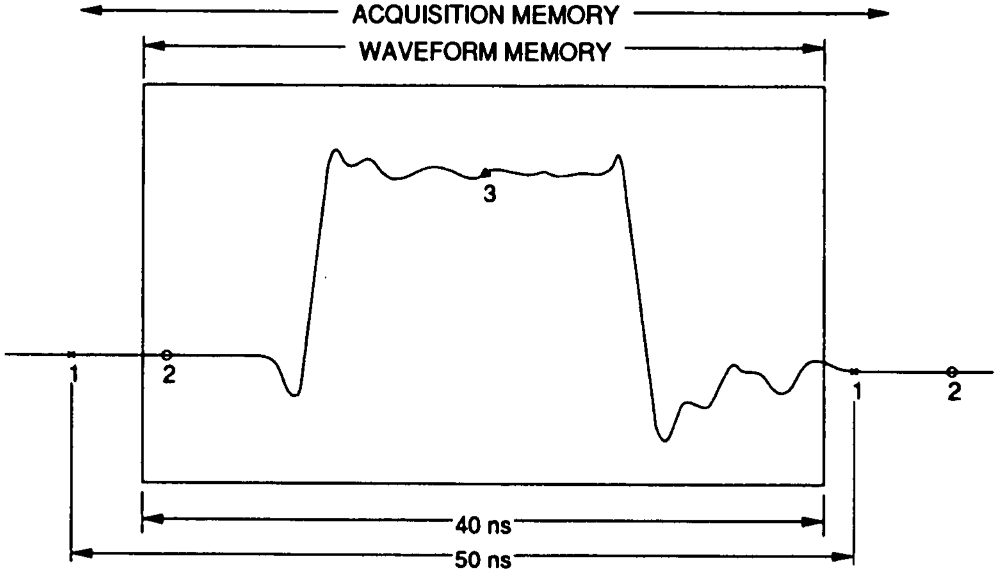
- Time Base,
- Repetition Rate,
- Completion Criteria,
- Acquisition Mode, and
- Number of Points.

Time Base

The time base setting directly affects the time it takes to digitize a waveform.

- At sweep ranges slower than 50µsec full scale, one acquisition will fill up the entire waveform memory (1024 points). This is because each sampled data point is spaced 50 nsec apart.
- At sweep ranges faster than 50µsec full scale, more than one acquisition will need to take place. This is because at a time range of 50µsec, only 1000 points will be acquired.

As an example, at a time base range of 40 nsec full scale, a single acquisition may not acquire any points within the waveform memory, since each point is spaced 50 nsec apart. However, since there is a random relationship between the trigger and the sample clock, points are acquired on the second and third acquisition and the waveform will eventually be constructed in the waveform memory.



Using the same example, it would take less than a hundred milliseconds to fill up the entire waveform memory at a time base range of 100 µsec. Some typical digitization times can be found in figure C-1 later in the appendix.

Maximum Throughput

Selecting Time Base For Maximum Digitizing Speed: Set the time base to give the required resolution for the desired measurement or waveform acquisition. Remember;

- Too fast a sweep speed will slow down the digitization.
- Too slow a speed will reduce the accuracy.

Repetition Rate

Another factor which affects digitization speed is the repetition rate of the trigger event.

Note

Do not confuse repetitive with periodic. A repetitive signal is any signal that repeats itself. It can happen every few milliseconds and then not again for another hour. Several examples of repetitive signals are shown at the beginning of the appendix.

A trigger event is required before the acquisition can take place. More triggers (up to a point) will produce more acquisitions, which will digitize the signal faster.

- If a trigger happens only once an hour, and the time base is set at a fast sweep speed, it could take several hours to digitize the waveform.
- If a trigger happens every microsecond there will be many trigger events which will not be acquired, since the acquisition hardware cannot reset itself in less than a microsecond. Reset time is discussed below.

There will be a certain amount of time when no triggers will be acquired. This is the time it takes the microprocessor to transfer and calibrate the data from the acquisition memory to the waveform memory, and then reset the trigger. The shortest reset time on the HP E1426A is 200 μsec. This means that for maximum digitization speed, a trigger should occur at a faster rate than 1/200 μsec, or faster than 5 kHz. If the repetition rate of the signal is slower than 5 kHz, the oscilloscope may not be digitizing at its fastest rate.

Maximum Throughput

Selecting Repetition Rate For Maximum Digitizing Speed: Always set the repetition rate of the trigger to greater than 5 kHz to ensure that the oscilloscope gets another trigger as soon as it is reset.

Completion Criteria

Selecting too high a percentage of completion is the major contributor of slow measurements and digitizations. Often times the user will set completion to 100%, believing that a full waveform is required for every measurement. If the instrument is set to acquire 500 points and completion criteria of 100% is specified, it will take much longer to digitize that signal than one specified with a 50% completion.

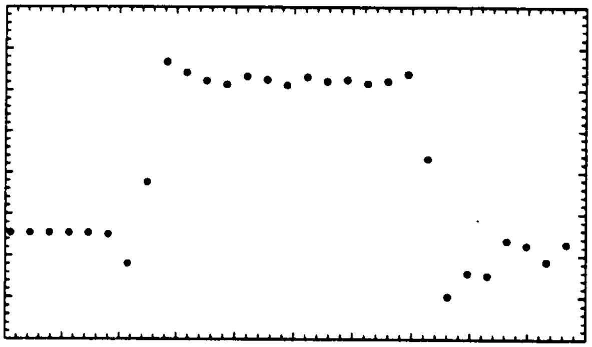
Completion criteria is not a linear relationship. 50% completion means that half the points must be "hit" before the digitization process is complete. It does not specify which 50% of the 500 points need to be filled. The randomness of the trigger and sample clock time relationship ensures that all the points will eventually be filled. However, the time it takes to fill 250 points is not 1/2 the time required to fill 500, since some of the acquisitions may hit the same "time bucket" as a previous acquisition.

The importance of this parameter can be seen by comparing the times required for a waveform to digitize in Figure C-1.

