HP 75000 SERIES C

1 GSa/s
Digitizing Oscilloscope
HP E1428A

User's Manual

HEWLETT
PACKARD

Manual Part Number: E1428-97000

Printed: October 1994
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 Duration and conditions of warranty for this product may be superseded when the product is integrated into (becomes a part of) other HP products.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 E1428A 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-Entstorbestimmungen unter Betriebsbedingungen an seiner Grundstucksgrenze eingehalten werden.

Manufacturer's Declaration

This is to certify that the equipment HP E1428A 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.

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 E1428-97000) . . . . . October 1994

Safety Summary

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.

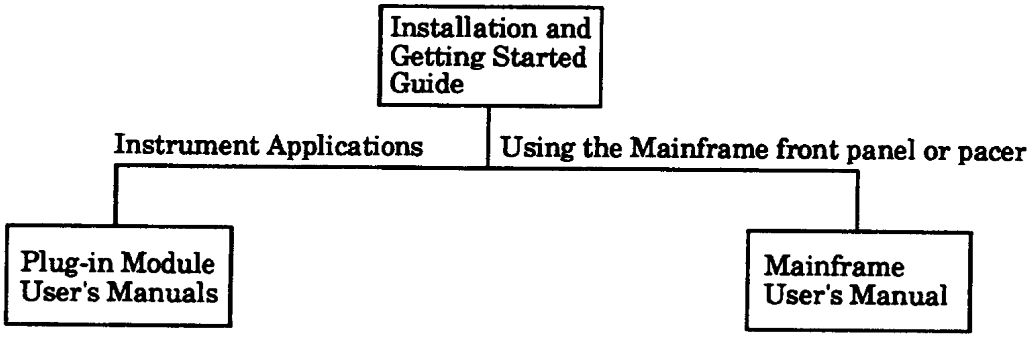
DO NOT substitute parts or modify equipment: Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the product. Return the product to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety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

Safety Symbols

~ **LINE**	AC line voltage input receptacle.	ϟ	Indicates hazardous voltages.
⚠	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.	ATTENTION Static Sensitive	Affixed to product containing static sensitive devices – use antistatic handling procedures to prevent electrostatic discharge damage to components.
⏚ OR ⏚	Indicates the field wiring terminal that must be connected to earth ground before operating the equipment – protects against electrical shock in case of fault.	**NOTE**	Calls attention to a procedure, practice, or condition that requires special attention by the reader
⊥ OR ⊥	Frame or chassis ground terminal – typically connects to the equipment's metal frame.	**WARNING**	Calls attention to a procedure, practice, or condition that could cause bodily injury or death.
∿	Alternating current (AC).	**CAUTION**	Calls attention to a procedure, practice, or condition that could possibly cause damage to equipment or permanent loss of data.
⎓	Direct current (DC).		

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

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 E1428A 1 GSa/s Digitizing Oscilloscope Module. Except where noted, the term "Oscilloscope" refers to the E1428A 1 GSa/s 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 54510A 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.

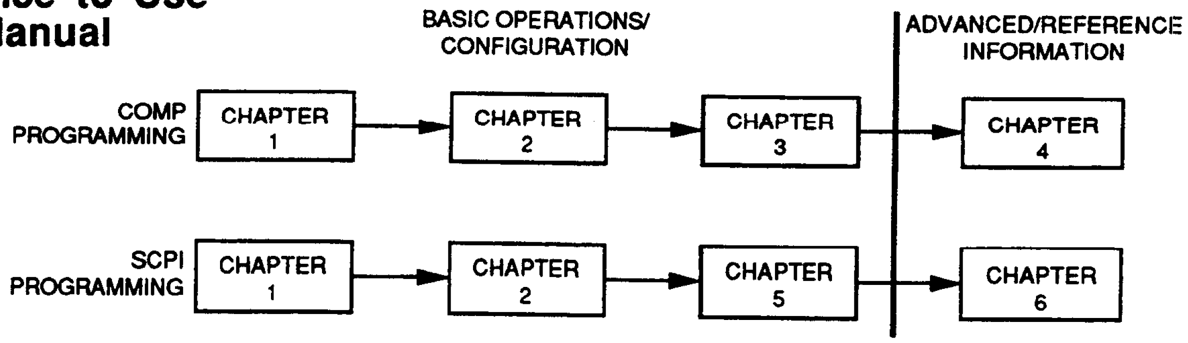
- Chapters 1 and 2 provide Oscilloscope descriptions and configurations.

  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 measurement throughput. Appendix D provides programming examples.

Suggested Sequence to Use This Manual



Contents

1. **Getting Started**

2. **Configuring the Oscilloscope Module**

3. **Using the Oscilloscope with HP 54510A Compatible Commands**

4. **Oscilloscope HP 54510A Compatible Language Command Reference**

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 54510A 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 E1428A 1 GSa/s Digital Oscilloscope module.

General Description

The Oscilloscope module is a general purpose, two channel, 1 GSa/s 250 MHz 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.

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

High precision and advanced triggering enables repeatable and accurate measurements on simple or complex waveforms. Capture of complicated and elusive events on both 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
- 48 nonvolatile set-up memories
- Four nonvolatile waveform memories
- Pre- and post-trigger viewing capability
- Measurement limit test capability
- Waveform compare capability

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

Basic Operation

The Oscilloscope module is comprised of a CPU/Acquisition Printed Circuit Assembly (PCA) (HP P/N E1428-69501) and a Memory PCA (HP P/N E1428-69502).

The CPU/Acquisition PCA attenuates/amplifies each of the two inputs. The signal is then routed to an A/D Converter where it is changed into a digital word. This digital information is stored for use by the CPU. 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.
- External trigger input doubles as external sample clock input.

The CPU/Acquisition PCA also 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 to perform the specific task. When the digital waveform information is received from the Acquisition, all the user requested parameters are measured and routed to the mainframe. Additional functions include:

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

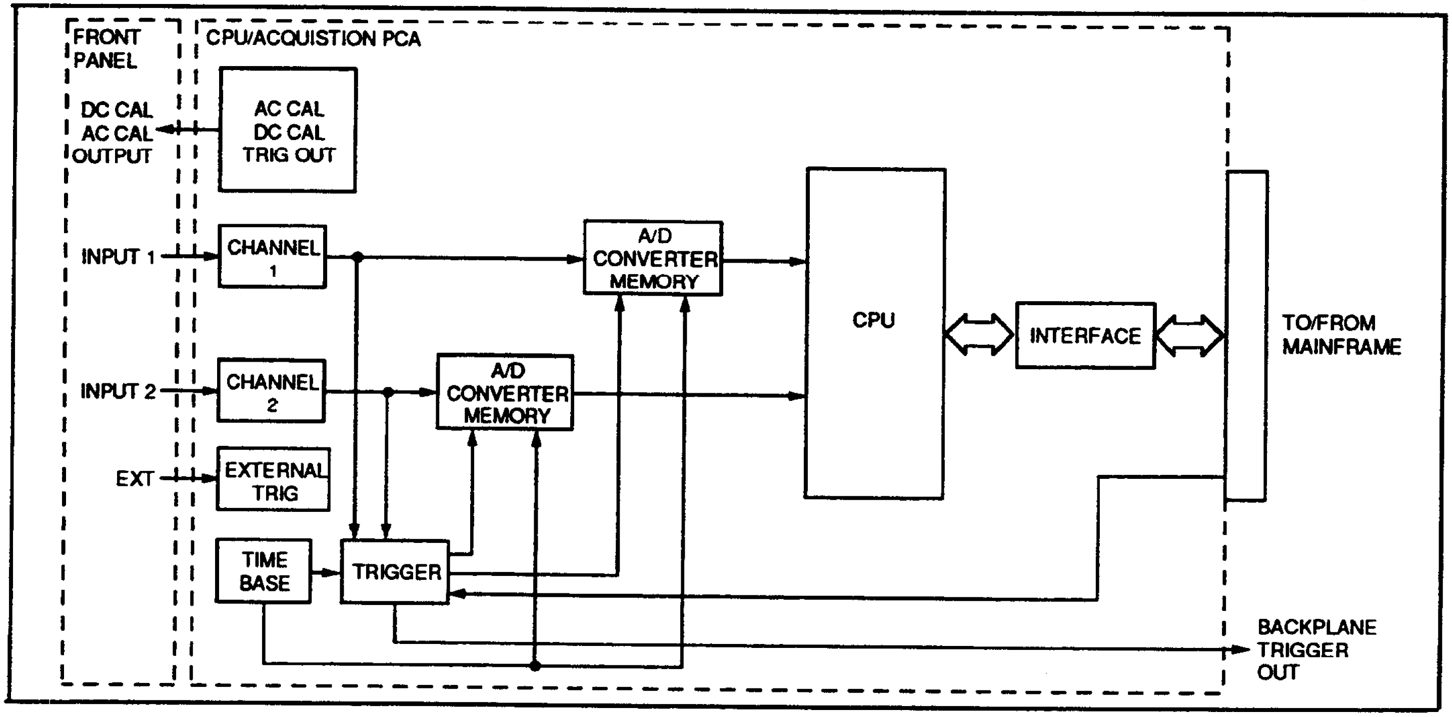


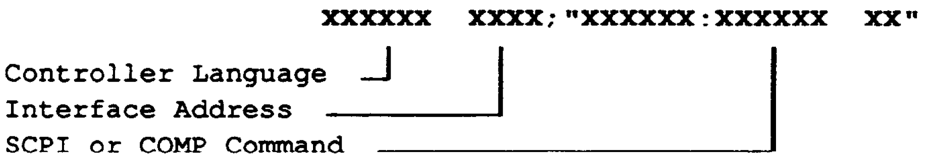
Figure 1-1. Oscilloscope Module Block Diagram

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 E1428A is unique in that it can be operated using two different programming languages.

- The Hewlett-Packard 54510A Compatible Language (COMP) is compatible with the Hewlett-Packard 54510A 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 mid-stream. Each language has certain advantages as follows:

The advantages of using COMP are:

- Allows use of existing HP 54510A programs.
- All the features of the instrument can be used (e.g. all five triggering modes).
- Allows the HP 54510A to be used as a test program development tool. (If a problem is encountered with a program for the HP E1428A, the HP 54510A can simplify program debugging by adding local operation capabilities and a display.)

The advantages of using SCPI are:

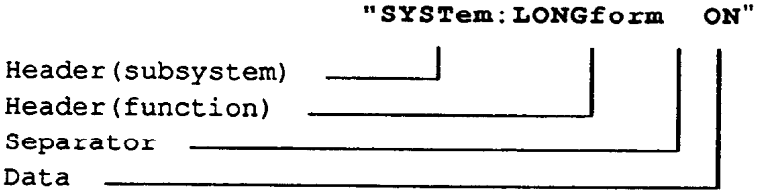
- Shorter learning curve for the user who will be programming numerous SCPI compatible instruments.

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

- COMP to HP 54510A 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 (e.g., 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
```

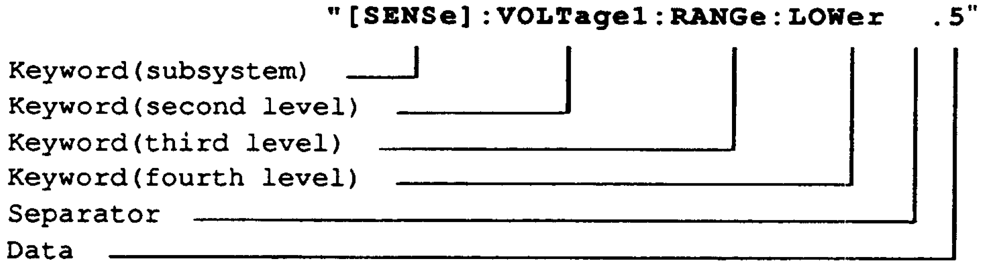
The same command shown without the lower case letters is the short form. The command then becomes:

```
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 (e.g., 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

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.

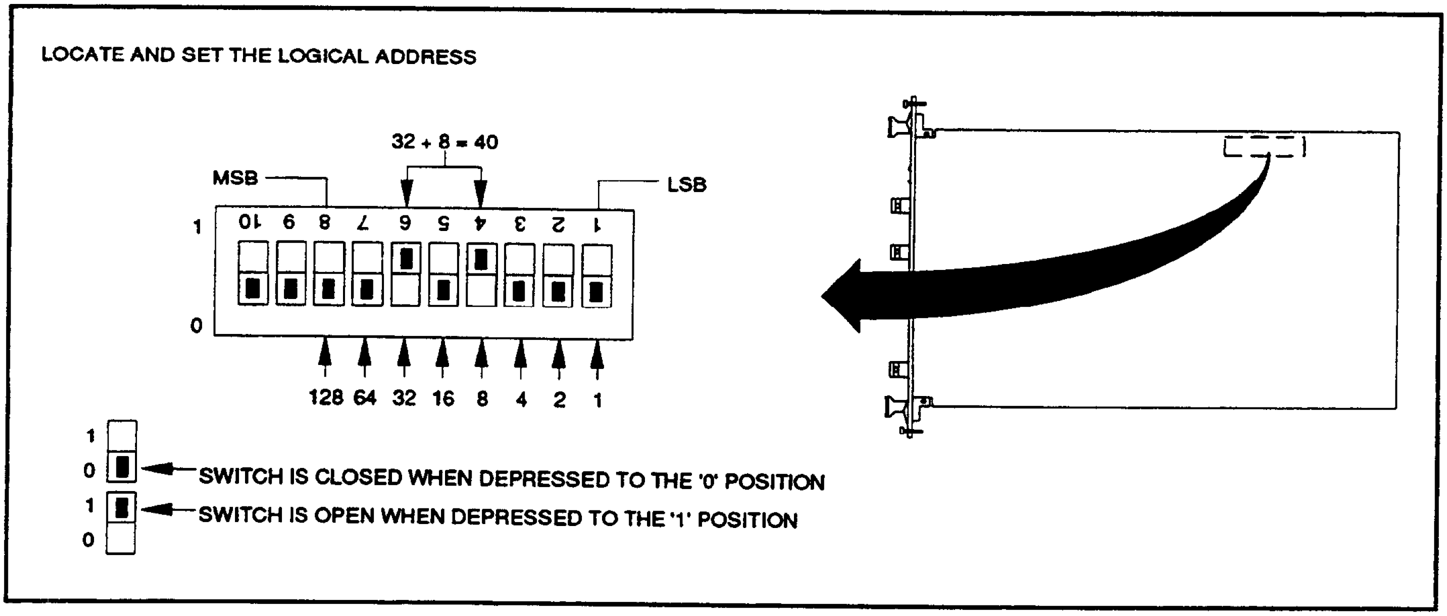

LOCATE AND SET THE LOGICAL ADDRESS
32 + 8 = 40
MSB
LSB
1
0
10 9 8 7 6 5 4 3 2 1
128 64 32 16 8 4 2 1
1
0
SWITCH IS CLOSED WHEN DEPRESSED TO THE '0' POSITION
1
0
SWITCH IS OPEN WHEN DEPRESSED TO THE '1' POSITION

Figure 2-1. Address Selection

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 two front panel module retaining screws must be screwed in completely.

Caution

MAXIMUM INPUT VOLTAGE. The maximum voltage that can be applied to the two 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.

Protecting Non-Volatile RAM

The Oscilloscope module is shipped from the factory with the Non-Volatile RAM that stores the calibration information for the oscilloscope in the Unprotected mode. This configuration should be suitable for most user's, as the calibration information can only be changed during calibration (loading default data, performing and loading new calibration information, etc).

Should it become necessary to limit the user's that can modify this calibration data to those who know a modifiable password:

- Locate and remove R208 (clip both leads). It may be necessary to remove the top shield by removing eight Torx screws and loosening the four front panel BNC connector nuts.
- Power-up the oscilloscope.
- Set the non-volatile protect mode to ON using the SYSTem:NVPRotect ON,SYSTEM command (see Chapter 4, SYSTem subsection, for more information), where "SYSTEM" is the factory default password.
- If desired, the factory default password can be changed using the SYSTem:NVPRotect:PASSword <old>,<new> command (see Chapter 4, SYSTem subsection, for more information).

Note

The old password must be specified to change to a new password. If the password is unknown, replace the 0Ω resistor in R208 (CAL PROTECT) space on the printed wiring board, then powering up the module will set the factory default password to "SYSTEM" and protect state to OFF.

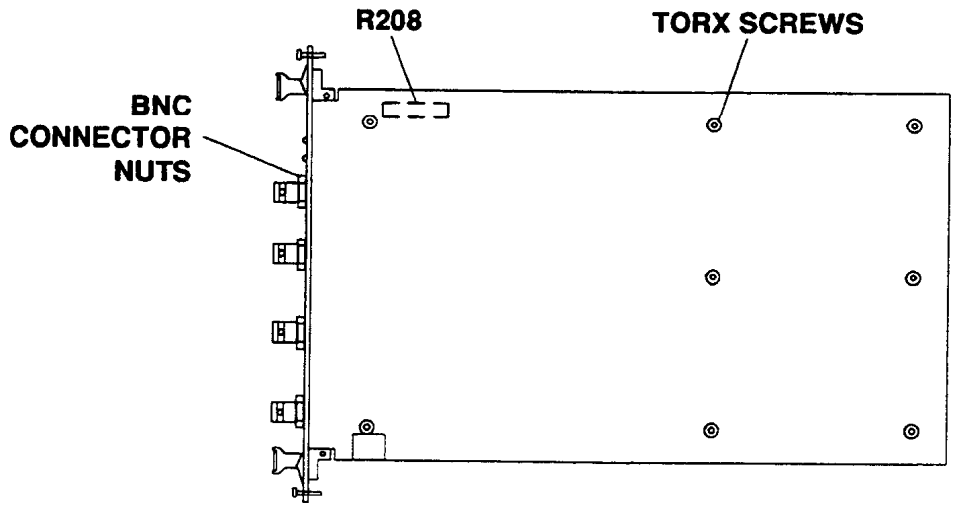


Figure 2-3. Protecting Non-Volatile RAM

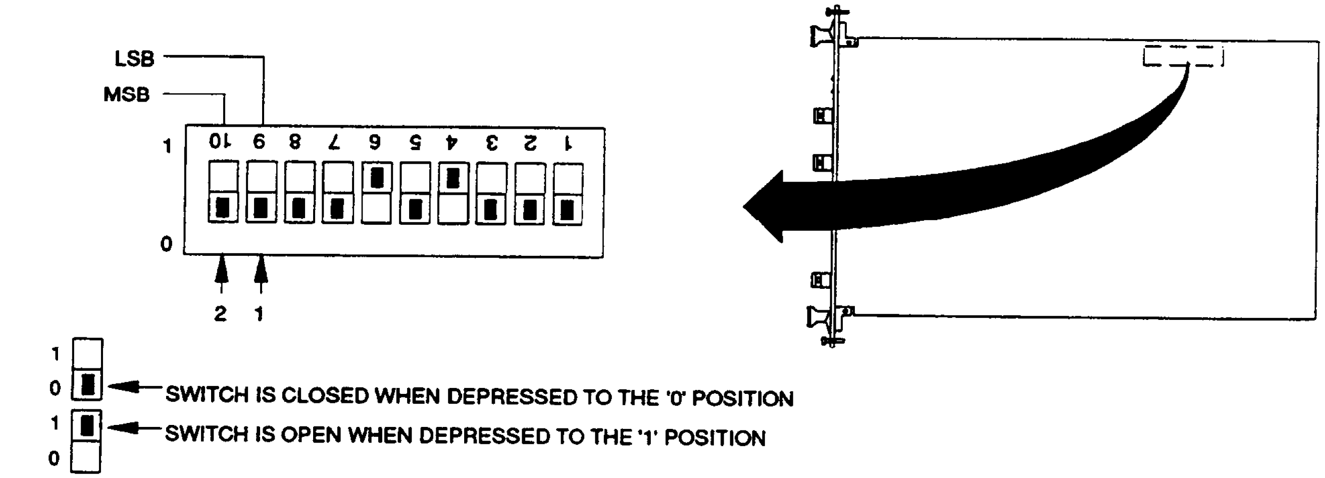
Selecting the Bus Request/Grant Level

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



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

Contents

Chapter 3 — Using the Oscilloscope with HP 54510A Compatible Commands

3

Using the Oscilloscope with HP 54510A Compatible Commands

Using This Chapter

This chapter uses typical examples to show how to use the Oscilloscope module using COMP (Hewlett-Packard 54510A 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 1-2.
:PROBe	Select the input probe attenuation factor.
:RANGe	Set the full scale vertical range.
DIGitize	Digitize waveform data.

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

Command	Description
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.
:DEFine	Define measurement parameters.
:DELay?	Perform a delay measurement and return the results.
: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.
STORe	Store waveform in waveform memory.
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.
:SAMPle	Set sample mode to realtime or repetitive.
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 a measurement, 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	DCCAL	DC CAL on.
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, channel 2 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.
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.
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%.

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

Parameter	Reset	Description
MEASure: (Cont'd)		
WCOMpare:		
ALLowance	0	Allows 0 DIV test deviation.
COMPare	CHAN1,WMEM1	Channel 1 compared to WMEMory1/2.
DESTination	OFF	Destination for violation to OFF.
WTESt	OFF	Waveform Comparison test OFF.
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.
SAMPle:	REALtime	Realtime mode.
CLOCk:	INT	Internal sample clock selected.
IMPedance	DC	Connector input impedance to DC.
LEVel	0	Clock level at 0 V.
TRIGger:		
COUPling	DC	Input at DC 1MΩ.
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.
BYTeorder	MSBFirst	Data read MSB to LSB.
SOURce	CHAN1	Channel 1 source for waveform commands.
OUTPut:	OFF	Output trigger 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	6AD14H	External memory size in bytes (hex).
STATe	OFF	External memory 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, two, and external using the following procedure:

1. Set the Non-volatile RAM protection mode to OFF (if the CAL PROTECT resistor (R208) has been removed).

Note

If R208 has been removed, a <password> parameter is required when changing Non-volatile RAM protection mode. <password> default is "SYSTEM" from factory. See Chapter 4, SYSTem Subsystem for more information.

2. Connect the Oscilloscope Probe Comp/Cal/Trig Output connector to the Input 1, 2, and EXT TRIG 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 no more than 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.

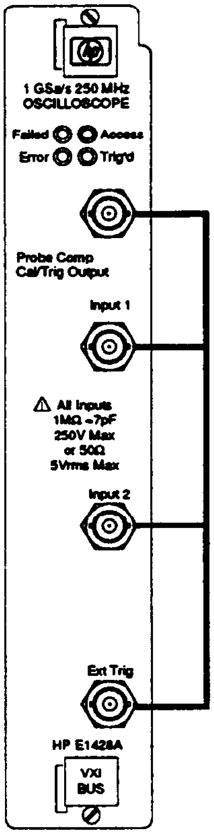


Figure 3-1. Example: Vertical Calibration Setup

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 and two (one at a time) using the following procedure:

1. Verify the Vertical Cal Procedure has been performed.
2. Connect the Oscilloscope Probe Comp/Cal/Trig 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. When complete, disconnect BNC cable and perform Time Null Calibration Procedure.

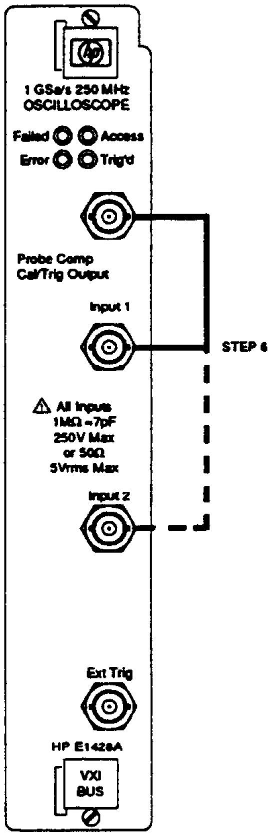


Figure 3-2. Example: Delay Calibration Setup

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 2                        Input testing loop.
 50 OUTPUT 70905;"CAL:SCAL:DEL CHAN"&VAL$(I)
                                        Selects input n delay calibration
                                        routine (n=input 1-2).
 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 input 2.
120 END                                 Terminate program.
```

Time Null Cal Procedure

Time Null calibration is performed on channels one to two, and one to EXT TRIG using the following procedure:

1. Verify the Vertical Cal and Delay Cal Procedures have been performed.
2. Connect the Oscilloscope Probe Comp/Cal/Trig 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 EXT TRIG. Repeat steps 4 and 5 for external trigger.
7. When complete, disconnect BNC cable and perform the Logic Trigger Calibration Procedure (if necessary).

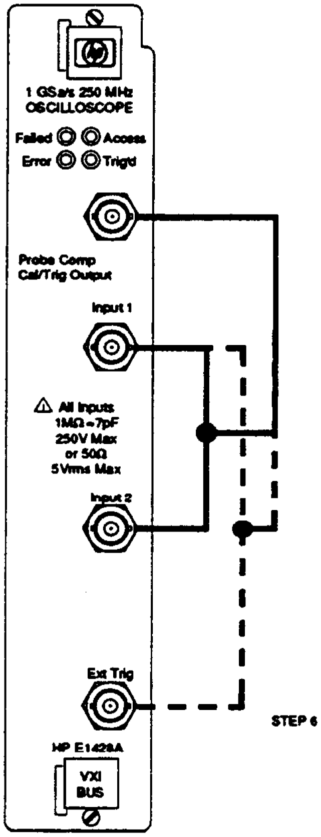


Figure 3-3. Example: Time Null Calibration Setup

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 OUTPUT 70905;"CAL:SCAL:TNUL CH1TO2"
                                     Selects channel 1 to 2 time null
                                     calibration routine.
 50 OUTPUT 70905;"CAL:SCAL:BCAL"     Starts time null calibration routine.
 60 OUTPUT 70905;"SUMM:QUES:CAL?"    Read calibration event register.
 70 ENTER 70905;C                    Enter calibration event register results.
 80 PRINT C                          Print calibration event register results.
 90 PAUSE                            Connect cable to EXT TRIG.
100 OUTPUT 70905;"CAL:SCAL:TNUL CH1TOEXT"
                                     Selects channel 1 to EXT time null
                                     calibration routine.
100 OUTPUT 70905;"CAL:SCAL:BCAL"     Starts time null calibration routine.
110 OUTPUT 70905;"SUMM:QUES:CAL?"    Read calibration event register.
120 ENTER 70905;D                    Enter calibration event register results.
130 PRINT D                          Print calibration event register results.
140 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, and protection of the calibration data in non-volatile RAM is desired, set the Non-volatile RAM protection mode to ON (SYST:NVPR ON,<password>).

1. Verify the Vertical Cal, Delay Cal, and Time Null Cal procedures have been performed.
2. Connect the Oscilloscope Probe Comp/Cal/Trig 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. Select and start the logic trigger calibration routine. Verify that the TRIG'D LED flashes during calibration.

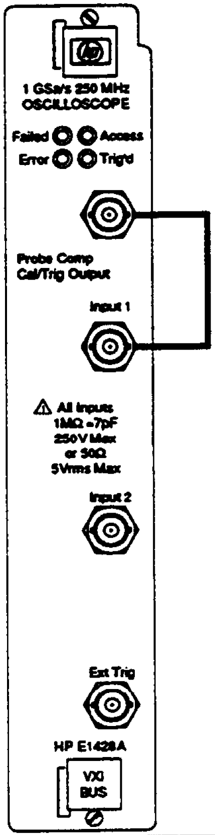


Figure 3-4. Example: Logic Trigger Calibration Setup

6. 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.
7. Disconnect BNC cable and set the Non-volatile RAM protection mode to ON (if desired and resistor R208 has been removed).

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;"CAL:SCAL:LTC"      Selects logic trigger calibration
                                     routine.
50  OUTPUT 70905;"CAL:SCAL:BCAL"     Starts logic trigger calibration
                                     routine.
60  OUTPUT 70905;"SUMM:QUES:CAL?"    Read calibration event register.
70  ENTER 70905;E                    Enter calibration event register
                                     results.
80  PRINT E                          Print calibration event register
                                     results.
                                     Calibration data to protected
                                     (optional).
90  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 and 2, 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 (must be in AC coupled mode (CHANn:COUP AC):

```
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:MODE xxx    (xxx = TRIG, SING, or AUT)
```

Range

Full scale (not per division) horizontal axis can be entered from 10 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:MODE xxx    (xxx = EDGE, 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 1-2), from the external trigger signal (EXT TRIG), or from one of two bus lines (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, EXT, 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 (inputs1-2), 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 = CHAN1-2, FUNC1-2, or WMEM1-4)
```

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

Four 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, ENVelope, and RAWData modes. NORMal is selected at reset. You select acquisition type with the following command:

```
ACQ:TYPE xxx    (xxx = NORM, AVER, ENV, RAWD)
```

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 (from 1 to 2048) to be averaged for each acquisition can be entered (Average mode), or the number of times a bucket is hit (Envelope mode). Count is not effective in Raw Data mode. 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 depending on the sample mode. During repetitive mode, 500 is the only acceptable value. During realtime mode, 500 and 8000 are acceptable. 500 is selected at reset. You enter points with the following command:

```
ACQ:POIN xxx    (xxx = 500 or 8000)
```

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 = CHAN1-2)
```

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 1-2, 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 = CHAN1-2, 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 500 points have been averaged at least four times. The digitized data sent to the controller is to be in byte format.

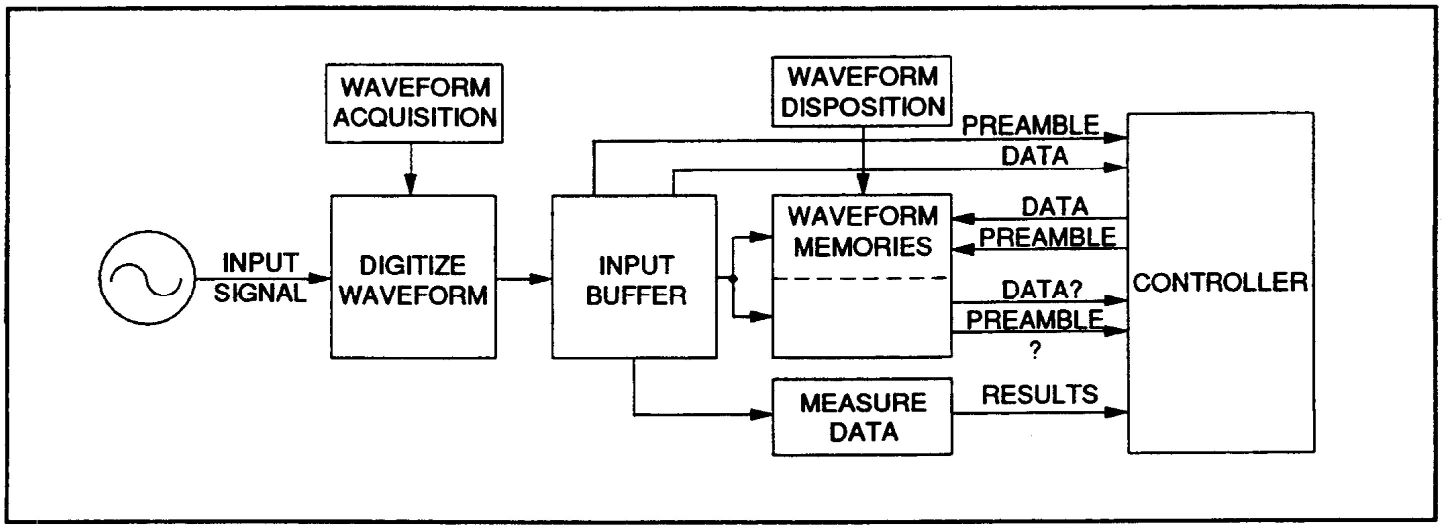


Figure 3-5. Example: Digitizing Waveforms

This example is written using:

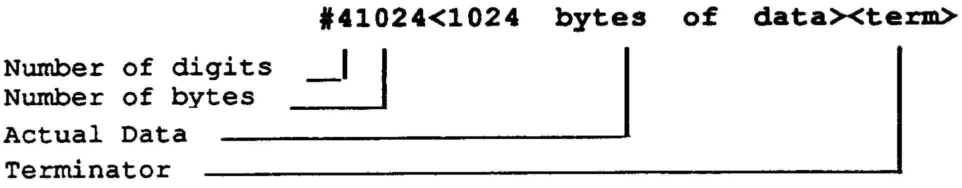
- an HP-IB select code of 7, primary address of 09, and secondary address of 05 for the Oscilloscope
- an HP Series 200/300 Computer with HP BASIC

Execute:

`10 DIM Pre$[100]`	*String for preamble data.*
`20 INTEGER Waveform(2000)`	*Temporary dimension for waveform data.*
`30 OUTPUT 70905;"SYST:HEAD ON"`	*System headers to on.*
`40 OUTPUT 70905;"SYST:LONG ON"`	*System headers to long form.*
`50 OUTPUT 70905;"*CLS"`	*Clear status.*
`60 OUTPUT 70905;"*RST"`	*Resets the Oscilloscope to its default state (table 3-2).*
`70 OUTPUT 70905;"AUT"`	*Perform an autoscale.*
`80 OUTPUT 70905;"ACQ:TYPE AVER"`	*Acquisition type to average.*
`90 OUTPUT 70905;"ACQ:COUN 4"`	*Number of averages to 4.*
`100 OUTPUT 70905;"ACQ:COMP 100"`	*Completion criteria to 100%.*
`110 OUTPUT 70905;"ACQ:POIN 500"`	*Data record to 500 points.*
`120 OUTPUT 70905;"DIG CHAN1"`	*Digitize channel 1 and place data in channel output buffer.*
`130 OUTPUT 70905;"WAV:SOUR CHAN1"`	*Waveform source is channel 1.*
`140 OUTPUT 70905;"WAV:FORM BYTE"`	*Format for waveform data is byte.*
`150 OUTPUT 70905;"WAV:PRE?"`	*Read preamble data.*
`160 ENTER 70905;Pre`	*Enter preamble data.*
`170 OUTPUT 70905;"WAV:DATA?"`	*Read waveform data.*
`180 GOSUB Get_data`	*Go to a subroutine that will read the header to determine the size of the waveform data, then re-dimension the waveform data array.*
`190 STOP`	*Stop main program.*
`200 Get_data: !`	*Data retrieved routine.*
`210 ENTER 70905 USING "#,1A";One_char$`	*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:



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., 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. Continuous measurement mode will increase digitize times since the oscilloscope must perform the measurement after the acquisition has completed.
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
- Jitter Measurement
- Using Delay Trigger
- Limit Testing

Caution

MAXIMUM INPUT VOLTAGE. The maximum voltage that can be applied to the two 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 two 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.

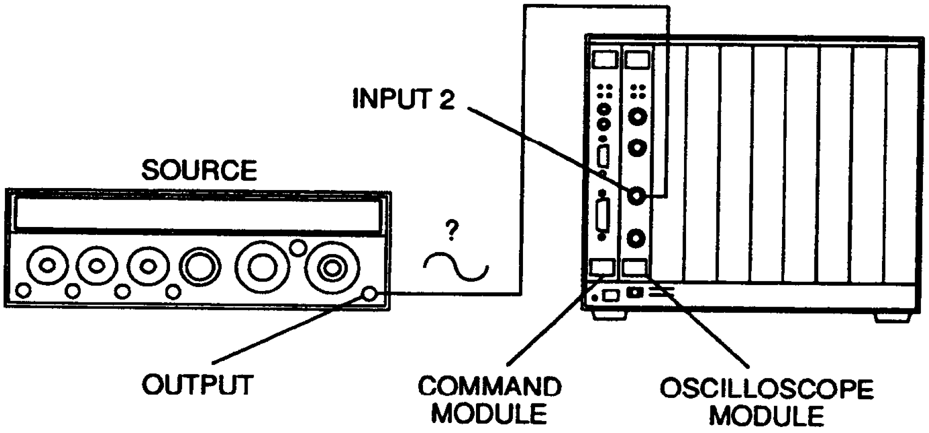


Figure 3-6. Example: Autoscale Frequency Measurement

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.

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 1 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.

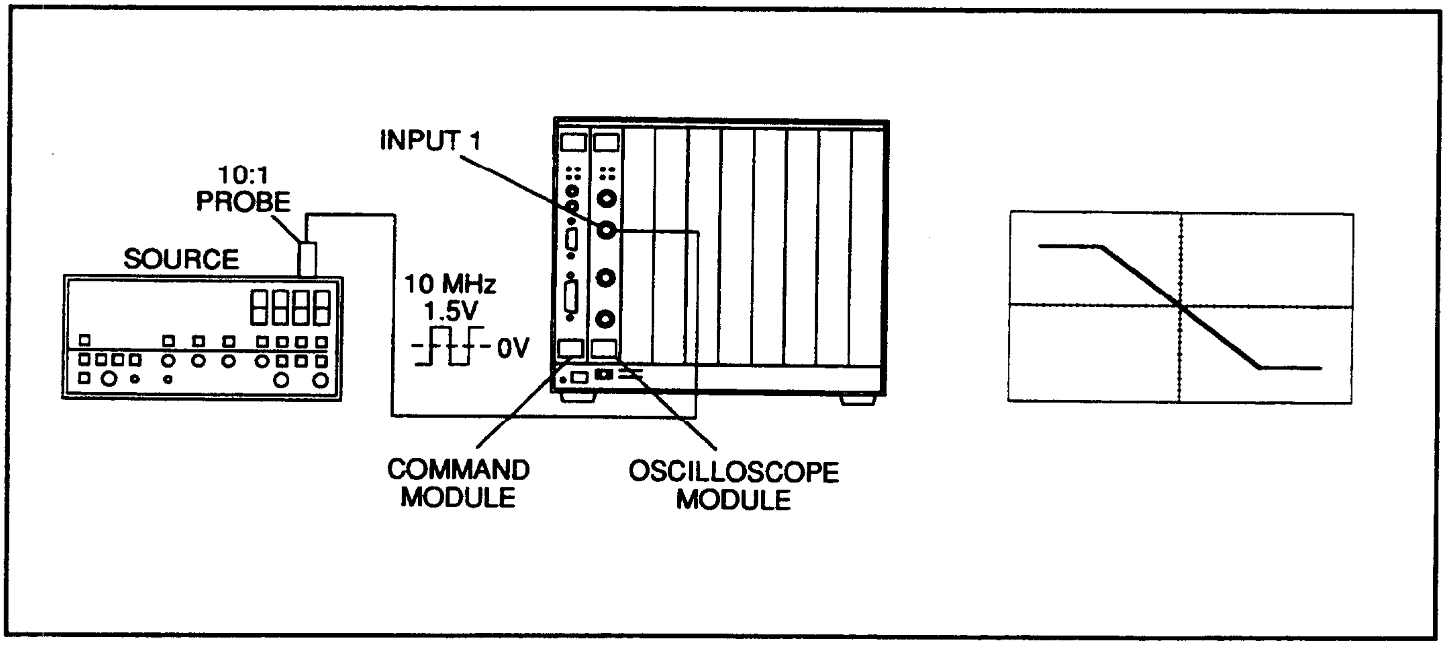


Figure 3-7. Example: Manual Fall Time Measurement

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;"CHAN1:PROB 10"`	*Set input 1 probe attenuation factor at 10:1.*
`50 OUTPUT 70905;"CHAN1:RANG 2"`	*Set input 1 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 CHAN1"`	*Trigger source to channel 1.*
`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 CHAN1"`	*Digitize channel 1 using default acquisition parameters (see table 3-2).*
`110 OUTPUT 70905;"MEAS:SOUR CHAN1"`	*Measure channel 1.*
`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.

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: Jitter Measurement

This example uses the Oscilloscope module to measure jitter of a signal connected to input 1 using a 1MΩ 10:1 probe. The expected input is a 1.5 V clock at 10 MHz.

Notes

When measuring jitter, the leading (positive-going) edge of the waveform must be present.

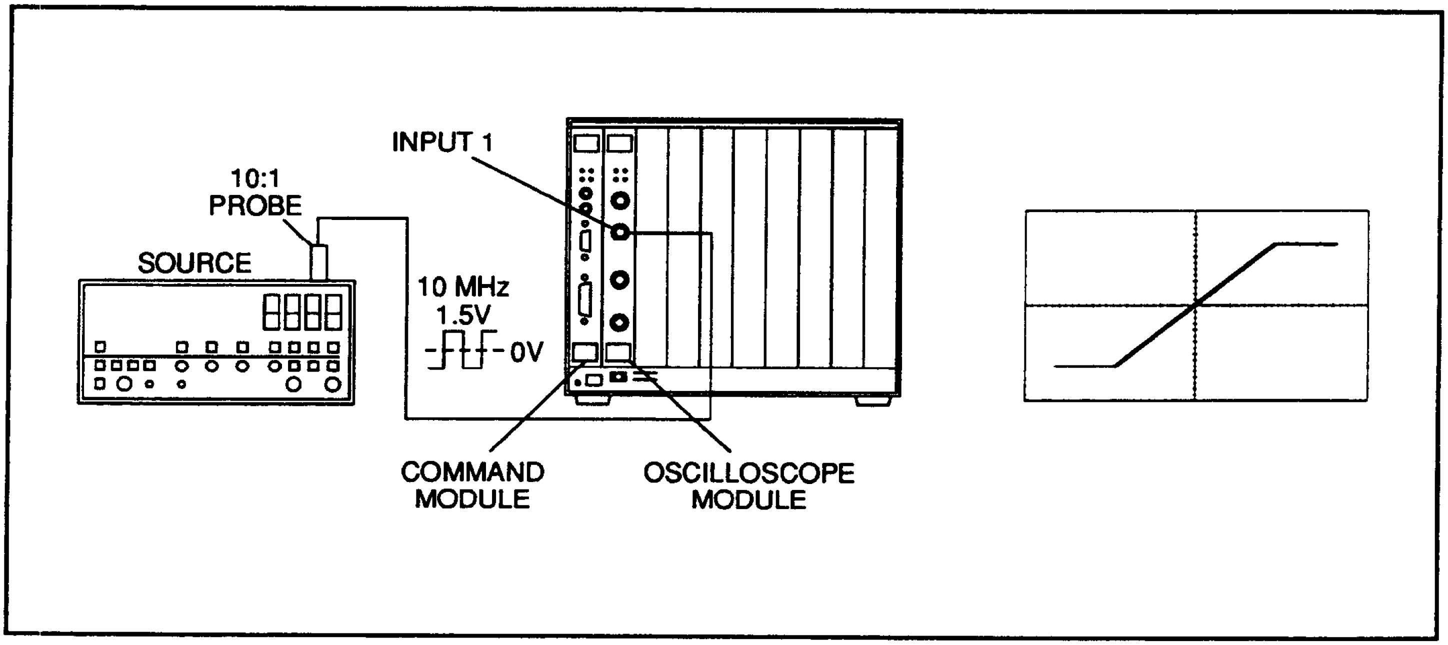


Figure 3-8. Example: Jitter Measurement

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;"CHAN1:PROB 10"`	*Set input 1 probe attenuation factor at 10:1.*
`50 OUTPUT 70905;"CHAN1:RANG 2"`	*Set input 1 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 CHAN1"`	*Trigger source to channel 1.*
`80 OUTPUT 70905;"TRIG:SLOP POS"`	*Trigger slope to positive (to ensure leading edge is acquired).*
`90 OUTPUT 70905;"TRIG:LEV .75"`	*Trigger level to .75 volts (approximately one-half the expected input).*
`100 OUTPUT 70905;"TIM:SAMP REP"`	*Sample mode to repetitive.*
`110 OUTPUT 70905;"AQU:TYPE ENV"`	*Acquisition type to envelope (for jitter measurement).*