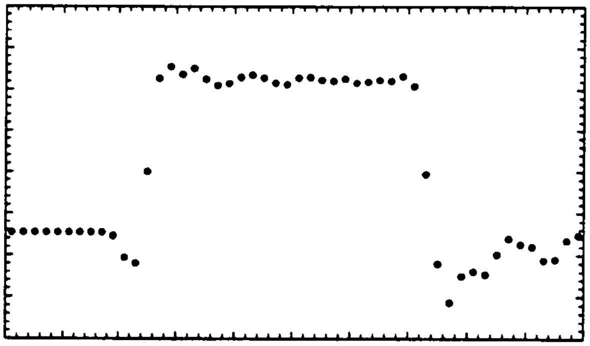
- At a time base range of 100 nsec it takes nearly 1 second to digitize a waveform at 100% completion.
- At a time base range of 100 nsec it takes only 200 msec seconds to digitize to 60% completion.

A comparison of waveforms with different completion criteria's is shown below. In some cases a 20% completion percentage may be all the user requires. If the user only needs the shape of the waveform, a 40% completion would be adequate and digitize time would be reduced.

- At 20% and 40% completion, an outline of the signal can be seen.

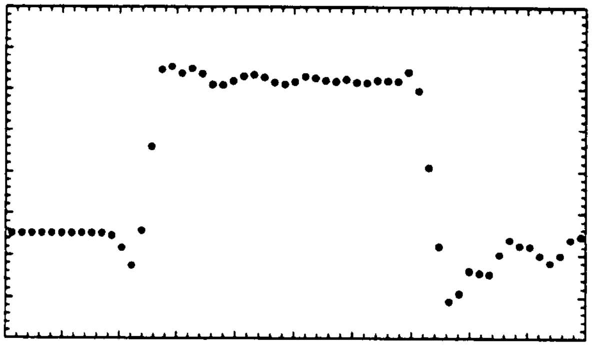


20% Complete

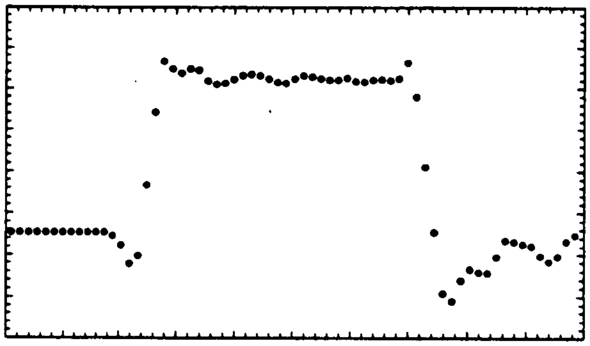


40% Complete

- As the completion percentage increases, the waveform becomes more discernible. At 60% and 80% completion criteria, the waveform is very discernible.

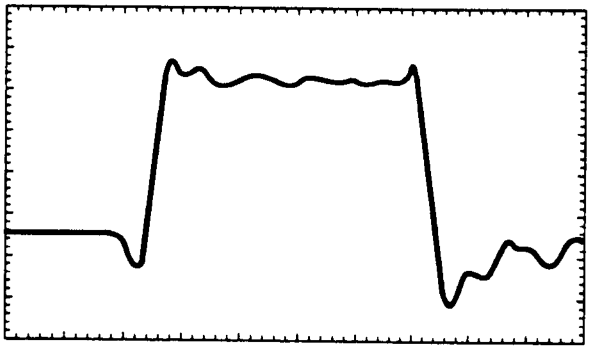


60% Complete



80% Complete

- A 100% completion criteria would give little extra information, but would take 5 times longer to digitize.



100% Complete

Maximum Throughput

Selecting Completion Criteria For Maximum Digitizing Speed: Select the smallest percentage of completion which will satisfy measurement requirements. 60% is a good value. See Accuracy versus Completion later in the appendix for more information.

- The lower the completion percentage, the faster the digitization.
- The higher the completion percentage, the more accurate the measurement results will be.

Acquisition Mode

Averaging is a method which the oscilloscope uses to reduce the amount of random noise present on the signal and improve vertical resolution. If the instrument is set up to average 16 times, each time point must be hit 16 times, and then the average of all 16 values used. Averaging can increase the digitize time by an order of magnitude. It is NOT recommended to use the averaging feature unless needed to reduce noise on the signal. Averaging can improve the repeatability and accuracy of the measurement, but for most cases such accuracy is not needed. The "normal or scalar" mode is recommended for most applications.

Maximum Throughput

Selecting Acquisition Mode For Maximum Digitizing Speed: The acquisition mode should be set to normal or scalar unless the signal is noisy and requires an averaging filter.

Number of Points

The last factor which affects digitization speed is the number of points requested. There is a fairly linear relationship between digitization time and transfer time, and the number of points selected.

- Digitizing Time: 1024 points will take twice as long to digitize as 512 points at fast sweep speeds. The user needs to decide how much time resolution is needed and this will determine the number of points to request. The E1426A defaults to 500 points which, for most users, should give plenty of time resolution.
- Transfer Time: 1024 points will take twice as long to transfer to the controller as 512 points. The HP E1426A supports shared memory so the waveform data can be dumped directly into the controller's memory. However, if the data is being dumped using word serial protocol, waveform transfer time can be reduced by sending only bytes (8 bits) of data rather than whole words (16 bits).

Maximum Throughput

Selecting Number of Points For Maximum Digitizing Speed: To select the number of points, determine the time resolution needed for each digitized point. For example, if the user needs 200 psec resolution at a time base range of 100 nsec, set the number of points to 500. 500 points is a good value for complex waveforms.

Typical Measurement Times

As previously discussed, there are two steps required when making an automatic measurement.

- Digitizing the waveform.
- Measuring the parameter using the digitized waveform.

Each step is performed separately, and will require a certain amount of time to complete.

- The measurement times are basically determined by the measurement algorithm in the instrument, so the user cannot really improve speed in this area. Measurement times are fairly constant with changes in time base setting, completion criteria, and number of points selected. Repetition rate has no effect on the measurement times.
- Digitization process will normally require the most time. Once the waveform is digitized, many measurements can be taken on that waveform without the need for a new digitization.