```
120  OUTPUT 70905;"DIG CHAN1"               Digitize channel 1 using default
                                            acquisition parameters (see table
                                            3-2).
130  OUTPUT 70905;"STOR CHAN1,WMEM1"        Store measured waveform in
                                            waveform memory pair 1/2
                                            (envelope mode).
140  OUTPUT 70905;"MEAS:SOUR WMEM1,WMEM2"
                                             Measure waveform memory pair
                                            1/2.
150  OUTPUT 70905;"MEAS:DEF DEL,POS,1,MIDD,POS,1,MIDD"
                                            Define as delay measurement
                                            from first positive edge midpoint
                                            to first positive edge midpoint.
160  OUTPUT 70905;"MEAS:DEL?"               Perform delay measurement  and
                                            return results.
170  ENTER 70905;Results                    Enter measurement results.
180  PRINT Results                          Print jitter measurement results.
190  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.

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: 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.

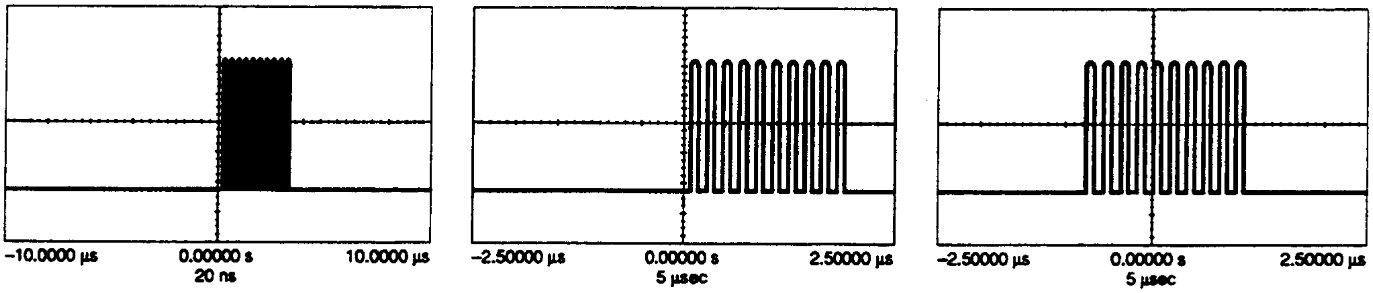


Figure 3-9. Example: Using the Delay Trigger

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:MODE DEL"        Delay trigger mode.
 50 OUTPUT 70905;"TRIG:QUAL:EDGE"       Qualify mode to edge.
 60 OUTPUT 70905;"TRIG:DEL TIME,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.

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.

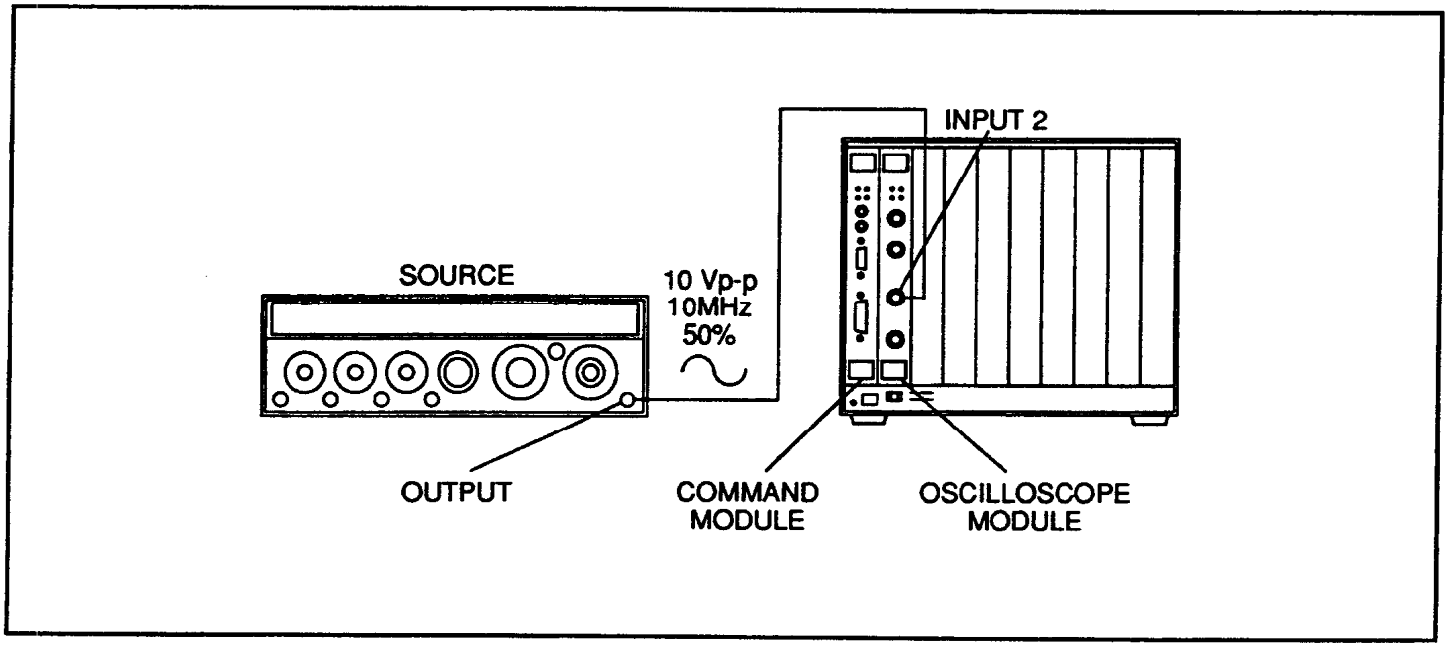


Figure 3-10. Example: Limit Testing

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.
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 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 measurements, on up to two different channels at the same time by stepping through the MEASure:SOURce parameter. Only one channel can be specified as the destination (using the MEASure:DESTination 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 controls using the signal connected closest to channel 1. If a signal is not found on input 1 then input 2 is used.

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 (1-48) 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:ECL, or CHANnel:TTL commands.

- Enter the number (1-48) 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

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, and BLANk.

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 1-2 - active waveform from input 1-2

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 CHAN1-2, WMEM1-
                     4, or FUNC1-2 and yyyy= memory
                     destination  WMEM1-4)
```

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 1-2 - active waveform from input 1-2
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 = CHAN1-2, 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 1-2 - active waveform from input 1-2

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 = CHAN1-2, WMEM 1-4, or FUNC 1-2)
```

Note

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

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 or 2).

Query	Description
ACQuire	Acquire subsystem queries
:COMPlete?	Acquisition complete value
:COUNt?	Acquisition count value
:POINts?	Acquisition points value
:TYPE?	Acquisition type
BNC?	AC Cal, DC Cal, Triggers output
CALibrate	Calibrate subsystem queries
:REPort? CHANnel<n>	Calibration report
:TNULl?	Current time null values for channel pair 1-2.
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
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
: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

Querying the Oscilloscope — Continued

Unless otherwise specified, <n> is the channel number (1 or 2).

Query	Description
MEASure—Continued	Measure subsystem queries
: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
:WCOMpare	Waveform comparison subsystem queries
:ALLowance?	Allowable divisions selected
:COMPare?	Channel and waveform memories selected
:DESTination?	Destination for limit comparison violation
:POSTfailure?	Comparison post-failure instruction
MEMory	Memory subsystem queries
:VME	
:ADDRess?	Memory address selected
:MAP?	Measurement result address and size
:SIZE?	Memory size
:STATe?	Overall VME memory state

Querying the Oscilloscope — Continued

Unless otherwise specified, <n> is the channel number (1 or 2).

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
POWerup?	Powerup state
SUMMary	Summary subsystem commands
:QUEStionable?	Questionable event register value
:CALibration?	Calibration event register value
:CHANnel<*n*>|EXTernal?	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 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
:NVPRotect?	Non-volatile RAM protect mode
:SETup?	Setup data (block)

Querying the Oscilloscope — Continued

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

Query	Description
RUN?	Current acquisition state
STATus? <xxxx>	Channel, function, or waveform 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
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
:BYTeorder?	Order bytes are read
: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 54510A Compatible Language Command Reference

4

Oscilloscope HP 54510A Compatible Language Command Reference

Using This Chapter

This chapter describes Hewlett-Packard 54510A Compatible Language (COMP) commands and summarizes IEEE 488.2 Common (*) commands applicable to the Oscilloscope module.

See the HP 54510A 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 54510A 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:

```
SUMMary
        :PRESet
        :QUEStionable
                :CONDition?
```

SUMMary is the root command, :PRESet and :QUEStionable are the second level sub commands, and :CONDition is a third level query.

Command Separator

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

```
SUMMary:QUEStionable:CONDition?
```

Colons separate the root command from the second level command (SUMMary:QUEStionable), and the second level from the third level query (QUEStionable:CONDition?).

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 54510A 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 fo lows:

  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 for repetitive acquisition. 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, in AVERage mode, for each time bucket before the acquisition (for that time bucket) is considered complete. In ENVelope mode, count selects the number of times the time bucket is hit.

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

- **TIMebase:SAMPle REPetitive:** ACQuire:COUNt is only used in REPetitive mode.
- **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. When operating in the repetitive mode, the only acceptable value is 500. When operating in the real-time mode, acceptable values are 500 or 8000. TIMebase:SAMPle sets mode.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
points	numeric	500|8000	none

Example

Set acquire points to 500

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  500                Acquire points are 500
```

Comments

- **Entering Points:** Any value between 0 and 1023 can be entered, however entry will be adjusted to 500. Any value entered that is >1024 will be adjusted to 8000. Values entered >1024 in repetitive mode will be adjusted to 500.
- **Waveform Points:** To determine the ACTUAL number of time buckets acquired, send the WAVeform:POINts? query.
- **Related Commands:** TIMebase:SAMPle, 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  500                Acquire points are 500
ACQ:POIN?                    Query instrument to return acquire
                             points value
enter  statement             Enter value into computer
```

:TYPE

ACQuire:TYPE *<mode>*[,*<length>*][,*<acquisitions>*] selects the type of acquisition that will take place when a DIGitize command is executed. When RAWData is selected, the length and number of acquisitions can be specified.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	NORMal|AVERage| ENVelope|RAWData	none
length	numeric	4 to 8000	none
acquisitions	numeric	See Determining Acquisitions below	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:TYPE   AVER          Acquisition type is average
```

Comments

- **Selecting Mode:** Mode is used to select how the acquisitions are used when generating the waveform. See the ACQuire: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. *length* and *acquisition* are not entered in this mode.

  AVERage: Used when reduction of signal noise and improved resolution is desired in repetitive timebase mode. The waveform reflects a minimum of *n* acquisitions averaged per time bucket, where *n* is the current ACQuire:COUNt value. *length* and *acquisition* are not entered in this mode.

  ENVelope: Used when measuring voltage or time jitter in repetitive timebase mode. The waveform reflects the minimum and maximum data points (hit) in each time bucket. *length* and *acquisition* are not entered in this mode.

  RAWData: Used to return 16-bit binary data over the bus (in WORD format). When selected, the number of points for each acquisition (*length*), and number of acquisitions to be taken in a single DIGitize operation (*acquisition*) are specified. Data cannot be stored in waveform memories. WAVeform:FORMat have no effect in this mode. This mode is exited by selecting another ACQuire:TYPE.

- **Entering Length:** In RAWData mode, the number of points for each acquisition (from 4 to 8000) can be specified as *length*. This optional parameter can be entered with or without specifying *acquisition*.

To receive a correct WAVeform:PREamble? with the corresponding 8000 point realtime acquisition, send the DIGitize or STOP command before sending the WAVeform:DATA? command

- **Determining Acquisition:** In RAWData mode, the maximum number of acquisitions taken during a DIGitize operation are determined by the *length* specified, and the limited buffer size available (200,000 bytes per channel). The number of acquisitions must satisfy the following condition:

  buffer_size = [(*length2) +8]**acquisitions***

  Where:
  length (entered) is from 4 to 8000
  buffer_size = 200,000 (maximum)

Example

Determine how many acquisitions can be made on channel 1 at 500 points (length) as follows:

200,000 = [(2*500) + 8] * acquisitions

$$\text{acquisitions} = \frac{200\,000}{1008} = 198$$

Each data point takes 2 bytes, and the time value (WAVeform:XORigin) associated with each acquisition takes 8 bytes. If this optional parameter is used, *length* must also be specified.

Note

In shared memory, the buffer size is determined by MEMory:VME:SIZE.

- **Related Commands:** ACQuire:COUNt, DIGitize, WAVeform:TYPE?, WAVeform:FORMat, and WAVeform:PREamble, MEMory:VME:SIZE
- ***RST Condition:** Defaults to NORMal.

:TYPE?

ACQuire:TYPE? returns the currently selected acquisition type, and if applicable, length and number of acquisitions. 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:TYPE  RAWD           Acquisition type is raw data
ACQ:TYPE?                Query instrument to return
                         acquisition type, length, and number
                         of acquisitions (as required)
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
            :BCALibration
            :CHANnel<number>
        :TNULl <time>
    :REPort? <channel>
    :SCALibration
        :BCALibration
        :DCALibration
        :DELay <channel>
        :DOUTput <level>
        :LTCalibration
        :TNULl <channel_skew>
        :VERTical
    :TNULl <value>
    :TNULl?
```

:PCALibration:ATTenuation:BCALibration

CALibrate:PCALibration:ATTenuation: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 Probe Comp/Cal/Trig 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 2**

This example calibrates the channel gain on input 2. For the example, a 10:1 attenuator probe is connected to the Probe Comp/Cal/Trig Output connector from the Input 2 connector.

`CAL:PCAL:ATT:CHAN2`	*Attenuation calibration channel 2*
`pause`	*To connect probe to Probe Comp / Cal / Trig Output from Input 2 connector*
`CAL:PCAL:ATT:BCAL`	*Perform attenuation calibration. Correction automatically stored in CHAN2: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:ATT:BCAL command is executed.

Parameters

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

Example **Set attenuation calibration channel to 2**

```
CAL:PCAL:ATT:CHAN2
```

Attenuation calibration channel to 2

Comments

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

:PCALibration:TNULl

CALibrate:PCALibration:TNULl *<time>* is used to set the timing of channel 2 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
time	numeric	–70NS to +70NS	S

Example **Set time null from channel 1 to 2 to 25 ns.**

```
CAL:PCAL:TNUL 25E-9
```

Channel 1 to 2 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 identical to the CALibrate:TNULl command.
- **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 2)|EXTernal	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	**A/D X, Gain X, Offset X, Hysteresis X, Trigger X, Delay X, Time Null X**
EXTernal	**Hysteresis 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.
- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a SCALibration routine (see SYSTem:NVPRotect).
- **Related Commands:** CAL:SCAL:VERTical, CAL:SCAL:DCALibration, CAL:SCAL:DELay, CAL:SCAL:LTCalibration, CAL:SCAL:TNULl, CAL:TNULl, SUMMary:QUEStionable, SYSTem:NVPRotect.

: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

- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a default calibration routine (see SYSTem:NVPRotect).
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?, SYSTem:NVPRotect.

:SCALibration:DELay

CALibrate:SCALibration:DELay ***<channel>*** performs a delay calibration on both inputs, one at a time. Each input must be connected to the front panel Probe Comp/Cal/Trig Output connector 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 2)	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.
- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a calibration routine (see SYSTem:NVPRotect).
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?, SYSTem:NVPRotect.

:SCALibration:DOUTput

CALibrate:SCALibration:DOUTput *<level>* is used to set the output level of the Probe Comp/Cal/Trig 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 Probe Comp/Cal/Trig Output connector to 5V**

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

Comments

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

: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 calibration results must indicate "P" before the logic trigger calibration can be executed.
- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a calibration routine (see SYSTem:NVPRotect).
- **Calibration Results:** Logic trigger calibration results can be reviewed using the CALibrate:REPort? query.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?, SYSTem:NVPRotect.

: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|CH1TOEXT	none

Example

Chapter 3 contains an example on performing a time null calibration

Comments

- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a calibration routine (see SYSTem:NVPRotect).
- **Calibration Results:** Time null calibration results can be reviewed using the CALibrate:REPort? query.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?, SYSTem:NVPRotect.

:SCALibration:VERTical

CALibrate:SCALibration:VERTical performs a vertical calibration on all three inputs. 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.
- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a calibration routine (see SYSTem:NVPRotect).
- **Related Commands:** Calibrate:SCALibration:BCALibration, Calibrate:REPort?, SYSTem:NVPRotect.

:TNULl

CALibrate:TNULl *<value>* is used to set the timing of channel 2 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	numeric	–70NS to +70NS	S

Example

Set time null to 25 ns from channel 1 to 2

```
CAL:TNUL  25E-9
```
Channel 1 to 2 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 identical to the CALibrate:PCALibration:ATTenuation:TNULl command.
- **Related Commands:** CAL:PCAL:TNULl.

:TNULl?

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

Example

Querying time nulls

```
CAL:TNUL  25E-9
CAL:TNUL?
enter  statement
```
Set channel 1 to 2 time null to 25 nsec
Query instrument to return time nulls
Enter value into computer

Comments

- **Related Commands:** CAL:PCAL: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, 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 *type*	numeric discrete	1 to 2 AC|DC|DCFifty	None 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.
- **Bandwidth:** AC coupling corner is 7 Hz.
- ***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 2	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 2	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 2	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 2	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 2	none
mode	boolean	OFF|0|ON|1	none

Example

Enable filter on channel 1

```
CHAN1:LFR 1
```
Channel 1 filter to on

Comments

- **Mode:** Integer values can be substituted for the OFF (0) and ON (1) parameters.
- **CHANnel:COUPling:** CHANnel<*n*>:COUPling must be set to AC before LFReject can be enabled.
- **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 2	none

Example

Querying channel 1 high pass filter state

`CHAN1:LFR 1`	*Channel 1 filter to on*
`CHAN1:LFR?`	*Query instrument to return channel 1 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	numeric	1 to 2	none
value	numeric	Depends on CHANnel<n>:RANGe	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 2	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 2	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 2	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 *range*	numeric numeric	1 to 2 8MV to 40.0V	none 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 2	none

Example

Querying channel 2 full scale range setting

```
CHAN2:RANG  10        Channel 2 range to 10 volts
CHAN2:RANG?           Query instrument to return channel 2
                      range setting
enter  statement      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 2	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.

FUNCtion

The FUNCtion command subsystem defines eight functions that use signals acquired on CHANnels 1 to 2 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).

Note

Functions are only performed on 500 point records.

Subsystem Syntax

```
FUNCtion<number>
    :ADD <source>,<source>
    :DIFF <source>
    :INTegrate <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 2)| 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.

:DIFF

FUNCtion<*number*>:DIFF <*source*> is used to calculate the voltage differences between consecutive points in time divided by the time bucket width Δt of the specified waveform source. 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 2)| WMEMory*n* (*n*=1 to 4)	none

Example

Calculate voltage difference between consecutive points in time on waveform memory 4 and retain results in function 1.

```
FUNC1:DIFF  WMEM4
```

Calculate the voltage differences on 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 differential waveform. Resulting differential waveform can be measured, turned on/off, stored, and/or viewed.
- **Differential Waveform:** Voltage differences between consecutive points in time divided by the time bucket width Δt are determined as follows:

  $d_1=0$

  $$d_n = \frac{c(n) - c(n-1)}{\Delta t}$$

  Where
  d = differential waveform
  c = channel or waveform memory
- **Related Commands:** FUNCtion<*n*>:OFFSet, RANGe, MEASure subsystem, BLANk, STORe, VIEW.

:INTegrate

FUNCtion<*number*>:INTegrate <*source*> is used to calculate the integral of the specified waveform source. 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 2)| WMEMory*n* (*n*=1 to 4)	none

Example

Calculate the integral on waveform memory 4 and retain results in function 1.

```
FUNC1:INT  WMEM4
```

Calculate the integral on 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 integral waveform. Resulting integral waveform can be measured, turned on/off, stored, and/or viewed.
- **Integral Waveform:** Integral is calculated by adding the voltage points multiplied by the time bucket width Δt as follows:

  $$I_n = \sum_{i=0}^{n-1} C_i \, \Delta t$$

  Where
  i = integral
  c = channel or waveform memory
- **Related Commands:** FUNCtion<*n*>:OFFSet, RANGe, MEASure subsystem, BLANk, STORe, VIEW.

: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 *source*	numeric discrete	1 or 2 CHANnel*n* (*n*=1 to 2)| WMEMory*n* (*n*=1 to 4)	none 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 *source*	numeric discrete	1 or 2 CHANnel*n* (*n*=1 to 2)| WMEMory*n* (*n*=1 to 4)	none 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	numeric	1 or 2	none
source	discrete	CHANnel*n* (*n*=1 to 2)| WMEMory*n* (*n*=1 to 4)	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	numeric	1 or 2	none
range	numeric	Depends on CHANnel<n>:RANGe OFFSet, and PROBe	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 2)| WMEMory*n* (*n*=1 to 4)	none none

Example

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

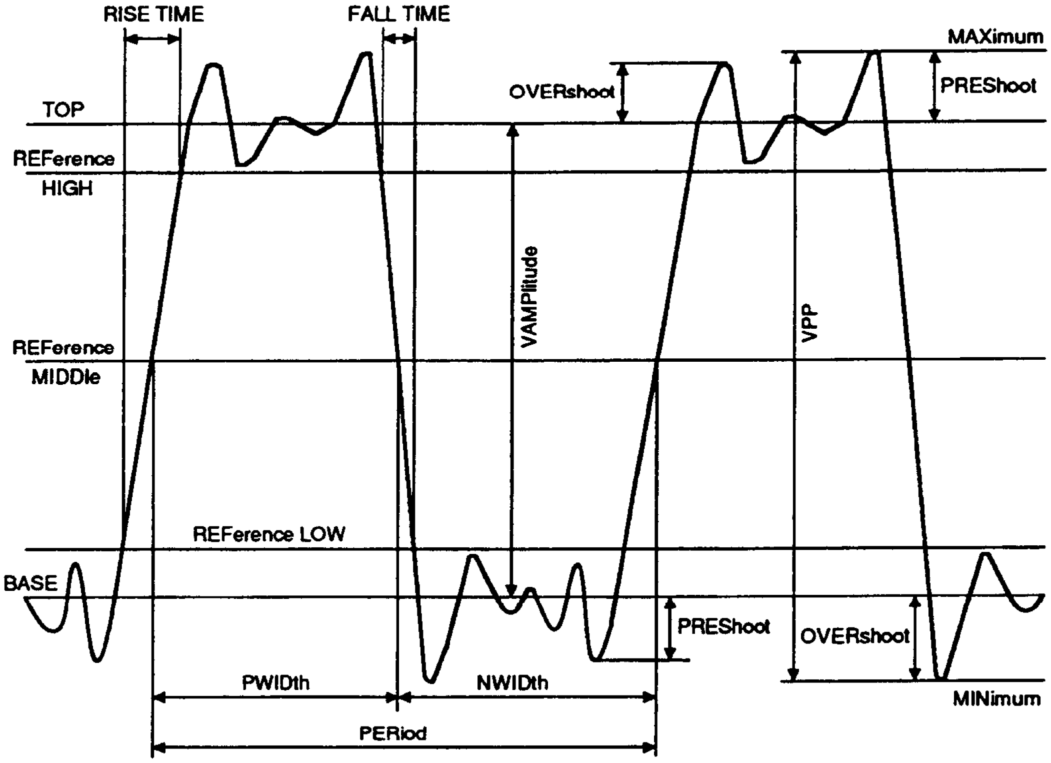
```
FUNC2:SUBT  CHAN2,WMEM1  Subtract waveform memory 1
                         from channel 2, retain as function
                         2
```

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.



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?
    :FREQuency
    :FREQuency?
```

Subsystem Syntax

```
MEASure — Continued
    :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?
    :VDELta?
    :VFIFty
```

Subsystem Syntax

```
MEASure — Continued
    :VMAX
    :VMAX?
    :VMIN
    :VMIN?
    :VPP
    :VPP?
    :VRELative <percent>
    :VRELative?
    :VRMS
    :VRMS?
    :VSTArt <voltage>
    :VSTArt?
    :VSTOp <voltage>
    :VSTOp?
    :VTIMe? <time>
    :VTOP
    :VTOP?
    :WCOMpare
        :ALLowance <value>
        :ALLowance?
        :COMPare <channel>, <memory>
        :COMPare?
        :DESTination <location>
        :DESTination?
        :POSTfailure <mode>
        :POSTfailure?
        :WTESt <mode>
```

: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 2 and return results**

`dimension statement`	*String to hold data*
`MEAS:SOUR CHAN2`	*Select channel 2 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 when making measurements on 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:TYPE 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)|OFF	none

Example

Set destination to waveform memory 4

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

`MEAS:DEST  WMEM4`	*Waveform memory 4 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
```

- **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 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  WMEM4`	*Set destination to waveform memory 4*
`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 1 and return results

`dimension statement`	*String for data*
`MEAS:SOUR CHAN1`	*Measure channel 1*
`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 or space), 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 or space), 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 2 and return results

```
dimension  statement    String for data
MEAS:SOUR  CHAN2        Measure channel 2
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 numeric	–250,000 to +250,000 –25.00 to +125.0	V 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 present UPPer and LOWer threshold levels. Changing LOWer threshold level will effect 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:MODE   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:MODE  USER         Set mode to USER
MEAS:MODE?              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 CHAN 2 & 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 1 and return results**

```
dimension  statement   String for data
MEAS:SOUR  CHAN1       Measure channel 1
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 1 and return results**

`dimension statement`	*String for data*
`MEAS:SOUR CHAN1`	*Measure channel 1*
`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 2)| 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 (CHAN1-2, 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
```

: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 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 2 and return results**

```
dimension  statement   String for data
MEAS:SOUR  CHAN2       Measure channel 2
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 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 2 and return results**

`dimension statement`	*String for data*
`MEAS:SOUR CHAN2`	*Measure channel 2*
`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 (VMarkerl) 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 (Vmarkerl 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 (VMarkerl and VMarker2) are set to the 50% voltage level on that source.

  When two sources are specified, VMarkerl 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 (Vmarker1 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 first example, the current position of Vmarker1 is at the base (0%) of the signal and VMarker2 is at the top (100%).

After execution of this example, VMarker1 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, VMarker1 will move to the 20% level and VMarker2 to the 80% level of the previous setting.

```
MEAS:VREL  80          Move voltage markers 30% from last
                       established position
```

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 VMarker1 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 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.

:WCOMpare:ALLowance

MEASure:WCOMpare:ALLowance ***<value>*** is used to set the allowable number of divisions that the second waveform comparison test can deviate from and still pass.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
value	numeric	0.000 to 8.000	DIV

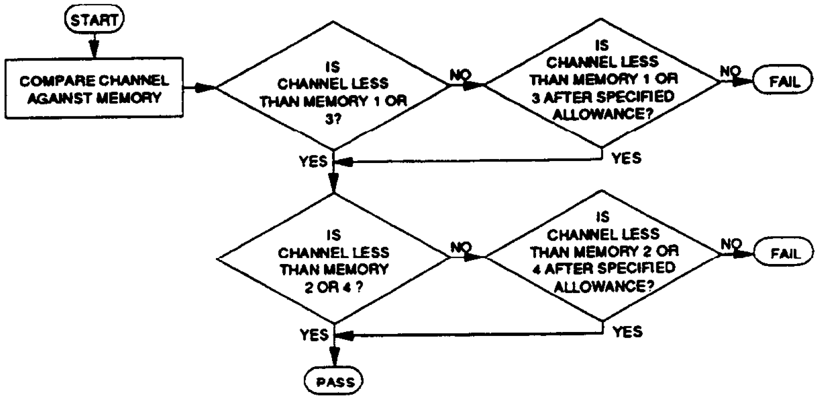
Example **Set second comparison allowance to 1 Division**

```
MEAS:WCOM:ALL  1.0
```
Test deviation to 1 division (or 1/8 range)

See the MEASure:WCOMpare:WTESt command for an example of configuring and performing a comparison test

Comments

- **Comparison Test:** When a waveform comparison test fails, a second test is performed on the waveform data and memory against the specified allowance (tolerance).
- **Range versus Division:** Range must be divided by 8 (total number of vertical divisions) to determine entry in divisions.
- **Entering Value:** The allowance value specified is automatically rounded to the nearest 0.025 divisions.
- **Related Commands:** MEASure:WCOMpare:WTESt?, COMPare, POSTfailure.
- ***RST Condition:** Defaults to 0 (DIV).


START
COMPARE CHANNEL AGAINST MEMORY
IS CHANNEL LESS THAN MEMORY 1 OR 3?
NO
IS CHANNEL LESS THAN MEMORY 1 OR 3 AFTER SPECIFIED ALLOWANCE?
NO
FAIL
YES
YES
IS CHANNEL LESS THAN MEMORY 2 OR 4 ?
NO
IS CHANNEL LESS THAN MEMORY 2 OR 4 AFTER SPECIFIED ALLOWANCE?
NO
FAIL
YES
YES
PASS

:WCOMpare:ALLowance?

MEASure:WCOMpare:ALLowance? returns the number of allowance divisions currently selected for the second comparison test. The value is sent to the output buffer.

Example **Query the second comparison allowance**

`MEAS:WCOM:ALL  1.0`	*Test deviation to 1 division*
`MEAS:WCOM:ALL?`	*Query instrument to return allowance*
`enter  statement`	*Enter data into computer*

:WCOMpare:COMPare

MEASure:WCOMpare:COMPare ***<channel>,<memory>*** is used to select the input channel (1 or 2), and the waveform memory pair (1/2 or 3/4) to be compared during a waveform comparison test.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
channel	discrete	CHANnel1|CHANnel2	none
memory	discrete	WMEMory1|WMEMory2| WMEMory3|WMEMory4	none

Example

Configure a comparison test on channel 1 using waveform memory pair 3/4

See the MEASure:WCOMpare:WTESt command for an example of configuring and performing a comparison test

```
MEAS:WCOM:COMP  CHAN1,WMEM3 Configure comparison test
```

Comments

- **Selecting Waveform Memories:** Memories are selected in pairs, creating a "mask" that the specified input channel will be compared to. Specifying WMEMory1 or WMEMory2 selects both WMEMory1 and WMEMory2 as the mask pair. Specifying WMEMory3 or WMEMory4 selects both WMEMory3 and WMEMory4 as the mask pair.
- **Related Commands:** MEASure:WCOMpare:WTESt?, ALLowance, POSTfailure.

:COMPare?

MEASure:COMPare? ***<measurement>*** returns the two sources currently selected for the compare test. The data is sent to the output buffer.

Example

Querying comparison test configuration

```
dimension  statement            String to hold data
MEAS:WCOM:COMP  CHAN1,WMEM3     Configure comparison test
MEAS:WCOM:COMP?                 Query instrument to return
                                comparison test
                                configuration
enter  statement                Enter data into computer
```

Comments

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

  <channel>,<memory>

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

:WCOMpare:DESTination

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

Parameters

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

Example **Set destination to waveform memory 1**

See the MEASure:WCOMpare:WTESt command for an example of configuring and performing a comparison test

`MEAS:WCOM:DEST WMEM1`	*Waveform memory 1 is the destination for comparison test violations*

Comments

- **Specifying Waveform Memories:** Data may be stored to any of the waveform memories EXCEPT the pair of memories used for the mask in the comparison test (specified using the MEASure:WCOMpare:COMPare command). If the waveform memory selected for the destination is the same as the waveform memories used for the mask, the destination is automatically set to OFF. When a waveform memory is specified, the current contents are overwritten each time a violation is found.
- **Disable Destination:** OFF is specified to disable the destination function.
- **Related Commands:** MEASure:WCOMpare:WTESt.
- ***RST Conditions:** Defaults to OFF.

:WCOMpare:DESTination?

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

Example **Query the violation destination**

`dimension statement`	*String for data*
`MEAS:WCOM:DEST WMEM1`	*Set destination to waveform memory 1*
`MEAS:WCOM:DEST?`	*Query instrument to return destination*
`enter statement`	*Enter data into computer*

:WCOMpare:POSTfailure

MEASure:WCOMpare:POSTfailure ***<mode>*** is used to specify what will occur (comparison test continue or stop) after a violation has been found during a comparison test.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	CONTinue|STOP	none

Example **Set to continue comparison test after violation**

See the MEASure:WCOMpare:WTESt command for an example of configuring and performing a comparison test

```
MEAS:WCOM:POST   CONT
```
Continue comparison test after violation

Comments

- **Selecting Mode:** After a violation (input waveform outside of allowable mask) the comparison test will:

  STOP the waveform comparison test and acquisition.

  CONTinue to look for other violations. If MEASure:WCOMpare:DESTination is not OFF, the violation will be written to the selected memory location, overwriting the previous violation stored in memory.

- **Related Commands:** MEASure:WCOMpare:WTESt, DESTination.
- ***RST Conditions:** Defaults to STOP.

:POSTfailure?

MEASure:WCOMpare:POSTfailure? returns the currently selected failure instructions for comparison test violations. The data is sent to the output buffer. If CONTinue is returned, the comparison test will continue. If STOP is returned, the comparison test will stop.

Example **Query the comparison test failure instruction**

```
dimension  statement     String for data
MEAS:WCOM:POST   CONT    Continue comparison test after
                         violation
MEAS:WCOM:POST?          Query instrument to return post
                         failure instruction
enter  statement         Enter data into computer
```

:WCOMpare:WTESt

MEASure:WCOMpare:WTESt ***<mode>*** used to perform a waveform compare test . The user specifies the acceptable range, and 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

Waveform Comparison Test

In this example, channel 1 data is compared with the mask waveform comprised of waveform memories 3 and 4, and the violation is **saved** in **waveform memory 1.** Test will **stop** after a violation.

```
MEAS:SCR                         Clear measurement queue
MEAS:WCOM:COMP   CHAN1,WMEM3     Configure comparison test
                                 (input channel 1, compare
                                 against waveform mask in
                                 waveform memories 3 and
                                 4
MEAS:WCOM:ALL  1.0               Second test deviation to 1
                                 division
MEAS:WCOM:POST   STOP            Stop comparison test after
                                 violation
MEAS:WCOM:DEST   WMEM1           Waveform memory 1 is the
                                 destination for a test
                                 violation
MEAS:WCOM:WTES   MEAS            Start comparison test
loop  beginning                  Loop beginning
LTER?                            Has limit test failed?
loop  end                    Loop end
MEAS:WCOM:WTES   OFF         Stop comparison test
```

Comments

- **Comparison Test Limitations:**

  MEASure:LIMitest: The comparison test and limit test cannot be performed at the same time.

  ACQuire:POINts: The comparison test runs only on 500 point records, and will not run in the expand mode (8000 points), or on digitized data if the count is not at 500.

- **Comparison Test Sequence:** A comparison test is configured using the following MEASure:WCOMpare subsystem commands:

  ALLowance - is used to set acceptable limits of the second test comparison. See MEASure:WCOMpare:ALLowance command for more information.

  COMPare - is used to set the desired input channel being compared and the waveform memory pairs configuring the comparison mask. See MEASure:WCOMpare:COMPare command for more information.

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

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

- **Comparison Test Pass/Fail Criteria:** To pass the test, the channel data must be less than the first waveform memory data (WMEMory 1 or 3) and greater than the second waveform memory data (WMEMory 2 or 4), plus or minus the value set by the MEASure:WCOMpare:ALLowance command.