Determining Typical Measurement Times

To determine the amount of time required for a measurement, add the "digitize" time to the "measure" time as described below. Typical numbers can be found in figures C-1 and C-2. Typical measurement time is calculated as follows:

1. Select the time base setting which will be used for the measurement. Use figure C-1 to determine the digitize time using 60% completion. This is referred to as the "digitize" time.

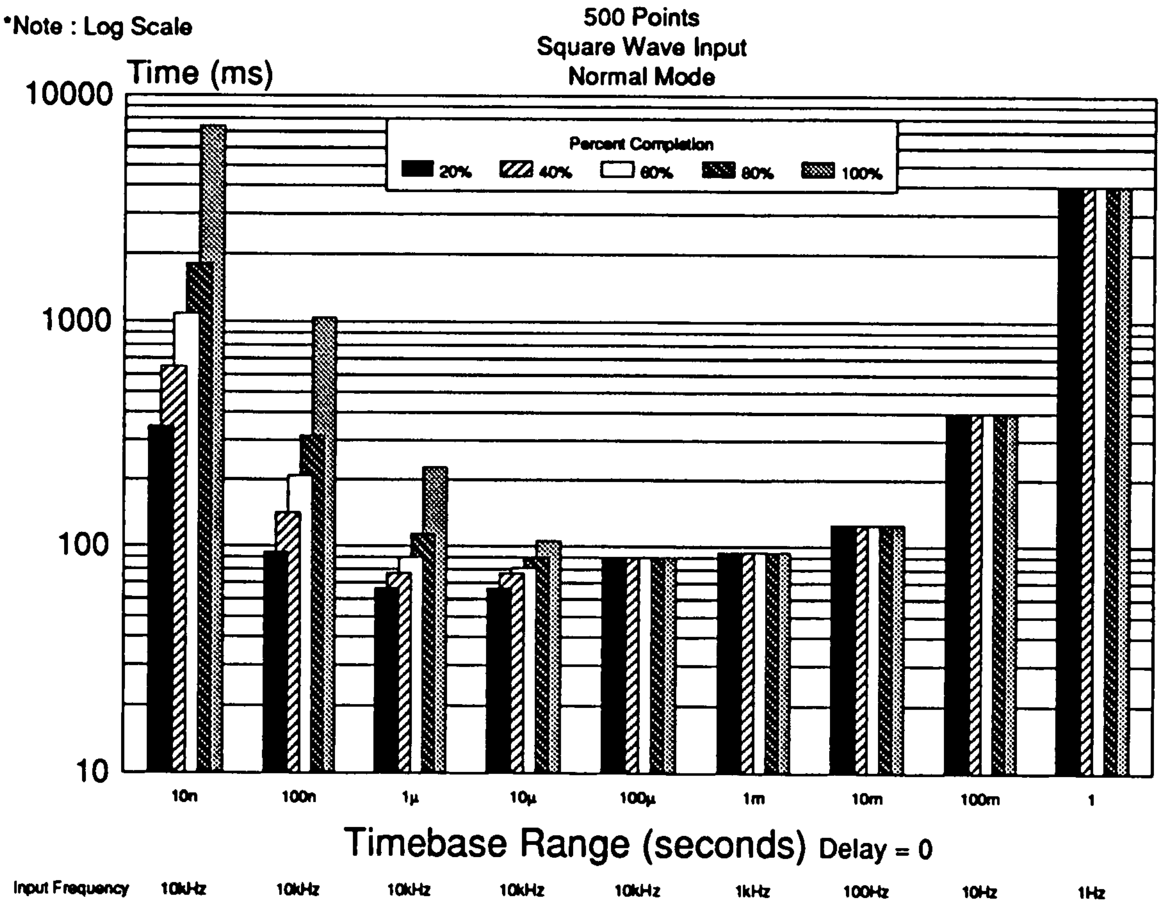
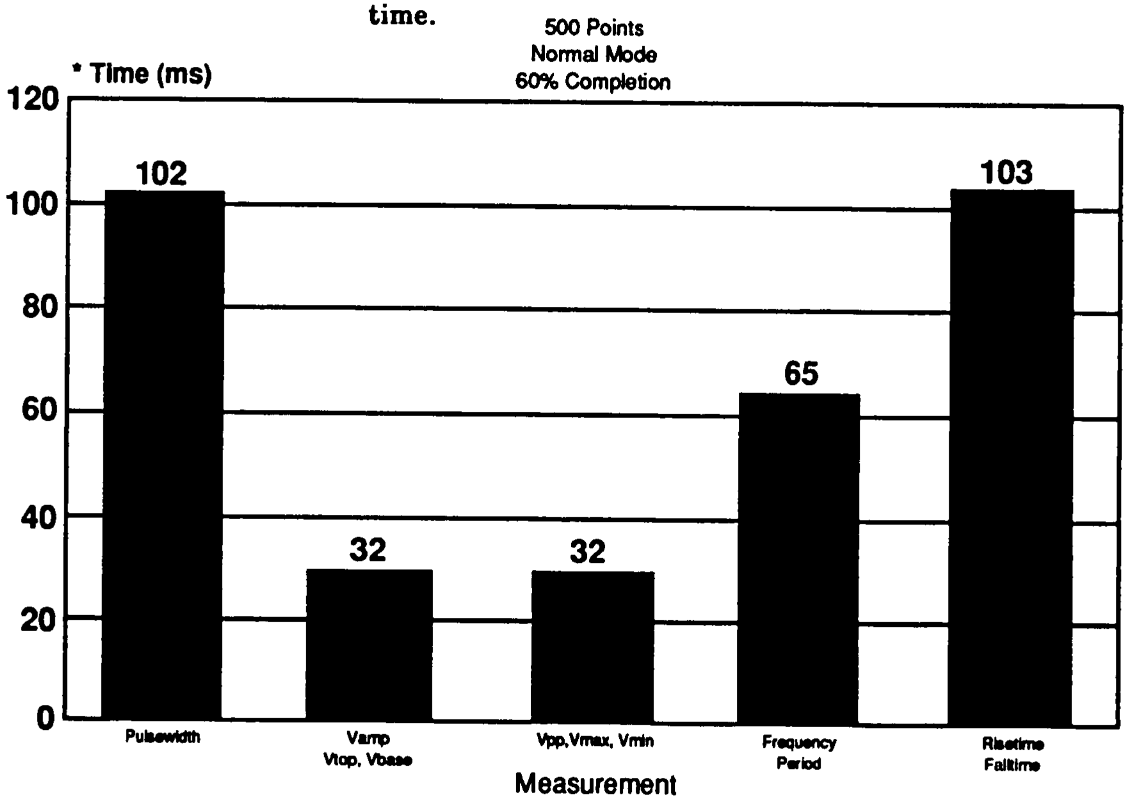


Figure C-1. Typical Digitize Times.