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

  LTER? Used to return if the comparison 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.

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.

Subsystem Syntax

```
MEMory
    :VME
        :ADDRess <address>
        :ADDRess? [MINimum|MAXimum]
        :MAP?
        :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 sample mode, acquisition type, and format selected using the following guidelines:

- **TIMebase:SAMPle REPetitive / 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 + #H016C (364 dec)
  INPUT2 (CHANnel2) = MEM:VME:ADDR + #H0940 (2368 dec)

```
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 = data bits. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the WAVeform:PREamble? query.

- **TIMebase:SAMPle REPetitive / ACQuire:TYPE AVERage** — The acquired data is 16 bits in length (stored in a 32 bit word). The starting location, in VME space, of each INPUT buffer is to be calculated as follows:

  INPUT1 (CHANnel1) = MEM:VME:ADDR + #H016C (364 dec)
  INPUT2 (CHANnel2) = MEM:VME:ADDR + #H0940 (2368 dec)

Data — 32 bits

```
                  33222222222211111111110000000000
                  10987654321098765432109876543210
                          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 = data bits. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the WAVeform:PREamble? query.

- **TIMebase:SAMPle REPetitive / 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 + #H016C (364 dec)
  INPUT2 (CHANnel2) = MEM:VME:ADDR + #H0940 (2368 dec)

  MAXIMUM

  INPUT1 (CHANnel1) = MEM:VME:ADDR + #H0556 (1366 dec)
  INPUT2 (CHANnel2) = MEM:VME:ADDR + #H0DA2 (3490 dec)

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 = data bits. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the WAVeform:PREamble? query.

- **TIMebase:SAMPle REALtime / ACQ:TYPE RAWData,** *<points>,<acquisitions>* – The acquired data is 16 bits length. The starting location, in VME space, of each INPUT buffer is calculated as follows:

  SEQUential1 (CHANnel1) = MEM:VME:ADDR + #H9294 (37524 dec)
  SEQUential2 (CHANnel2) = MEM:VME:ADDR + #H39FD4 (237524 dec)

Note: The above calculations are based on the default size of #H6AD14 (437524 dec). If the size is increased the location for SEQUential2 will change due the fact that both SEQ1 and SEQ2 are equally incremented based on available size.

It is recommended for any size to use the MEM:VME:MAP? query to determine exact addresses in VME space.

The data block consists of two arrays. The first array consists of double precision 64-bit floating point numbers. These are the xorigin values of the waveform records to follow. The second array consists of 16-bit integers and represents the actual data for each acquistion.

Double precision format for xorgs:

The most significant bit is the sign bit, the next 11 most significant bits are the exponent field, and the remaining 52 bits are the fraction field. The bias of the exponent is 1023.

```
63     62----------52        51---------------0
S      Exp.+Bias             Fraction
```

Data — 16 Bits

```
1111110000000000
5432109876543210
xxxxxxxxxxxxxxxx

      WORD
15                0
xxxxxxxxxxxxxxx>
      shift right 1 bit
```

*Note: x = data bits.

- **TIMebase:SAMPle REALtime / 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:

Length = 500 points

INPUT1(CHANnel1) = MEM:VME:ADDR + #H016C (364 dec)
INPUT2(CHANnel2) = MEM:VME:ADDR + #H0940 (2368 dec)

Length = 8000 points

INPUT1(CHANnel1) = MEM:VME:ADDR + Dynamic
INPUT2(CHANnel2) = MEM:VME:ADDR + Dynamic

For realtime sampling, 8000 points, it is necessary to query with MEMory:VME:MAP? every time the time/div setting is changed.

Through a query, the starting address of the memory space can be determined for real time, 8000 points at a particular time/division (use BUF8000_1 for CHANnel1, BUF8000_2 for CHANnel2.)

Example in HP BASIC to find memory location for CHANnel1:

```
OUTPUT @Scope; "MEMory:VME:MAP? BUF8000_1"
ENTER @Scope; Memory_loc
PRINT "Memory location is "; Memory_loc
```

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 = data bits. 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:MAP?

MEMory:VME:MAP?[<*source*>] returns the address assignment and size for measurement results and internal buffers, of the source specified, to the output buffer.

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	BUF500_*n* (n=1 to 2)| BUF8000_*n* (n=1 to 2)| SEQUential*n* (n=1 to 2)| MRESults	none

Example **Querying the VME memory map**

`dimension statement`	*Dimension computer string array*
`MEM:VME:MAP?`	*Query instrument to return size (see comments below)*
`enter statement`	*Enter string into computer*

Comments

- **Returned Formats:** The following responses are based on VME address set to #H00200000 and the size set to #H06AD14 (default):

MEMory:VME:MAP? **MRES**

after :MEAS:SCRATCH	*"MRES no measurements"*
after :MEAS:FREQ?	*"MRES frequency at #H0020000C"*
after: :MEAS:FREQ?;PER?	*"MRES frequency at #H0020000C;PER at #H00200030"*

MEMory:VME:MAP? **BUFF500_1**

"BUFF500_,#H0020016C,#000007D4"

MEMory:VME:MAP? **SEQUential1**

"SEQU1,#H00209294,#00030D40"

MEMory:VME:MAP?

after: :MEAS:SCRATCH

```
"MRES no measurements;
                         BUFF500_1,#H0020016C,#000007D4;
                         BUFF500_2,#H00200940,#000007D4;
                         BUFF8000_1,#H00201114,#000040C0;
                         BUFF8000_2,#H0020514D,#000040C0;
                         SEQU1,#H00209294,#00030D40;
                         SEQU2,#H00239FD4,#00030D40;"
```

after: :MEAS:FREQ?;PER?

```
"MRES frequency at #H0020000C;PER at #H00200030;
                         BUFF500_1,#H0020016C,#000007D4;
                         BUFF500_2,#H00200940,#000007D4;
                         BUFF8000_1,#H00201114,#000040C0;
                         BUFF8000_2,#H0020514D,#000040C0;
                         SEQU1,#H00209294,#00030D40;
                         SEQU2,#H00239FD4,#00030D40;"
```

after: :MEAS:FREQ?;PER? and MEAS:STAT ON

```
"MRES frequency at #H0020000C, MIN at #H00200010,
MAX at #H00200014; PER at #H00200030 MIN at
#H00200034, MAX at #H00200038;
                         BUFF500_1,#H0020016C,#000007D4;
                         BUFF500_2,#H00200940,#000007D4;
                         BUFF8000_1,#H00201114,#000040C0;
                         BUFF8000_2,#H0020514D,#000040C0;
                         SEQU1,#H00209294,#00030D40;
                         SEQU2,#H00239FD4,#00030D40;"
```

If any values either from the measurements or the buffers are read directly thru the VXI backplane, they have to be accessed at these addresses.

:VME:SIZE

MEMory:VME:SIZE ***<bytes>*** sets the size, in bytes, of the available VME address space. The oscilloscope has 1 Mbyte of internal shared memory.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
bytes	numeric	#H92B4 through #HC00000|MIN	bytes

Example **Setting the VME memory size**

```
MEM:VME:SIZE  64000
```
Set memory size to 64000

Comments

- **Entering Size:** Memory size can be specified as a value in decimal or hexadecimal (#H....), or as MINimum (#H3D090 or 250,000 dec).

  The absolute minimum size of VME memory required is based on the minimum data size in the in the RAW Data mode (4 points and 1 acquisition) as follows:

meas_queue	= #H016C (364 dec)
buff_500_1	= #H07D4 (2004 dec)
buff_500_2 =	#H07D4 (2004 dec)
buff_8000_1 =	#H40C0 (16576 dec)
buff_8000_2 =	#H40C0 (16576 dec)
sequ1 =	#H0010 (16 dec)
sequ2 =	#H0010 (16 dec)
Total =	#H92B4 (37556 dec)

- **Internal Buffer Size:** The internal buffer size of the oscilloscope is #H6AD14 (default value).
- **Maximum Memory Size:** The maximum memory size available in VME address space is DF7FFE (14647294 dec).
- ***RST Condition:** MEM:VME:SIZE defaults to #H6AD14.

: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 64,000 Bytes*
`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 64,000 Bytes*
`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.
- ***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 or 1), or the "Probe Comp/Cal/Trig Output" BNC port on the Oscilloscope front panel.

OUTPut[:STATe] acts like the master switch for the OUTPut subsystem. If the ECLTrg 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]?
```

: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:** Both trigger outputs, ECLTrg (0 or 1) 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 EXTernal and ECLTrg2, send the following commands:

```
OUTP:EXT 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 "Probe Comp/Cal/Trig 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 trigger output BNC port

```
OUTP:EXT  1             Enable "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 "Probe Comp/Cal/Trig Output" BNC port on the Oscilloscope Module. The output is a negative going pulse.
- **Numerous outputs selected at a time:** Both trigger outputs, ECLTrg (0 or 1) 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 EXTernal and ECLTrg2, send the following commands:

  ```
  OUTP:EXT 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 "Probe Comp/Cal/Trig Output" BNC port for trigger output. 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 Trigger Output BNC Port Enable State**

```
dimension  statement    String for data
OUTP:EXT  ON            Enable "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 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 "Trigger Output" BNC port to
                        output pulse
OUTP  1                 Enable output subsystem
```

Comments

- **Selecting Outputs:** Use the 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
```

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>]]]
LTER?
POWerup <state>
POWerup?
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
  FUNCtion to OFF
  MEASure to OFF
  TIMebase:RANGe as required
  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.

- **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 the specified CHANnel<*n*>, FUNCtion<*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 2) FUNCtion*n* (*n*=1 to 2) WMEMory*n* (*n*=1 to 4)	None

Example

Blank Channel 2

```
BLAN  CHAN2
```

Turn off channel 2

Comments

- **Turn On Waveform:** Use the VIEW command to turn on the specified channel, function, or waveform memory.
- **Related Commands:** VIEW.

BNC

BNC **<*output*>** selects either PROBe, CAL, TRIGger, 0 volts, or 5 volts as the output mode of the Probe Comp/Cal/Trig Output BNC connector. PROBe outputs a square wave signal at ≈500 Hz, and TRIGger outputs a rising edge when an internal trigger occurs. DC CAL is used during calibration. ZVOLt sets the output to 0 volts, and FVOLt sets the output to 5 volts.

Subsystem Syntax BNC <*output*>

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
output	discrete	PROBe|TRIGger|DCCAL| ZVOLt|FVOLt	none

Example **Set BNC connector to trigger**

```
BNC  TRIG
```

Probe Comp / Cal / Trig output to trigger

Comments

- ***RST Condition:** Defaults to DCCAL.

BNC?

BNC? returns the currently selected Probe Comp/Cal/Trig Output BNC connector output signal. Returns PROBe, TRIGger, DCCAL, ZVOLt, or FVOLt depending on current selection. 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`	*Probe Comp / Cal / Trig 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 and/or 2 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 2)|CONTinue	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 two channels can be digitized using a single command.
- **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.
- **Using CONTinue:** If power is interrupted during a digitize operation and POWerup STOPped state is selected, then prior to extracting previously digitized data, a DIGitize:CONTinue must be executed to finish the digitizing process (after the oscilloscope powers up). Attempting to extract data (using the WAVeform PREamble? or :DATA? queries) without sending the DIGitize:CONTinue command will cause a settings conflict error.

- **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:, TIMEbase:, WAVeform:, POWerup, RUN.

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?.

POWerup

POWerup <*state*> is used to enable the oscilloscope to power up in either the stopped state or running state.

Subsystem Syntax

POWerup <*state*>

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
state	discrete	STOPped|RUNning	none

Example

Oscilloscope powers up in the stopped state

```
POW  STOP
```

Set to power up in the stopped state

Comments

- **POWerup versus DIGitize:** If power is interrupted during a digitize operation and the POWerup STOPped state is enabled, then the DIGitize:CONTinue must be executed to finish the digitizing process (after the oscilloscope powers up) prior to extracting the previously digitized data using the WAVeform PREamble? or :DATA? query. Attempting to extract data without sending the DIGitize:CONTinue command will cause a settings conflict error.
- **POWerup STOPped:** When selected, forces real-time raw data mode, with 1 acquisition.
- **Related Commands:** STOP, RUN, DIGitize:CONTinue.

POWerup?

POWerup? returns the power up state currently enabled. The data is sent to the output buffer. See POWerup command for more information on available types.

Subsystem Syntax

POWerup?

Example

Querying power up state

```
Dimension  statement
POW  STOP
POW?
enter  statement
```

String for data

Set to power up in the stopped state

Query instrument to return power up state

Enter results into computer

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.
- **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 (").
- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to writing a new serial number to the protected non-volatile ram within the instrument. (see SYSTem:NVPRotect)
- **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*>, 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 2)| FUNCtion*n* (*n*=1 to 2)| WMEMory*n* (*n*=1 to 4)|	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, or waveform 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 2)| 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.

  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*>, 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 2)| FUNCtion*n* (*n*=1 to 2)| WMEMory*n* (*n*=1 to 4)|	None

Example

View Channel 2

```
VIEW  CHAN2                Turn on channel 2
```

Comments

- **Turn off Waveform:** Use the BLANk command to turn off the specified channel, function, or waveform memory.
- **Unused Channels:** BLANk all channels not in use. See Appendix C, Optimizing Measurements, for additional information.
- **Related Commands:** BLANk.

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.

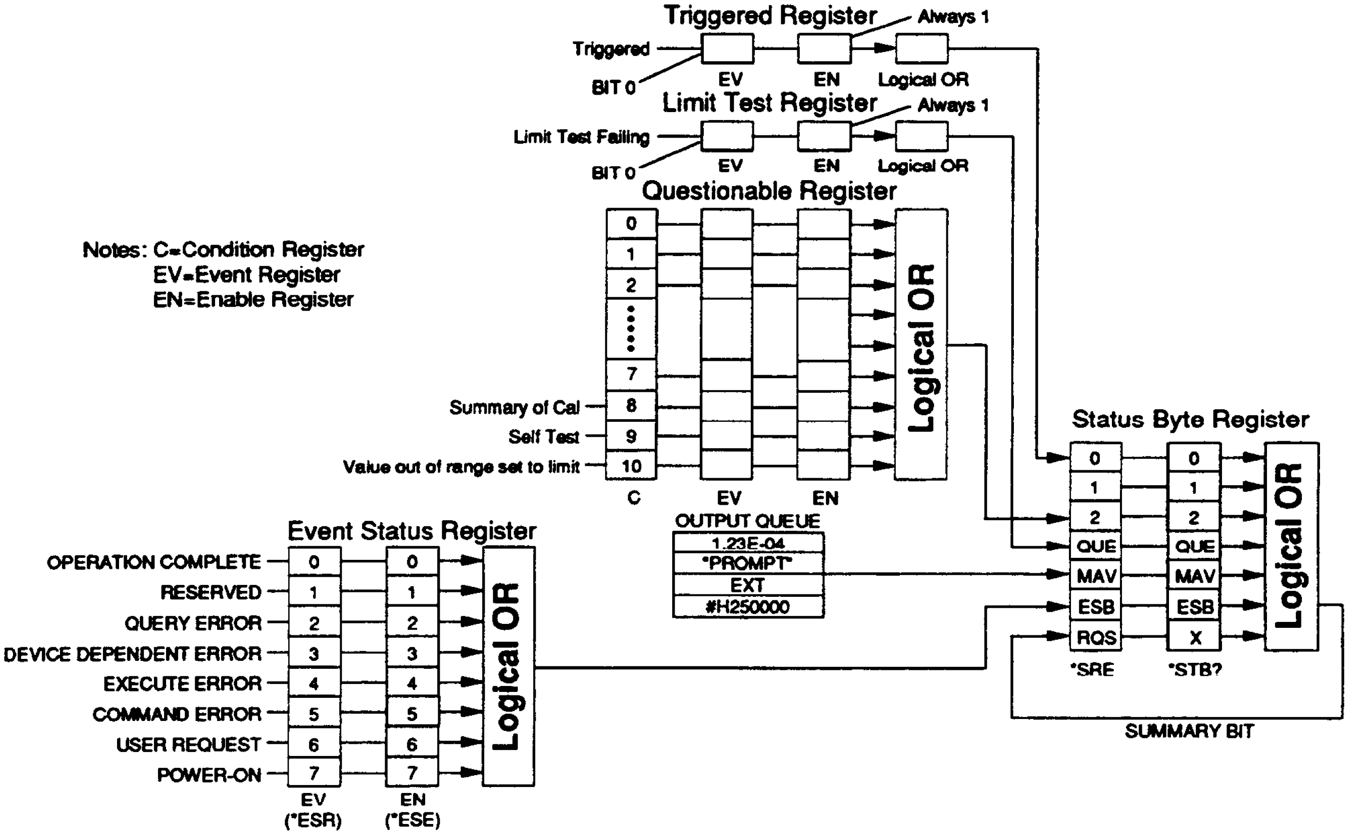


Figure 4-1. Oscilloscope Summary Registers

Subsystem Syntax

```
SUMMary
    :PRESet
    :QUEStionable
        :CONDition?
        :ENABle
        [:EVENt]?
        :CALibration
            :CONDition?
            :ENABle
            [:EVENt]?
            :CHANnel<number> | EXTernal
                :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

```
SUMMary — Continued
    :QUEStionable — Continued
        :CALibration — 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]?
                :NVOLatile
                    :CONDition?
                    :ENABle
                    [:EVENt]?
```

Subsystem Syntax

```
SUMMary — Continued
    :QUEStionable — Continued
        :TEST — Continued
            :RAM — Continued
                :SYSTem
                    :CONDition?
                    :ENABle
                    [:EVENt]?
            :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.

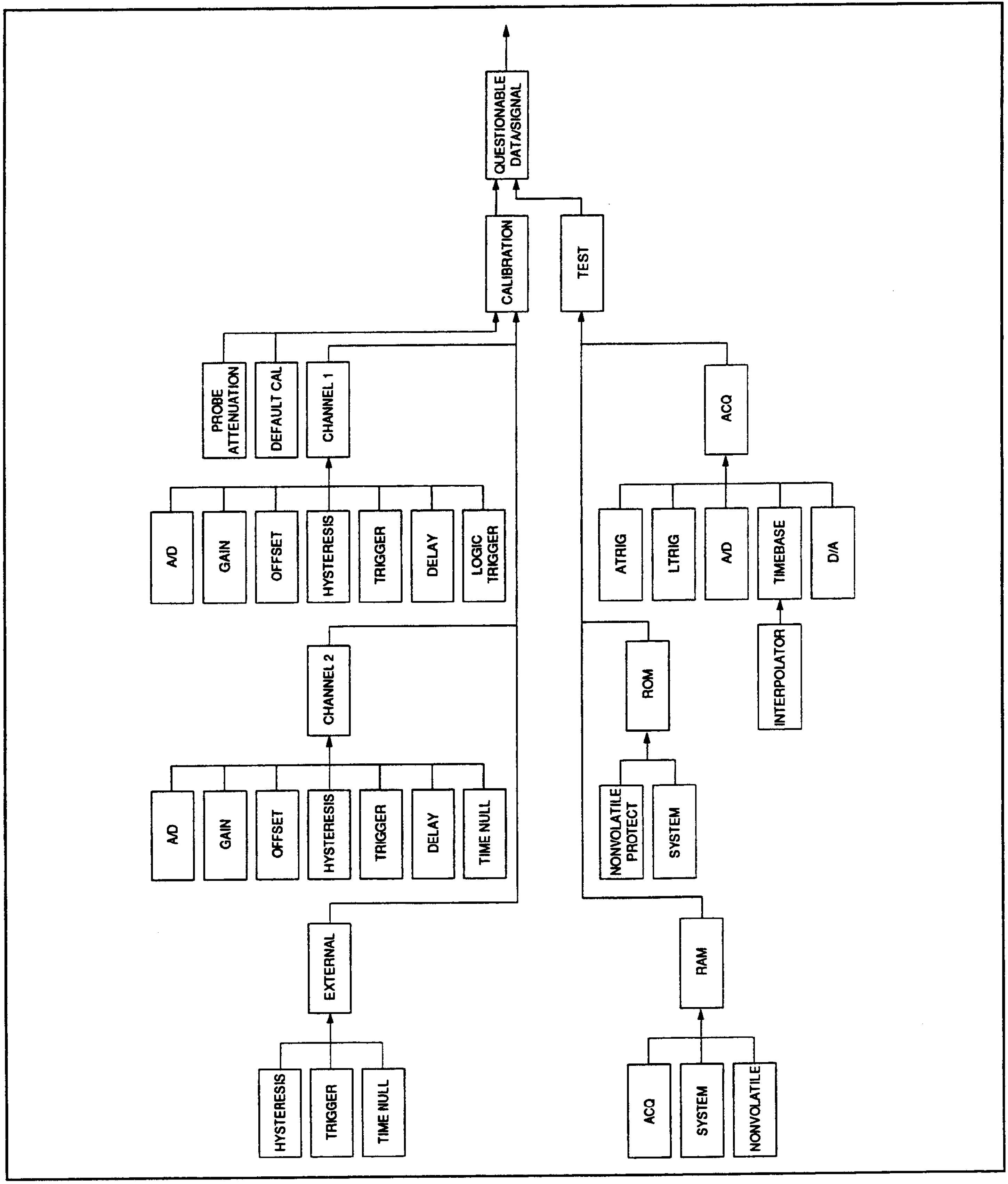
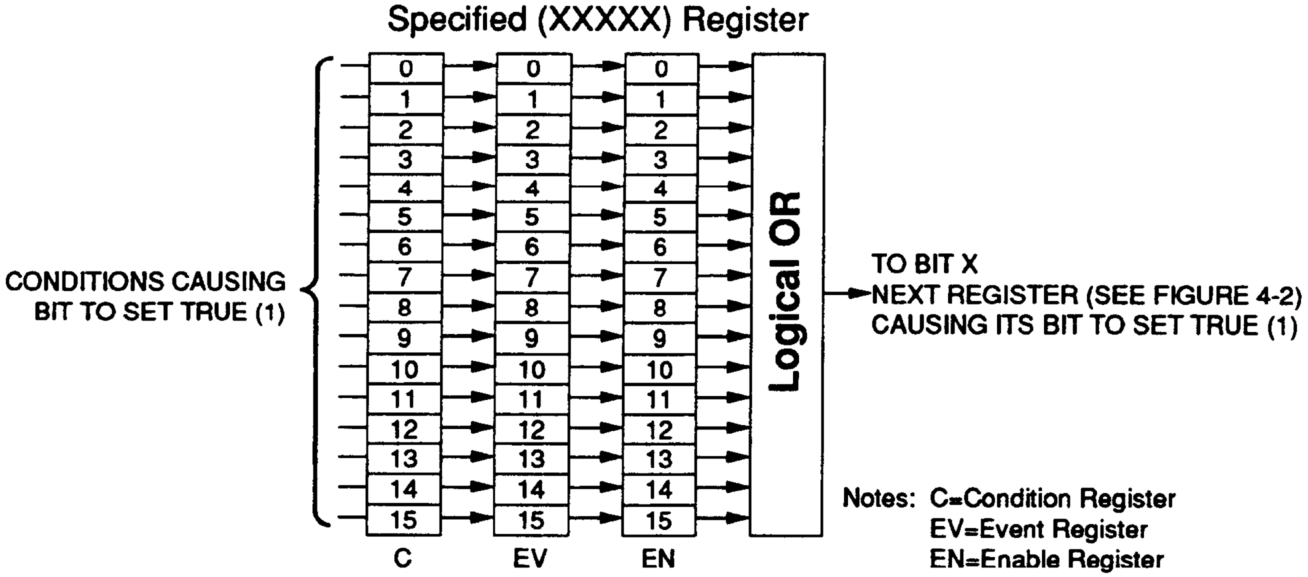


Figure 4-2. Oscilloscope Summary Questionable Data/Signal Register Subsystem

:QUEStionable

SUMMary:QUEStionable 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.

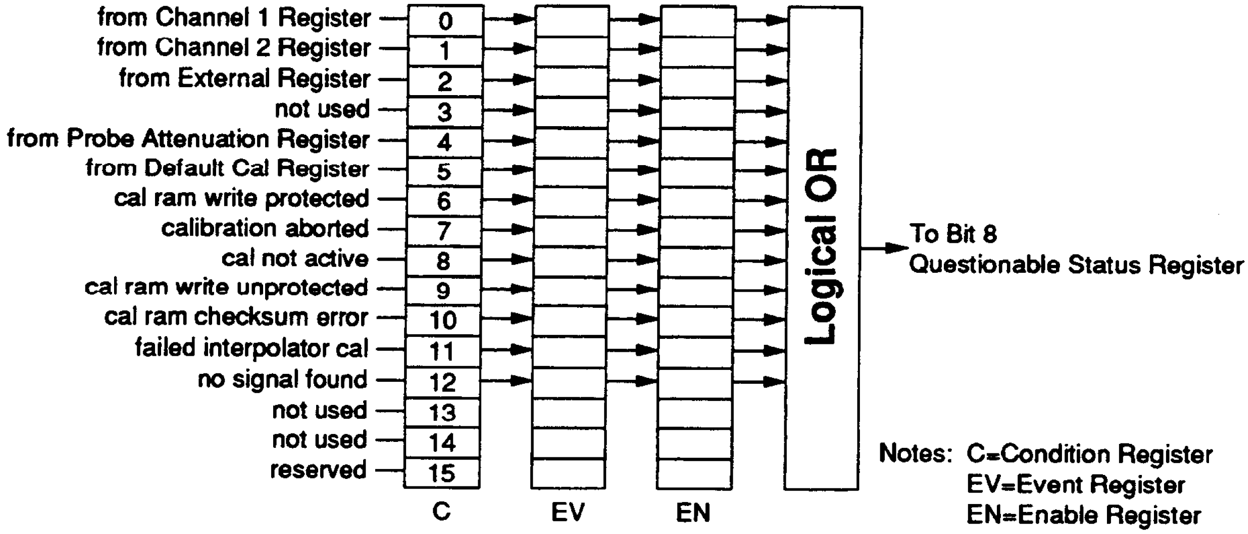
Example **Read the event register**

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?
```

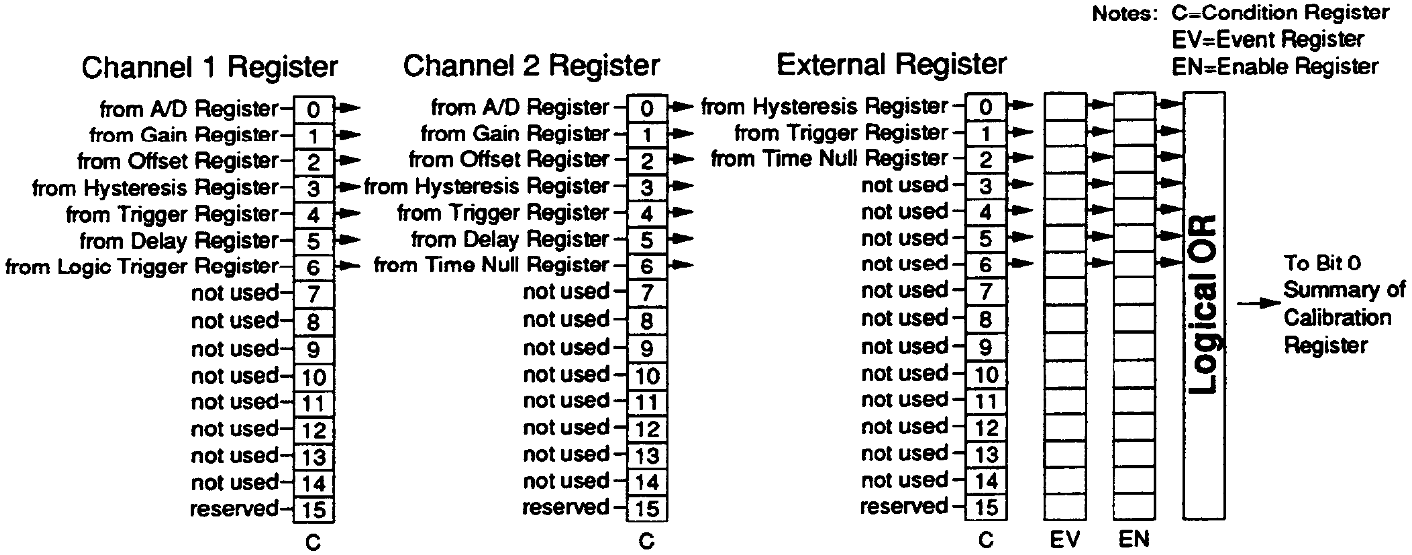
Query instrument to return register contents

:QUEStionable:CALibration:CHANnel|EXTernal

SUMMary:QUEStionable:CALibration:CHANnel*<number>* | EXTernal register reports the status of calibration data for the input specified. Channel *number* (1 to 2) or EXTernal specifies the desired input. 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 2	none



Example

Query channel 1 event register

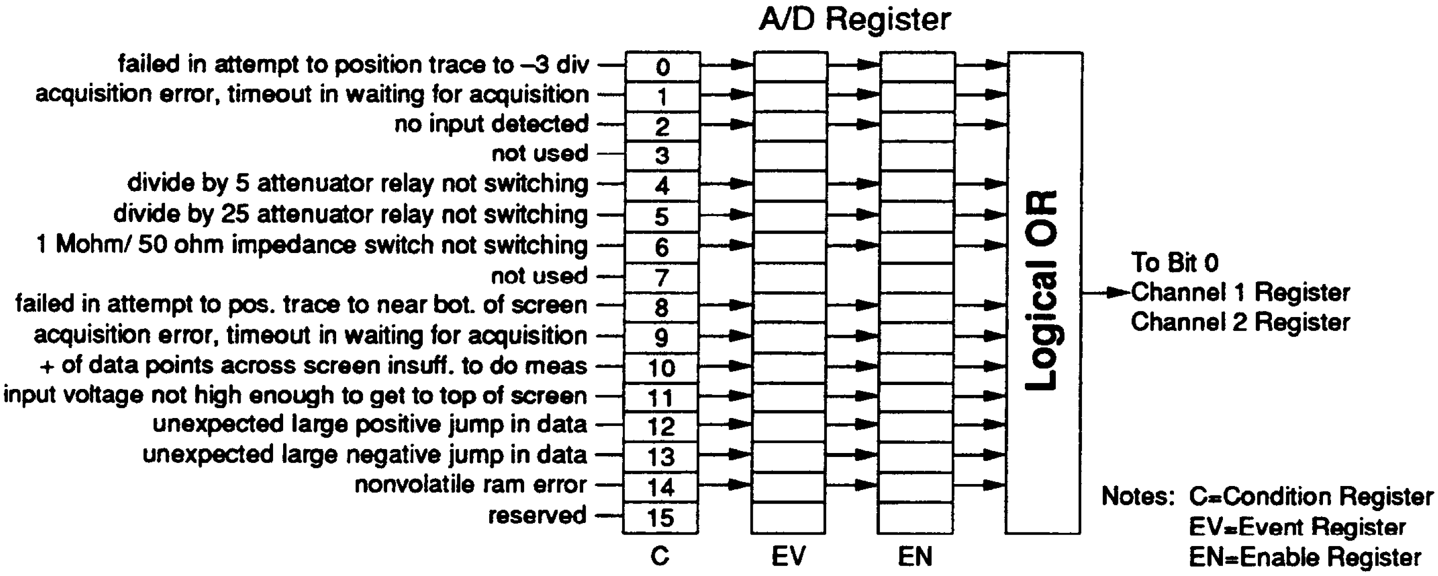
SUMM:QUES:CAL:CHAN1? *Query instrument to return register contents*

: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 2) 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 2	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
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query channel 2 A/D event register**