2. Use Figure C-2 to determine the measurement time of the type of measurement being performed. Similar measurements will have similar times. The time shown is the time from sending out the command to receiving the measurement using a Radix VXI microcontroller. This time is referred to as the "measurement" time.



* Times do not include digitize times

Figure C-2. Typical Measurement Times.

3. Add the "digitize" time to the "measurement" time to determine a typical time required by the E1426A to perform the measurement.

For example, to perform a VAMP measurement at a time base setting of 10 µsec would take 112 msec. The "digitize" time would take 80 msec and the "measurement" time would take 32 msec.

Maximum Throughput

If multiple measurements will be taken on only one waveform, then include only one digitize time and add the required measurements. This can significantly improve measurement throughput.

Accuracy vs. Percent Completion

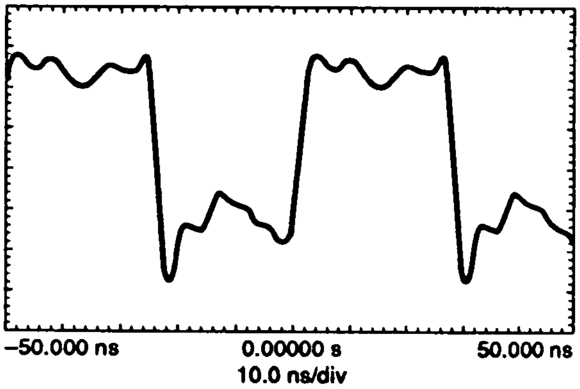
The HP E1426A Oscilloscope often acquires more data than is needed for typical measurements taken in ATE systems. For example, to determine the pulse width of a particular waveform, does the oscilloscope really need 100%, or would 60% be enough to provide an accurate measurement?

To answer the question "How small a completion percentage can be used and still provide accurate results?", an experiment was performed to determine the optimum completion criteria for typical waveform measurements. Data on eight different measurements was taken:

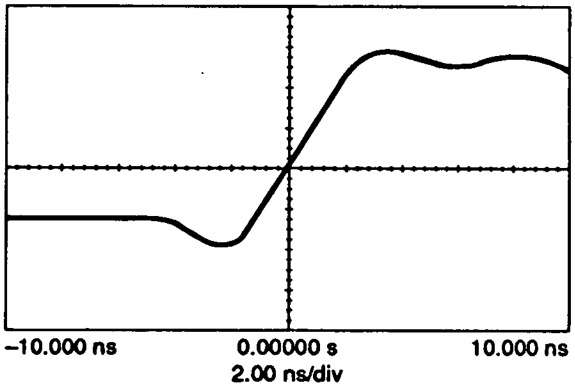
- Pulse Width (PWIDth)
- VPP,
- VMAXimum
- VAMPlitude
- VTOP (HIGH)
- VBASe (LOW)
- PERiod
- RISetime (RTIMe)

A waveform was selected which is typical of the types test engineers might encounter. The waveforms for each of the measurements are shown below.

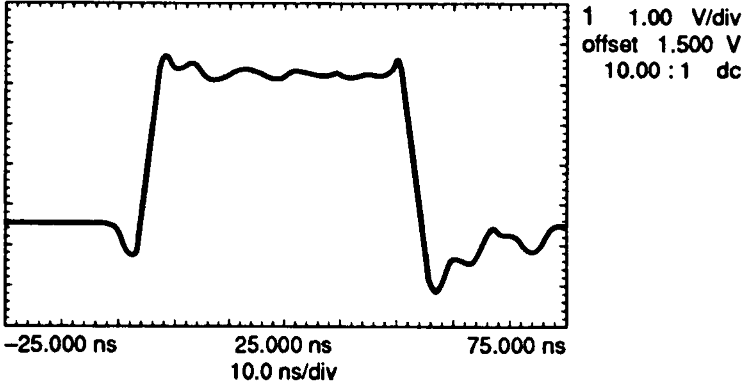
Period



Rise time



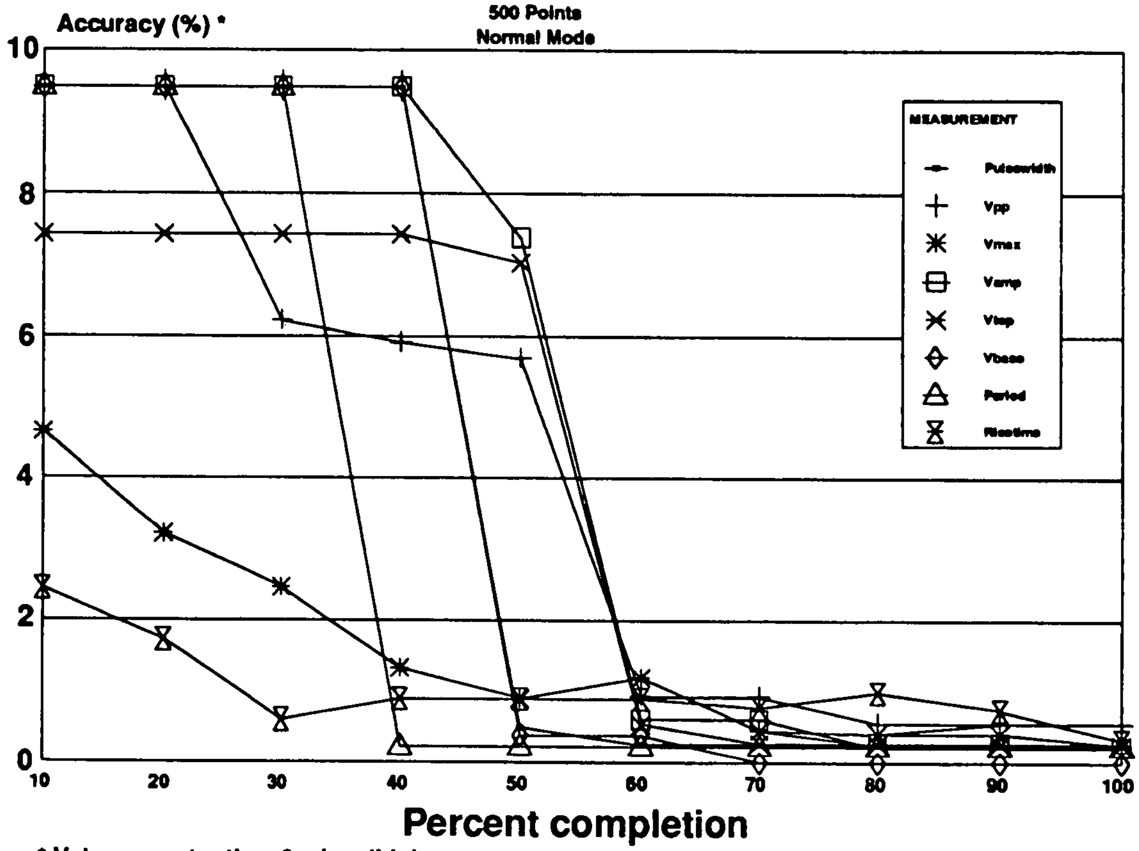
Pwidth, Vpp, Vmax, Vamp, Vtop, Vbase



The waveform was digitized at the completion criteria specified and then a measurement was taken on that waveform. This setup was performed 100 times and the worst case measurement was compared to the actual value using this formula:

(worst case value – actual value)/full scale setting * 100%

Using this formula, the data could be normalized to any full scale setting. Completion criteria's were used which ranged from 10% to 100% on a point selection of 500. The figure below shows the results of this experiment.

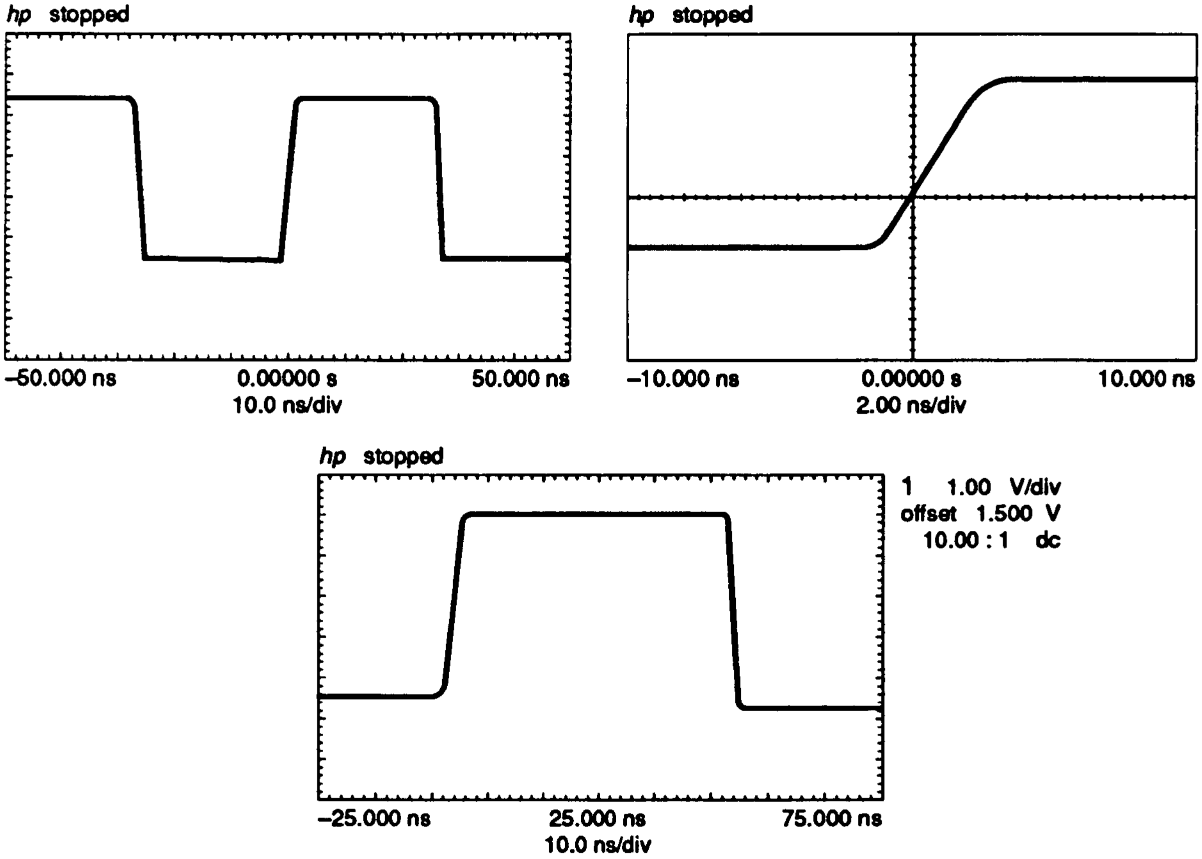


* Values greater than 8 = Invalid data

Note that any value over 8 actually represents invalid measurements because of lack of data points. From this data it can be seen that a completion criteria of greater than 60% does not improve the measurement accuracy very much. Some measurements, such as rise time, require even fewer points to provide accurate results.

This experiment was performed on a fairly complex waveform. On simpler waveforms (shown below) even fewer points would be needed to provide accurate measurements. However, by using a completion criteria of 60% with 500 data points, the test engineer can perform accurate measurements in the least amount of the time and use the oscilloscope to its full potential.