```
SUMM:QUES:CAL:CHAN2:AD?
```

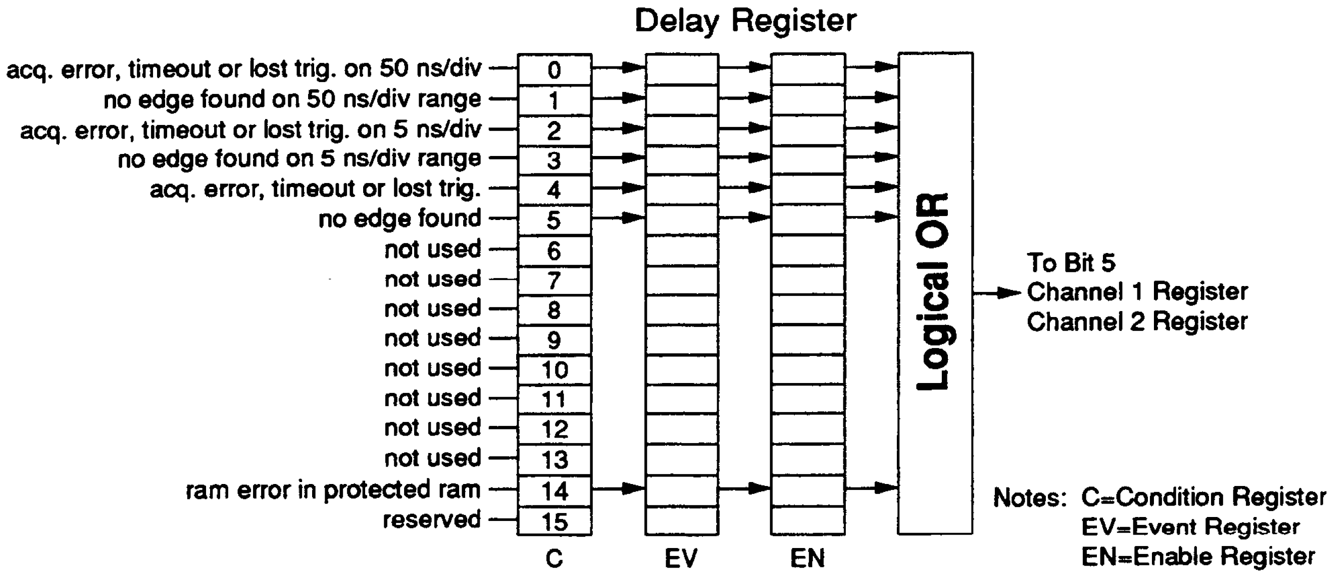
Query instrument to return register contents

: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 2) 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 2	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.
no edge found
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
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example

Query channel 2 delay event register

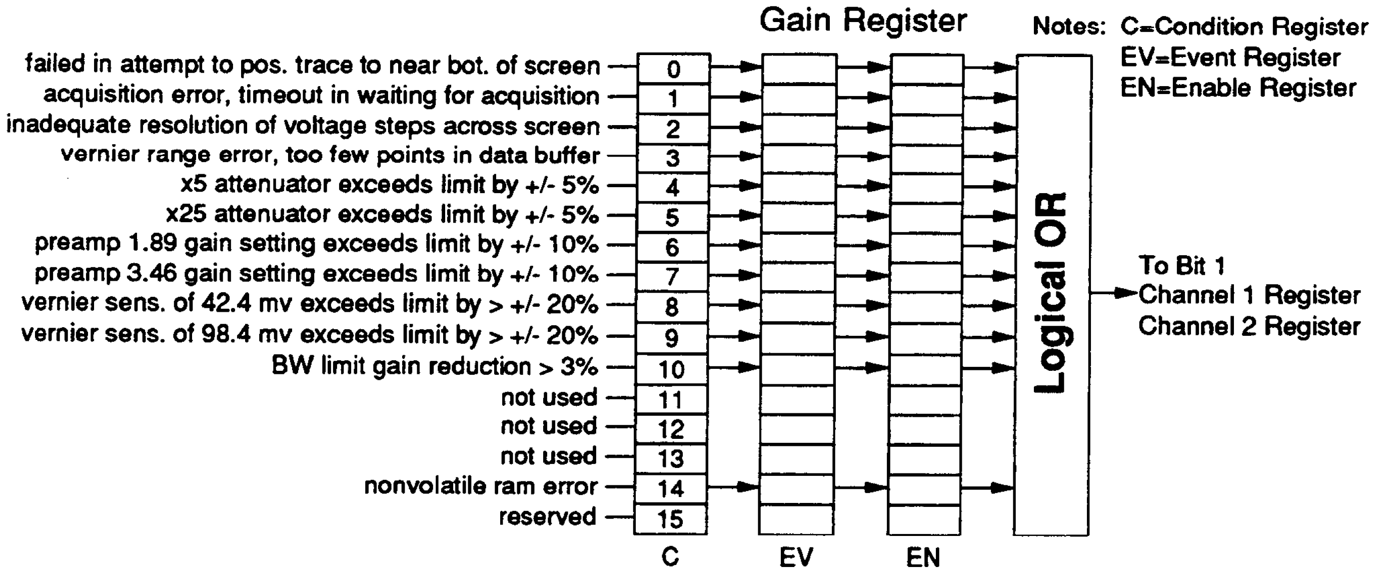
`SUMM:QUES:CAL:CHAN2:DEL?`	*Query instrument to return register contents*

: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 2) 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 2	none


Gain Register
Notes: C=Condition Register
EV=Event Register
EN=Enable 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%
BW 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
To Bit 1
Channel 1 Register
Channel 2 Register

Example

Query channel 2 gain event register

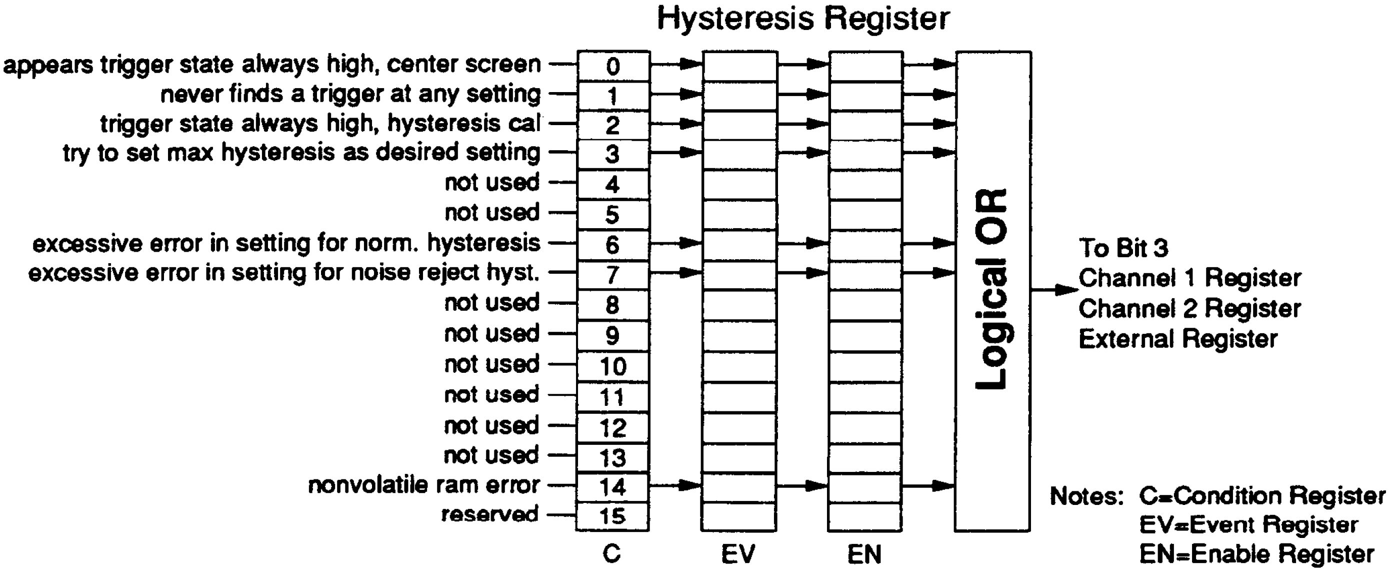
`SUMM:QUES:CAL:CHAN2:GAIN?` *Query instrument to return register contents*

:QUEStionable:CALibration:CHANnel|EXTernal:HYSTeresis

SUMMary:QUEStionable:CALibration:CHANnel<*number*> | EXTernal:HYSTeresis register reports the status of hysteresis calibration data for the input specified. Channel *number* (1 to 2) or EXTernal specifies the desired input. 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 2	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
External Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example

Query external trigger hysteresis event register

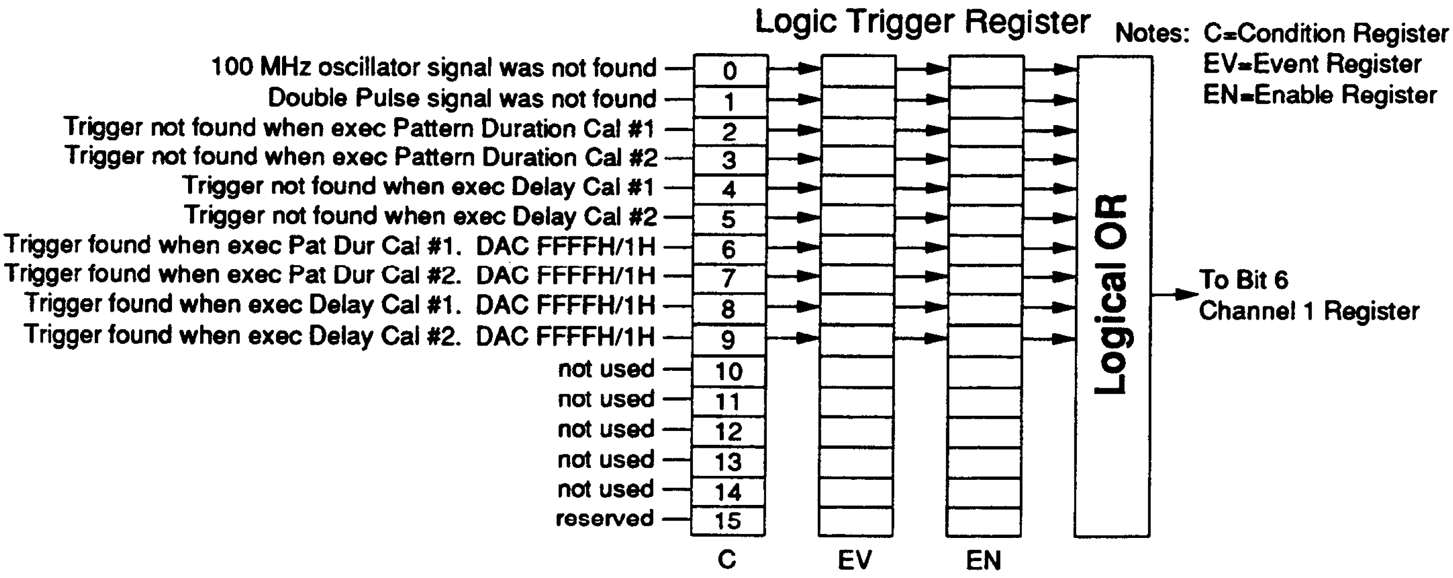
`SUMM:QUES:CAL:EXT:HYST?`	*Query instrument to return register contents*

: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 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
Channel 1 Register

Example **Query channel 1 logic trigger event register**

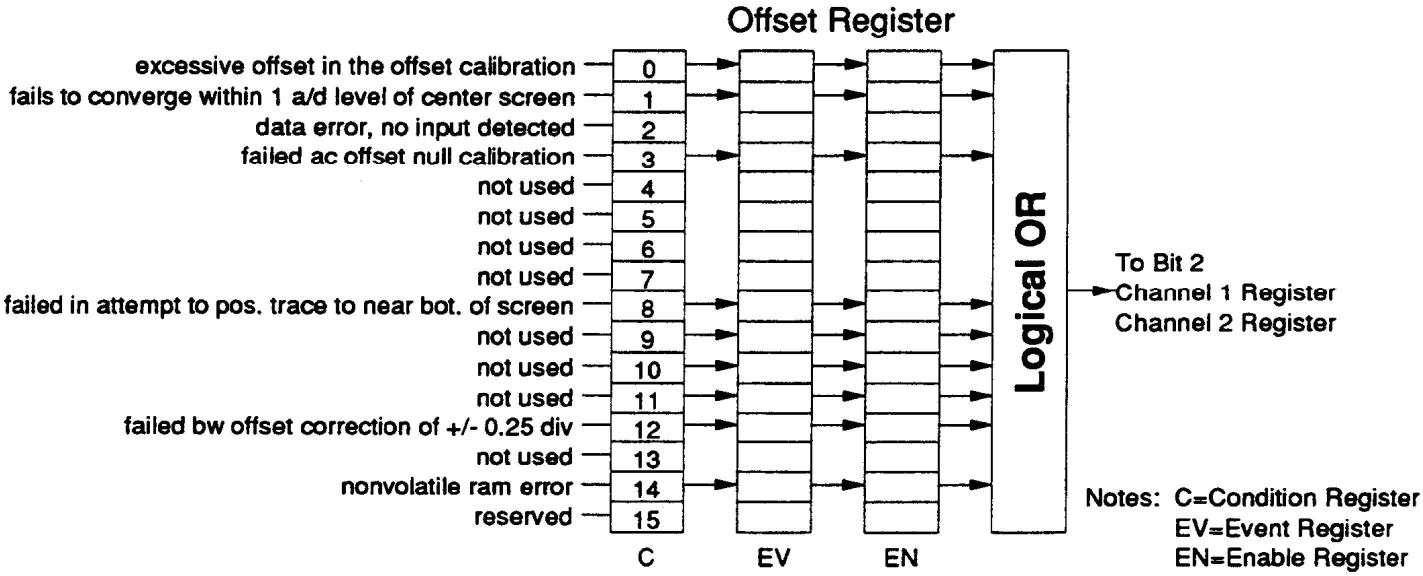
`SUMM:QUES:CAL:CHAN1:LTR?` *Query instrument to return register contents*

: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 2) 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 2	none


Offset Register
excessive offset in the offset calibration
fails to converge within 1 a/d level of center screen
data error, no input detected
failed ac offset null calibration
not used
not used
not used
not used
failed in attempt to pos. trace to near bot. of screen
not used
not used
not used
failed bw offset correction of +/- 0.25 div
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 2
Channel 1 Register
Channel 2 Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example

Query channel 2 offset event register

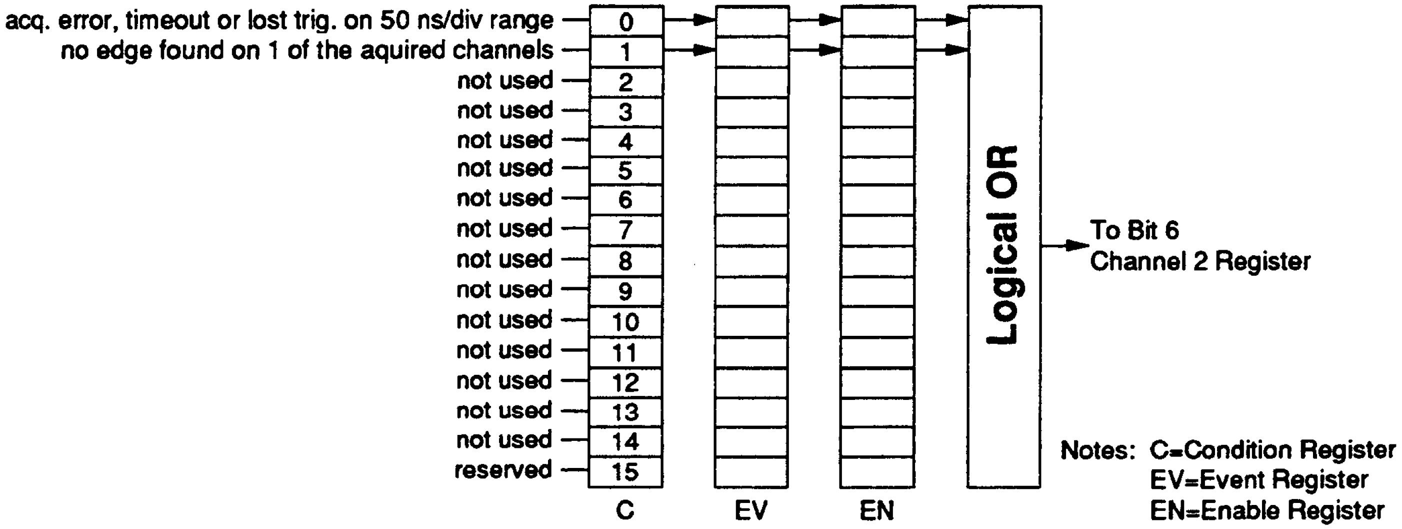
```
SUMM:QUES:CAL:CHAN2:OFFS?
```

Query instrument to return register contents

:QUEStionable:CALibration:CHANnel2|EXTernal:TNULl

SUMMary:QUEStionable:CALibration:CHANnel2 | EXTernal:TNULl register reports the status of time null calibration data for the input specified. Channel 2 or EXTernal specifies the desired input. 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.

Time Null Register



Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query channel 1 time null event register**

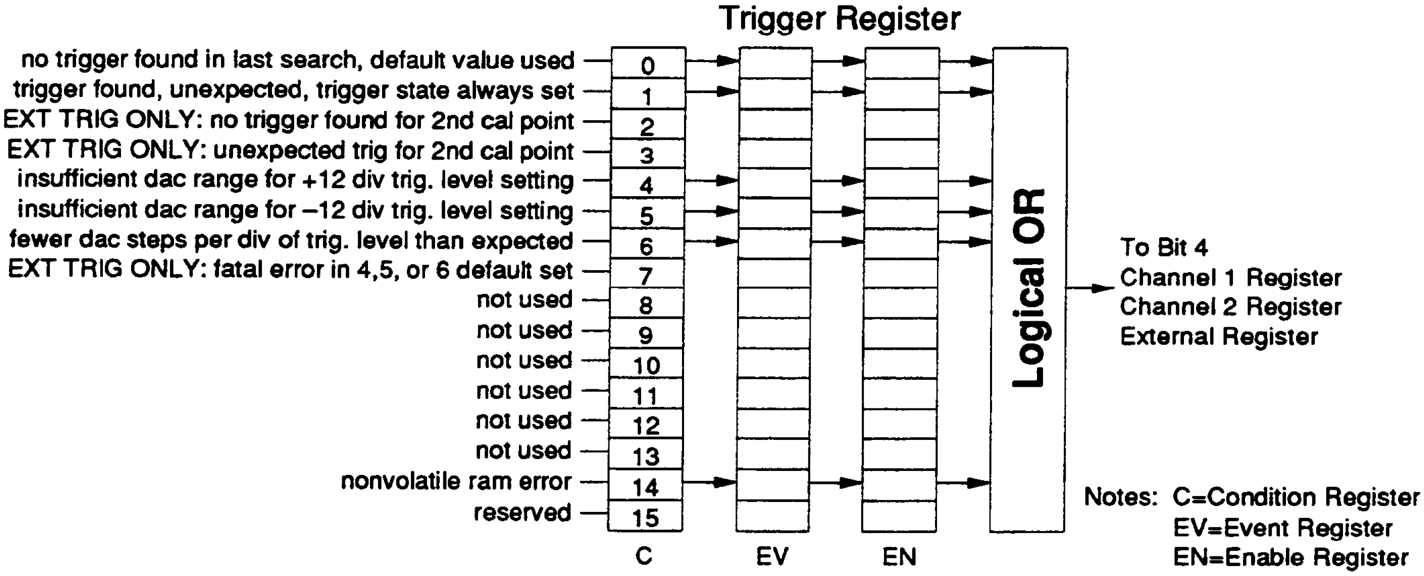
`SUMM:QUES:CAL:CHAN1:TNUL?` *Query instrument to return register contents*

:QUEStionable:CALibration:CHANnel|EXTernal:TRIGger

SUMMary:QUEStionable:CALibration:CHANnel*<number>* | EXTernal:TRIGger register reports the status of trigger calibration data for the input specified. Channel *number* (1 to 2) or EXTernal specifies the desired input. 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 2	none


Trigger Register
no trigger found in last search, default value used
trigger found, unexpected, trigger state always set
EXT TRIG ONLY: no trigger found for 2nd cal point
EXT TRIG ONLY: unexpected trig for 2nd cal point
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
EXT TRIG ONLY: fatal error in 4,5, or 6 default set
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
External Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

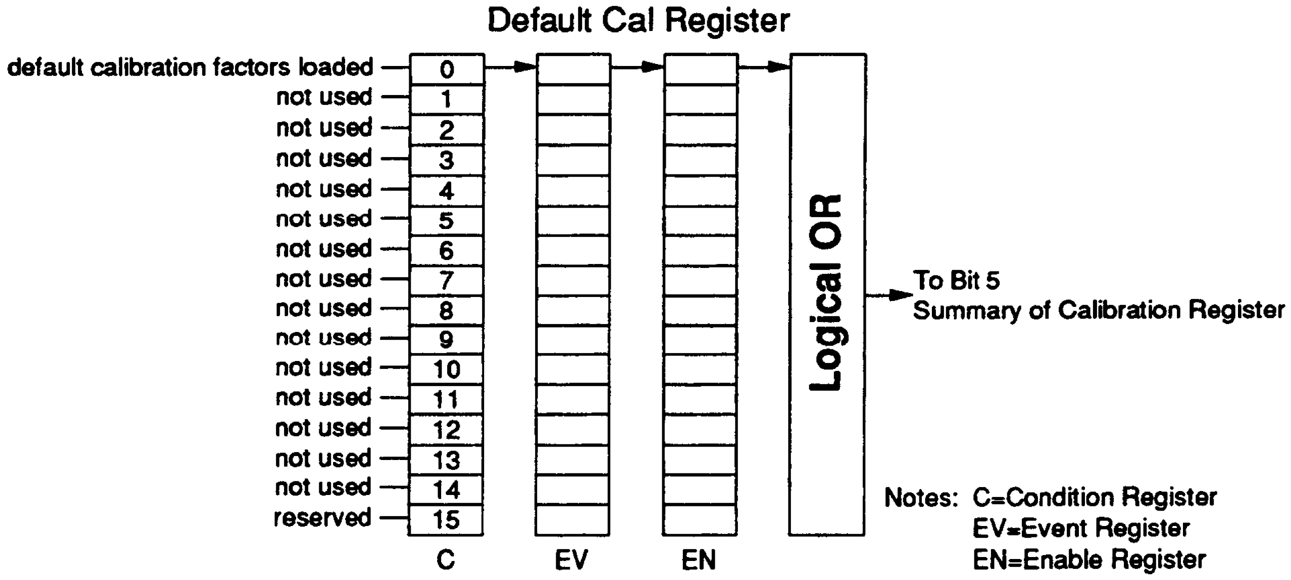
Example

Query channel 2 trigger event register

`SUMM:QUES:CAL:CHAN2: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.

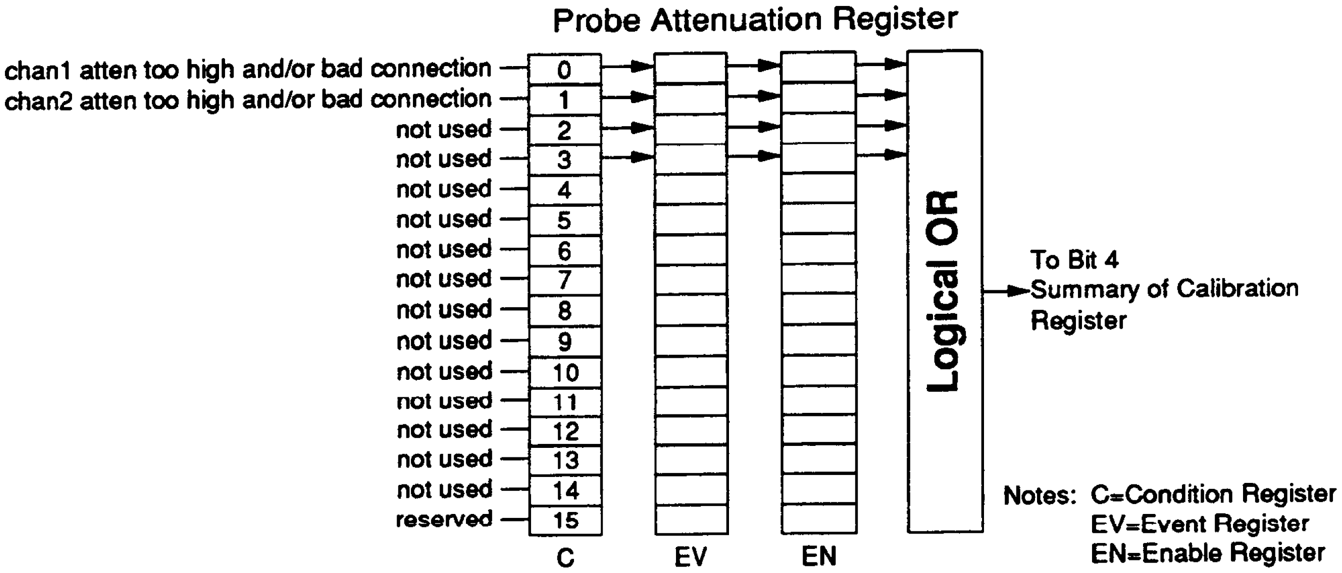


Example **Query default calibration event register**

`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.

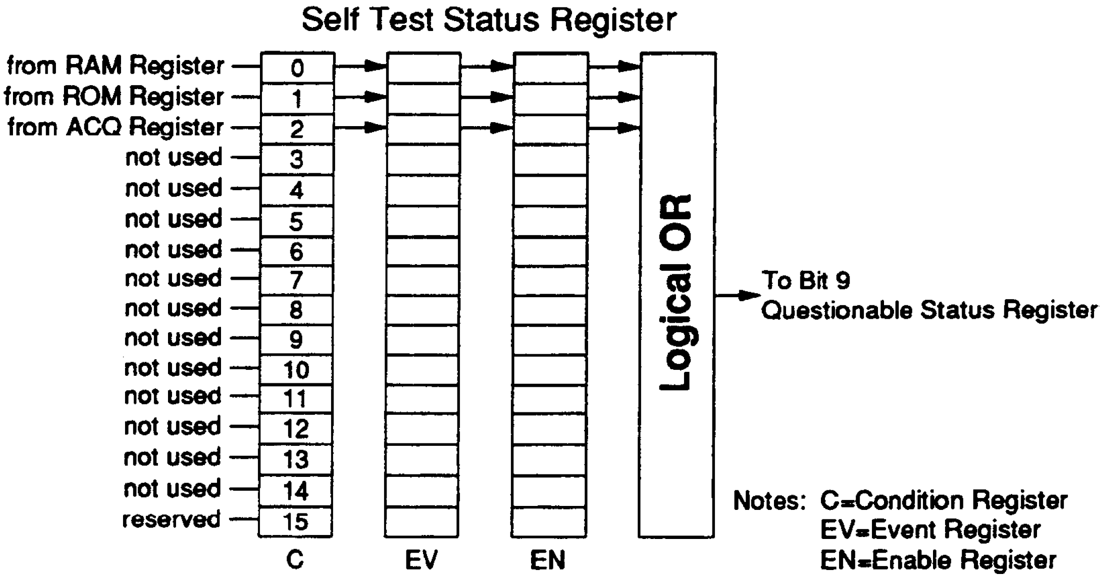


Example **Query probe calibration attenuation event register**

`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.

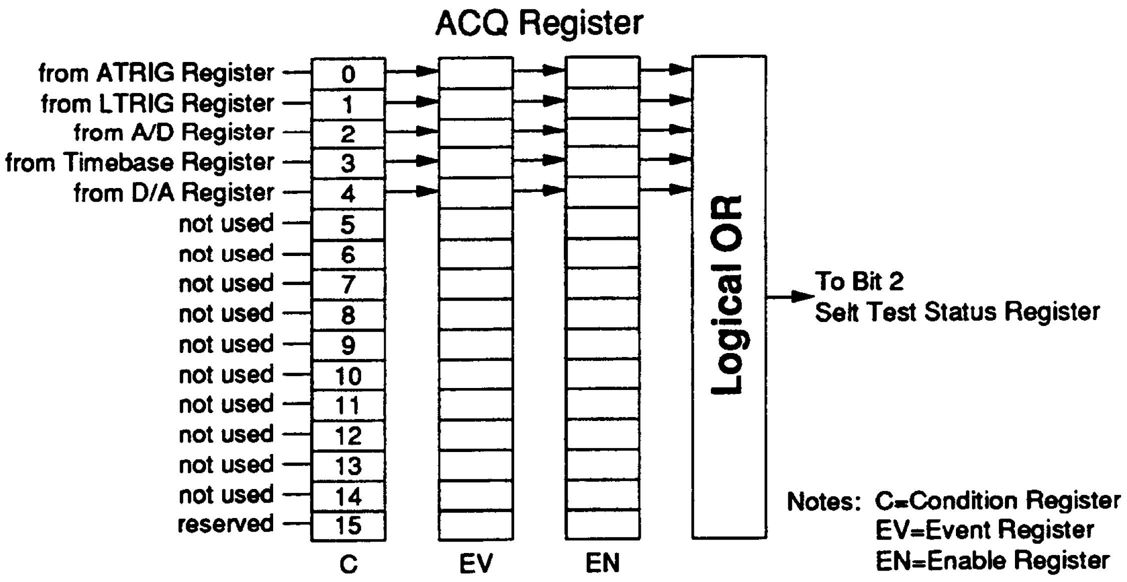


Example **Query test event register**

`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.

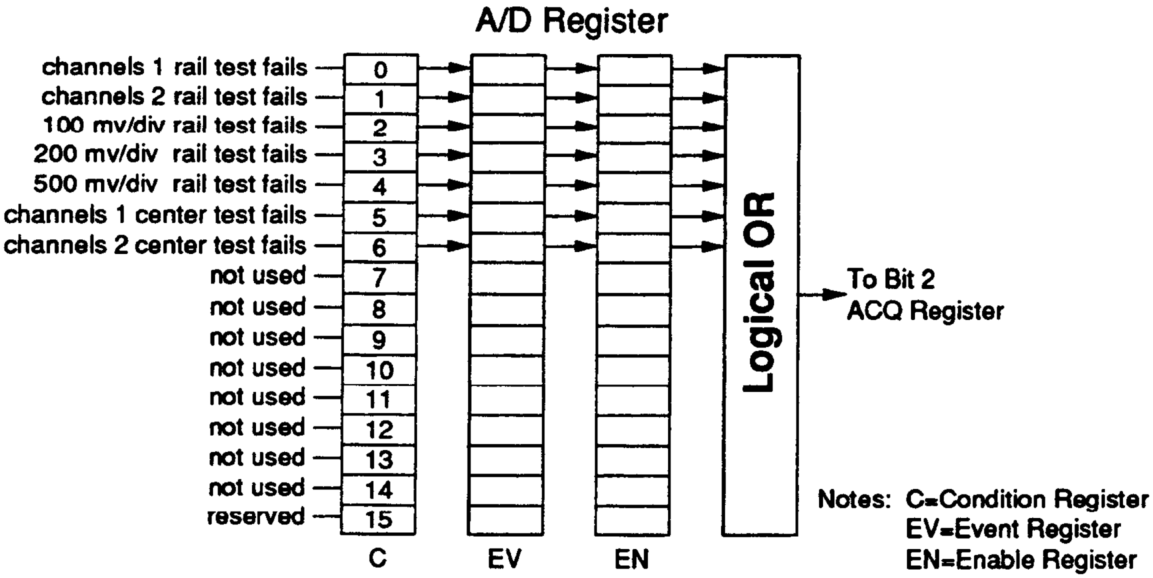


Example **Query acquisition event register**

`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.

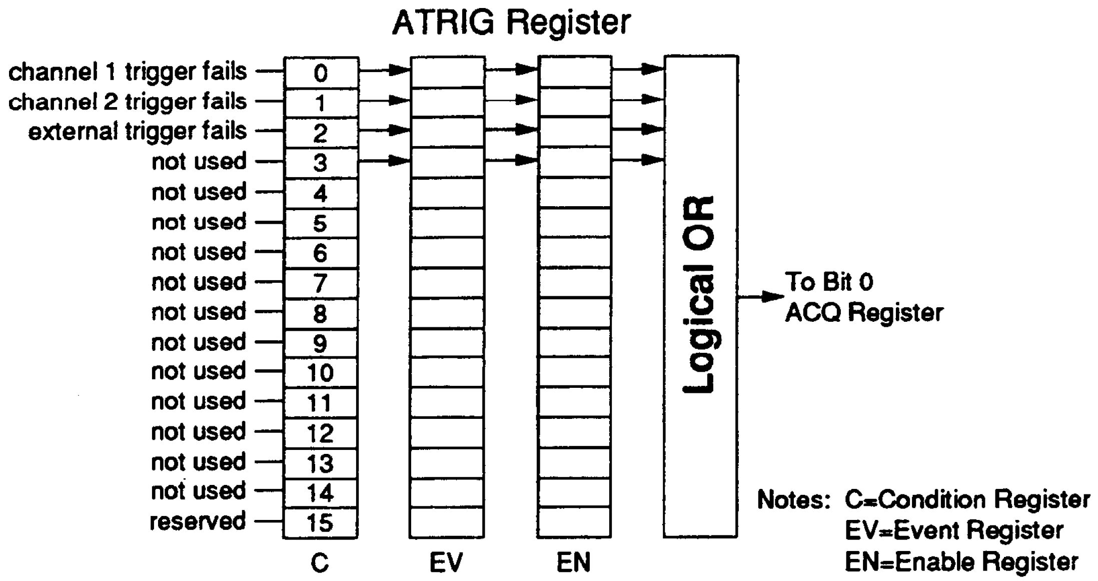

A/D Register
channels 1 rail test fails
channels 2 rail test fails
100 mv/div rail test fails
200 mv/div rail test fails
500 mv/div rail test fails
channels 1 center test fails
channels 2 center test fails
not used
not used
not used
not used
not used
not used
not used
not used
reserved
0
1
2
3
4
5
6
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.



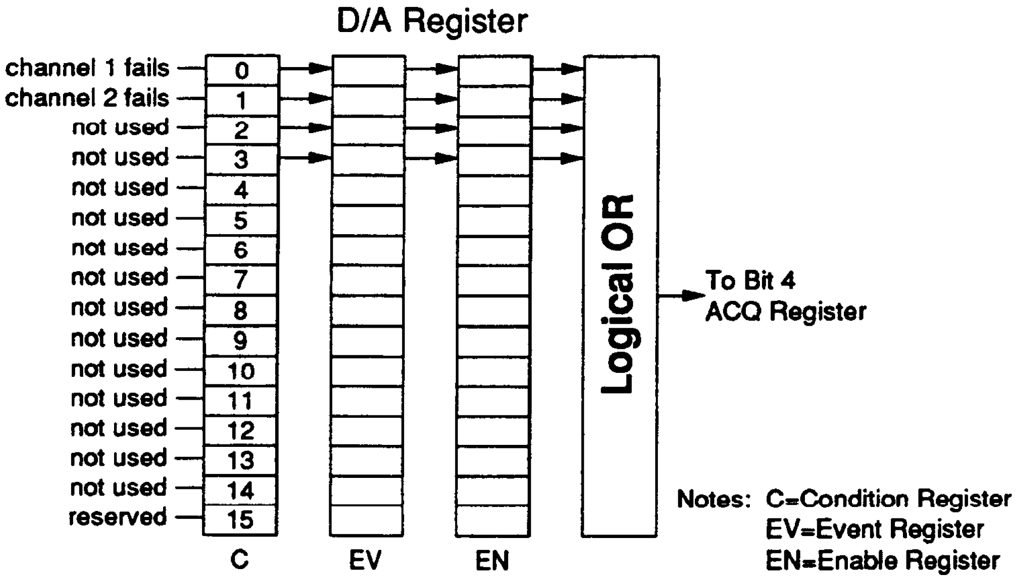
Example **Query acquisition analog trigger event register**