Pwidth, Vpp, Vmax, Vamp, Vtop, Vbase



Throughput Comparison HP 54503A - E1426A

This section compares the various digitize, measurement and transfer times of the HP 54503A (rack and stack) vs the E1426A (VXI) 500 MHz Digitizing Oscilloscopes. All of the measurements were done with the RadiSys AT-compatible computer. The RadiSys computer was located in slot 0 and the E1426A oscilloscope in slot 2. The HP 54503A Oscilloscope was connected to the GP-IB port of the RadiSys computer. The times recorded are typical times.

Note

The HP 54503A was set up with the screen off to reduce the amount of overhead in waveform processing.

Digitize Times

The "digitize" time is the time from when the digitize command is sent out until the Oscilloscope signals the computer that it has completed the operation. Both instruments were set up as follows:

- Acquire 500 points.
- 60% complete.
- Input a 10 kHz square wave.

Digitize times were as follows:

Time Range	HP 54503A	HP E1426A
100 ns	224 ms	202 ms
100 µs	110 ms	92 ms

Transfer Times

The "transfer" time is the time from when the WAVeform:DATa? query is sent to the time all the data is transferred. The data is transferred using "byte" format (one data byte is sent for each data point).

Transfer times were as follows:

# points	HP 54503A	HP E1426A	HP E1426A (w/shared ram)
1024	35 ms	45 ms	0
512	27 ms	25 ms	0
256	24 ms	15 ms	0
128	21 ms	10 ms	0
64	21 ms	7 ms	0

Measurement Times

The "measurement" time is the time from when the measurement query is sent until the number is returned. For the values of "t" in the table below see Figure C-2.

Measurement times were as follows:

HP 54503A	HP E1426A
t+42 ms	meas. time (t)

Programming Example

Test Programs Using Functions

Two example programs are provided, one Compatible and one SCPI, to assist in programming the oscilloscope. Both programs will initialize the system, digitize to acquire data, then analyze the acquired data. See programming examples for additional information.

Each example is written using:

- an HP-IB select code of 7, primary address of 09, and secondary address of 01 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC
- An HP HP 8116A or suitable Pulse Generator

Execute:

```
10    !DIGI_COM    E1426A  HP Basic Compatible program
20    !
30    !MAIN PROGRAM
40    !
50    REAL Preamble(1:10)              !array for preamble information
60    !
70    CLEAR SCREEN
80    PRINT "This example program will perform the following tasks:"
90    PRINT "               a.  initialize interface and scope"
100   PRINT "               b.  digitize to acquire  data"
110   PRINT "               c.  do two simple parametric measurements"
120   PRINT "               d.  acquire waveform data from scope"
130   PRINT "               e.  store data to disk"
140   PRINT "               f.  retrieve data from disk"
150   PRINT "               g.  draw signal on computer"
160   PRINT
170   PRINT "Before running this sample program, set addresses to your"
180   PRINT "system configuration.  This sample program ASSIGNS:"
190   PRINT "            Isc to 7     (HP-IB interface)"
200   PRINT "            Scope to 70901  (scope address)"
210   PRINT
220   PRINT "PRESS Continue TO START PROGRAM OR SHIFT/RESET TO TERMINATE"
230   PAUSE
240   GOSUB Initialize                  !initialize interface and scope
250   GOSUB Capture_data                !digitize signal
260   GOSUB Measure                     !do Vpp and frequency measurement
270   GOSUB Get_waveform                !transfer signal data to controller
280   GOSUB Save_waveform               !store to disk
290   !
300   CLEAR SCREEN
310   PRINT "The waveform data and preamble information have now been"
320   PRINT "read from the scope and stored in the computer's disk."
330   PRINT "This information will now be retrieved from the disk, and"
340   PRINT "will be used to plot the waveform, calculate and plot the"
350   PRINT "integral, as well as calculate scaling information."
360   PRINT "Press CONTINUE to continue."
```

```
370   PAUSE
380   GOSUB Retrieve_wave             !retrieve from disk
390   GOSUB Graph                     !draw waveform
400   STOP
410   !
420   !
430   !INITIALIZE INTERFACE AND SCOPE
440 Initialize:                      !
450   PRINT "INITIALIZE"
460   ON TIMEOUT 7,20 GOTO 1620      !timout on hpib
470   ASSIGN @Isc TO 7               !Interface Select Code = 7
480   ASSIGN @Scope TO 70904         !scope address
490   CLEAR @Isc                     !clear HP-IB interface
500   OUTPUT @Scope;":SYSTEM:LANGUAGE COMP"!set to compatible
510   WAIT 1                         !1 sec wait for language switch
520   !The language switch does an *RST and hence if the language switch
530   !is in the program, the following *RST is redundant and should be
540   !removed.
550   OUTPUT @Scope;"*RST"           !set scope to default config
560   OUTPUT @Scope;":AUTOSCALE"     !AUTOSCALE
570   OUTPUT @Scope;":SYSTEM:HEADER OFF" !turn headers off
580     ! The following four commands are set by the *RST, but are
590     ! included here for completeness to emphasize the fact that
600     ! they can be controlled by the programmer.
610     !
620   OUTPUT @Scope;":WAVEFORM:SOURCE CHAN1"  !set source to channel 1
630   OUTPUT @Scope;":ACQUIRE:TYPE NORMAL"    !set to normal acquisition mode
640   OUTPUT @Scope;":ACQUIRE:COMPLETE 100"   !set complete criteria
650   OUTPUT @Scope;":ACQUIRE:POINTS 500"    !set # of pts to 4000
660     !
670   CLEAR SCREEN
680   RETURN
690  !
700  ! DIGITIZE signal:  DIGITIZE is a macro that will acquire data to
710  ! programmer's specification and will stop when complete.
720  !
730 Capture_data:                   !
740   OUTPUT @Scope;":DIGITIZE CHANNEL1"  !DIGITIZE channel 1
750   RETURN
760   !'
770   !measure the Vpp and frequency of the digitized signal
780 Measure:                        !
790   CLEAR SCREEN
800   OUTPUT @Scope;"measure:vpp?;frequency?" !measure the signal
810   ENTER @Scope;V,F
820   PRINT "Vpp = ";V;" volts and frequency = ";F;" Hz"
830   PRINT
840   PRINT "Press Continue to transfer and plot waveform data to controller"
850   PAUSE
860   RETURN
870   !
880   !read waveform data and preamble from scope; waveform data is in
890   !scope format and the preamble information is used to convert
900   !it to voltage/timing information.
910   !
920 Get_waveform:                    !
930     !
940   OUTPUT @Scope;":WAVEFORM:FORMAT COMP"   !format data in compressed format
950   OUTPUT @Scope;":WAVEFORM:DATA?"         !query scope for data
960   ENTER @Scope USING "#,1A,";One_char$    !strip off header & size
970   IF One_char$="#" THEN Found_pound
980   PRINT "BAD DATA"
990   STOP
1000 Found_pound:    !read record length from header
```

```
1010  ENTER @Scope USING "#,1D";Digits        !get length of record
1020  ENTER @Scope USING "#,"&VAL$(Digits)&"D";Length
1030  PRINT "reading ";Length;" bytes from scope"
1040  !
1050  !allocate an array for the waveform data.  After the array has been
1060  !read in, one extra byte read is done to input the line feed (10)
1070  !attached to the end of the scope's output buffer.
1080  !
1090  ALLOCATE INTEGER Waveform(1:Length)
1100  ENTER @Scope USING "#,B";Waveform(*)     !read waveform information
1110  ENTER @Scope USING "-K,B";End$           !get last byte (line feed)
1120  OUTPUT @Scope;":WAVEFORM:PREAMBLE?"      !query for preamble
1130  ENTER @Scope;Preamble(*)                 !read preamble information
1140  RETURN
1150  !
1160  !
1170  !plot waveform data to display
1180  !
1190 Graph:                                    !
1200  GCLEAR                                   !initialize graphics
1210  CLEAR SCREEN
1220  GINIT
1230  GRAPHICS ON
1240  VIEWPORT 0,130,35,100
1250  WINDOW 1,Preamble(3),0,256
1260  FRAME
1270  PEN 4
1280  MOVE 0,0
1290  FOR I=1 TO Preamble(3)                   !plot data points
1300    MOVE I,Waveform(I)
1310    DRAW I,Waveform(I)
1320  NEXT I
1330  OUTPUT @Scope;"TIMEBASE:RANGE?"
1340  ENTER @Scope;Srange
1350  OUTPUT @Scope;"TIMEBASE:DELAY?"
1360  ENTER @Scope;Dvalue
1370  OUTPUT @Scope;"CHAN1:RANGE?"
1380  ENTER @Scope;Vrange
1390  OUTPUT @Scope;"CHAN1:OFFSET?"
1400  ENTER @Scope;Offset
1410  PRINT TABXY(0,18),"Vertical=";Vrange;" V";TAB(50),"Offset = ";Offset;"V"
1420  PRINT TABXY(0,19),"Time=";Srange;" s";TAB(50),"Delay =";Dvalue;"S"
1430  RETURN
1440    !
1450    !save waveform data and preamble information to computer disk
1460    !
1470 Save_waveform:                         !
1480  ON ERROR GOTO 1500
1490  PURGE "WAVESAMPLE"
1500  OFF ERROR
1510  CREATE BDAT "WAVESAMPLE",1,4080
1520  ASSIGN @Path TO "WAVESAMPLE"
1530  OUTPUT @Path;Waveform(*),Preamble(*)
1540  RETURN
1550   !
1560   !retrieve waveform data and preamble infromation from disk
1570   !
1580 Retrieve_wave:                         !
1590  ASSIGN @Path TO "WAVESAMPLE"
1600  ENTER @Path;Waveform(*),Preamble(*)
1610  RETURN
1620  PRINT "timeout error -- check cables, addresses, etc. and rerun"
1630  STOP
1640  END
```

```
10    !DIGI_SCP            HP Basic SCPI program
20    !
30    !MAIN PROGRAM
40    !
50    REAL Preamble(1:10)                 !array for preamble information
60    !
70    CLEAR SCREEN
80    PRINT "This example program will perform the following tasks:"
90    PRINT "              a.  initialize interface and scope"
100   PRINT "              b.  digitize and acquire data"
110   PRINT "              c.  do simple parametric measurements on signal"
120   PRINT "              d.  store data to disk"
130   PRINT "              e.  retrieve data from disk"
140   PRINT "              f.  draw signal on computer"
150   PRINT
160   PRINT "Before running this sample program, set addresses to your"
170   PRINT "system configuration.  This sample program ASSIGNS:"
180   PRINT "   Isc to 7         HP-IB interface"
190   PRINT "   scope to 70901   scope address"
200   PRINT
210   PRINT "PRESS Continue TO START OR SHIFT/RESET TO TERMINATE"
220   PAUSE
230   GOSUB Initialize                    !initialize interface and scope
240   GOSUB Capture_data                  !digitize signal
250   GOSUB Measure                       !query for freq and vpp. Print values
260   GOSUB Get_waveform                  !transfer signal data to controller
270   GOSUB Save_waveform                 !store to disk
280   !
290   CLEAR SCREEN
300   PRINT "The waveform data and preamble information have now been"
310   PRINT "read from the scope and stored in the computer's disk."
320   PRINT "This information will now be retrieved from the disk, and"
330   PRINT "will be used to plot the waveform, calculate and plot the"
340   PRINT "integral, as well as calculate scaling information."
350   PRINT "Press CONTINUE to continue."
360   PAUSE
370   GOSUB Retrieve_wave                 !retrieve from disk
380   GOSUB Graph                         !draw waveform
390   STOP
400   !
410   !INITIALIZE INTERFACE AND SCOPE
420 Initialize:                           !
430   PRINT "INITIALIZE"
440   ON TIMEOUT 7,20 GOTO 1660
450   ASSIGN @Isc TO 7                    !Interface Select Code = 7
460   ASSIGN @Scope TO 70901              !scope address
470   CLEAR @Isc                          !clear HP-IB interface
480   OUTPUT @Scope;":SYSTEM:LANGUAGE SCPI"!set to SCPI
490   WAIT 1                              !1 sec wait for language switch
500   !The language switch does an *RST and  hence if the language switch
510   !is in the program, the following *RST is redundant and should be
520   !removed.
530   OUTPUT @Scope;"*RST"                !set scope to default config
540   OUTPUT @Scope;"*CLS"                !clear status registers
550   OUTPUT @Scope;":SYST:AUT"           !AUTOSCALE
560     !
570     !the following 3 commands are the default configuration setting
580     !that the RST sets up; but, they are included here for clarity and
590     !completeness.  This ensures scope is configured for DIGITIZE, if
600     !*RST was not done.
610     !
620   OUTPUT @Scope;":AVER:TYPE SCAL"         !set to normal acq mode
630   OUTPUT @Scope;":SWEEP:POIN:COMP 100"    !set complete criteria
640   OUTPUT @Scope;":SWEEP:POINTS 500"       !set # of pts to 500
```

```
650      !
660    CLEAR SCREEN
670    RETURN
680    !
690   !DIGITIZE signal:  INIT is a macro that will acquire data
700   !programmer's specification and will stop when complete.
710   !
720  Capture_data:                          !
730    OUTPUT @Scope;":INP1 ON"                 !turn on channel 1
740    OUTPUT @Scope;"ABORT"                    !stop acquisition
750    OUTPUT @Scope;":INIT"                    !macro to acquire data & stop
760    RETURN
770    !
780    !complete scope configuration; DIGITIZE and acquire waveform data
790    !and preamble (voltage/timing) information into computer.
800    !
810    !
820    !Find frequency and Vpp
830    !
840  Measure:                               !
850    OUTPUT @Scope;":CONFIGURE:VOLT:FREQUENCY (@INP1)"
860    OUTPUT @Scope;":FETCH:VOLT:FREQ?"        !get frequency
870    ENTER @Scope;Freq
880    CLEAR SCREEN
890    PRINT "Frequency = ";Freq;" Hz"
900    OUTPUT @Scope;":CONFIGURE:VOLT:AMPL (@INP1)"
910    OUTPUT @Scope;":FETCH:VOLT:AMPL?"
920    ENTER @Scope;Vpp
930    PRINT "Vpp = ";Vpp;" V"
940    PRINT
950    PRINT "PRESS continue to transfer data to computer and plot"
960    PAUSE
970    RETURN
980  Get_waveform:                          !
990    OUTPUT @Scope;":FORM INT,8"              !data in 8 bit format across bus
1000   OUTPUT @Scope;":TRACE:DATA? INP1"        !query scope for channel 1 data
1010   ENTER @Scope USING "#,1A,";One_char$     !strip off header & size
1020   IF One_char$="#" THEN Found_pound
1030   PRINT "BAD DATA"
1040   STOP
1050  Found_pound:     !read record length from header
1060   ENTER @Scope USING "#,1D";Digits         !get length of record
1070   ENTER @Scope USING "#,"&VAL$(Digits)&"D";Length
1080   PRINT "reading ";Length;" bytes from scope"
1090   !
1100   !allocate an array for the waveform data.  After the array has been
1110   !read in, one extra byte read is done to input the line feed (10)
1120   !attached to the end of the scope's output buffer.
1130   !
1140   ALLOCATE INTEGER Waveform(1:Length)
1150   ENTER @Scope USING "#,B";Waveform(*)     !read waveform information
1160   ENTER @Scope USING "-K,B";End$           !get last byte (line feed)
1170   OUTPUT @Scope;":TRACE:PRE? INP1"         !query for preamble
1180   ENTER @Scope;Preamble(*)                 !read preamble information
1190   RETURN
1200   !
1210   !plot waveform data to display
1220   !
1230  Graph:                                    !
1240   GCLEAR                                   !initialize graphics
1250   CLEAR SCREEN
1260   GINIT
1270   GRAPHICS ON
1280   VIEWPORT 0,130,35,100
```

```
1290  WINDOW 1,Preamble(3),0,128              !data between 0 and 128
1300  FRAME
1310  PEN 4
1320  MOVE 0,0
1330  FOR I=1 TO Preamble(3)                  !plot data points
1340    MOVE I,Waveform(I)
1350    DRAW I,Waveform(I)
1360  NEXT I
1370  OUTPUT @Scope;"SWEEP:TIME:RANGE?"
1380  ENTER @Scope;Srange
1390  OUTPUT @Scope;"SWEEP:TIME:DELAY?"
1400  ENTER @Scope;Dvalue
1410  OUTPUT @Scope;"VOLT1:RANGE?"
1420  ENTER @Scope;Vrange
1430  OUTPUT @Scope;"VOLT1:RANGE:OFFSET?"
1440  ENTER @Scope;Offset
1450  PRINT TABXY(0,18),"Vertical=";Vrange;" V";TAB(50),"Offset = ";Offset;"V"
1460  PRINT TABXY(0,19),"Time=";Srange;" s";TAB(50),"Delay =";Dvalue;"S"
1470  RETURN
1480    !
1490    !save waveform data and preamble information to computer disk
1500    !
1510 Save_waveform:                        !
1520  ON ERROR GOTO 1540
1530  PURGE "WAVESAMPLE"
1540  OFF ERROR
1550  CREATE BDAT "WAVESAMPLE",1,4080
1560  ASSIGN @Path TO "WAVESAMPLE"
1570  OUTPUT @Path;Waveform(*),Preamble(*)
1580  RETURN
1590   !
1600   !retrieve waveform data and preamble infromation from disk
1610   !
1620 Retrieve_wave:                        !
1630  ASSIGN @Path TO "WAVESAMPLE"
1640  ENTER @Path;Waveform(*),Preamble(*)
1650  RETURN
1660  PRINT "timeout error -- check cables, addresses, and rerun"
1670  END
```

Index

Hewlett-Packard
Printed in the USA