```
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.

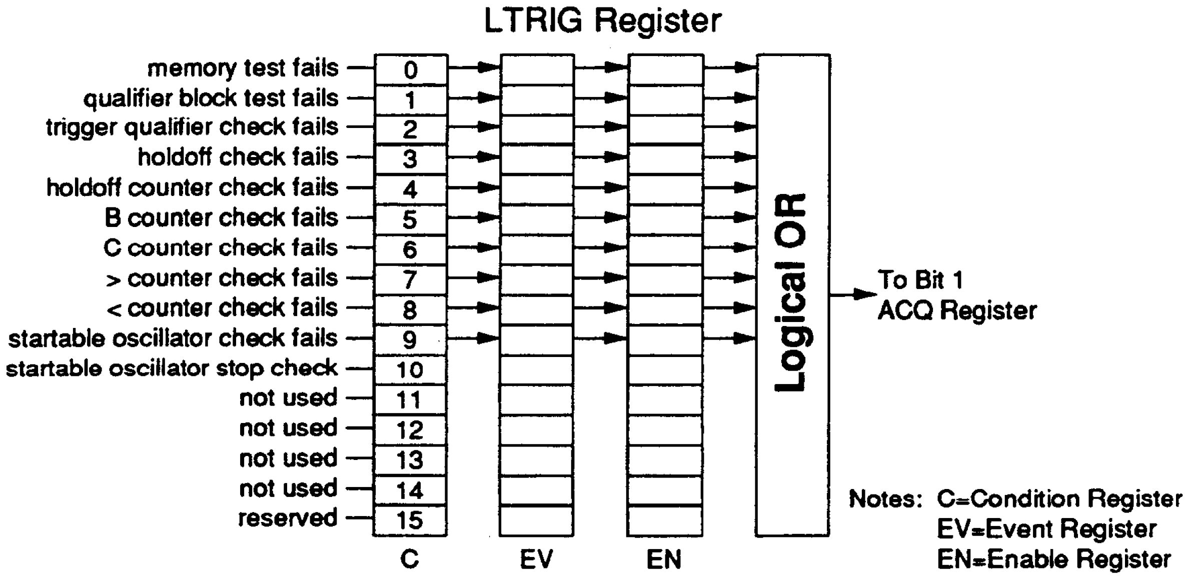


Example **Query acquisition D/A event register**

`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.



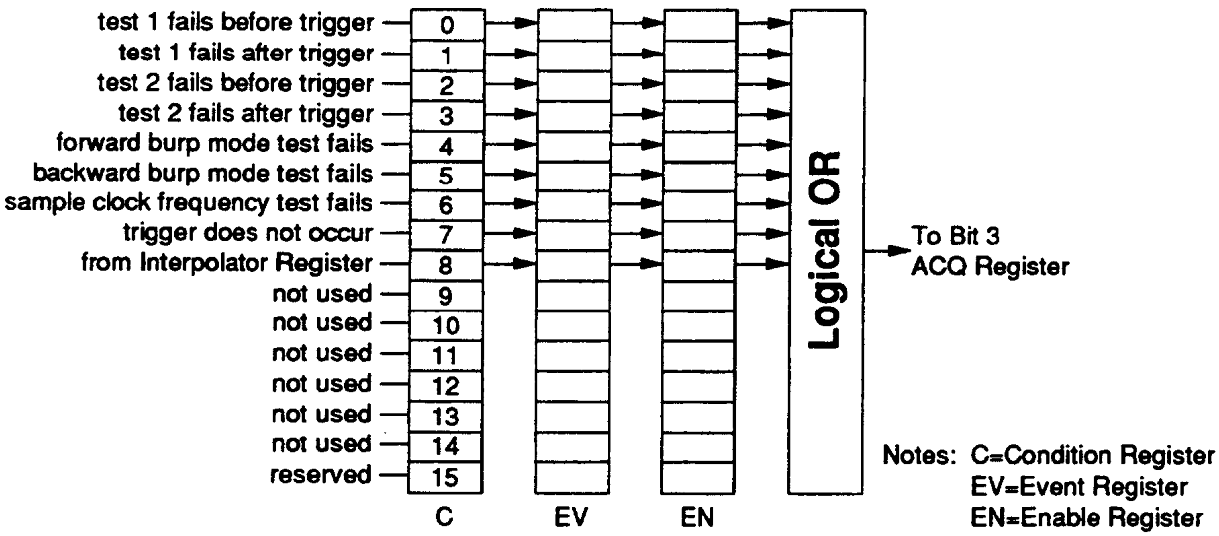
Example **Query acquisition logic trigger event register**

`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

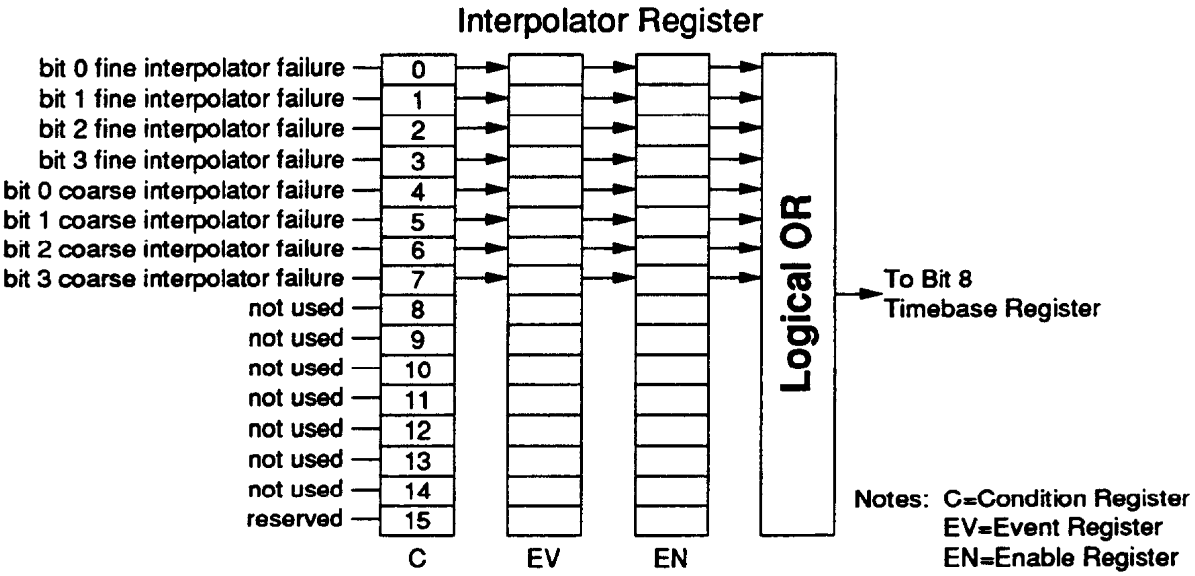
Example **Query acquisition time base event register**

```
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.

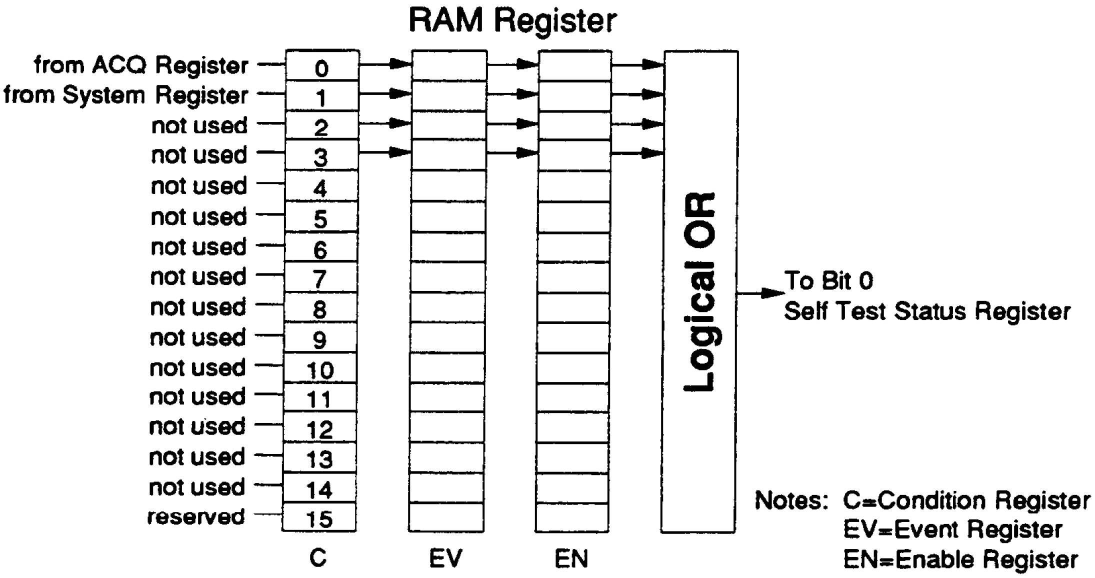

Interpolator Register
bit 0 fine interpolator failure
bit 1 fine interpolator failure
bit 2 fine interpolator failure
bit 3 fine interpolator failure
bit 0 coarse interpolator failure
bit 1 coarse interpolator failure
bit 2 coarse interpolator failure
bit 3 coarse interpolator failure
not used
not used
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.

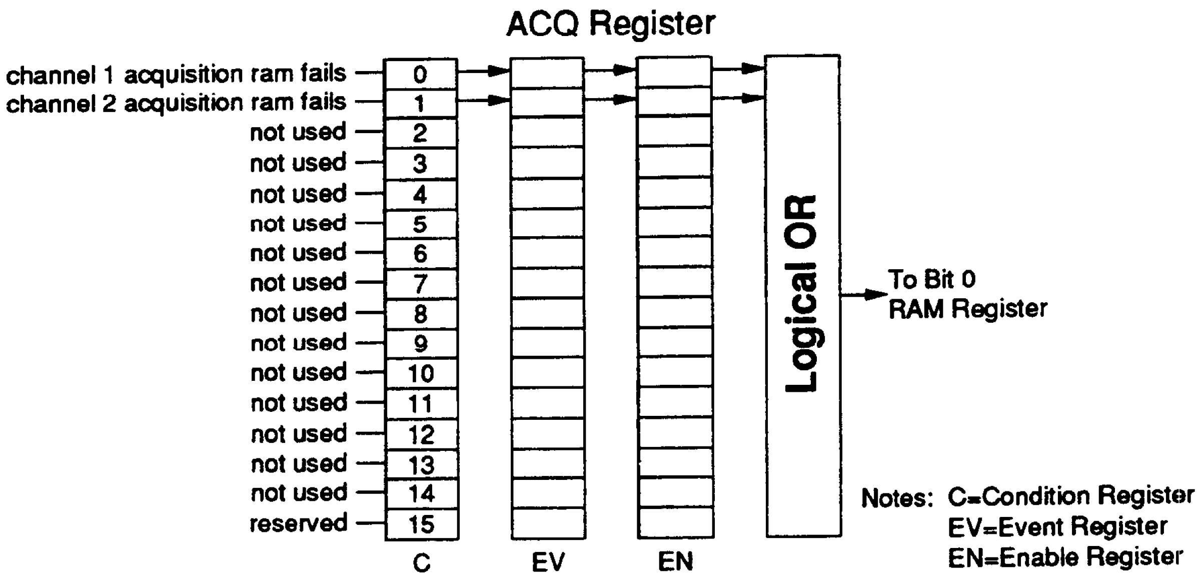


Example **Query RAM event register**

`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.

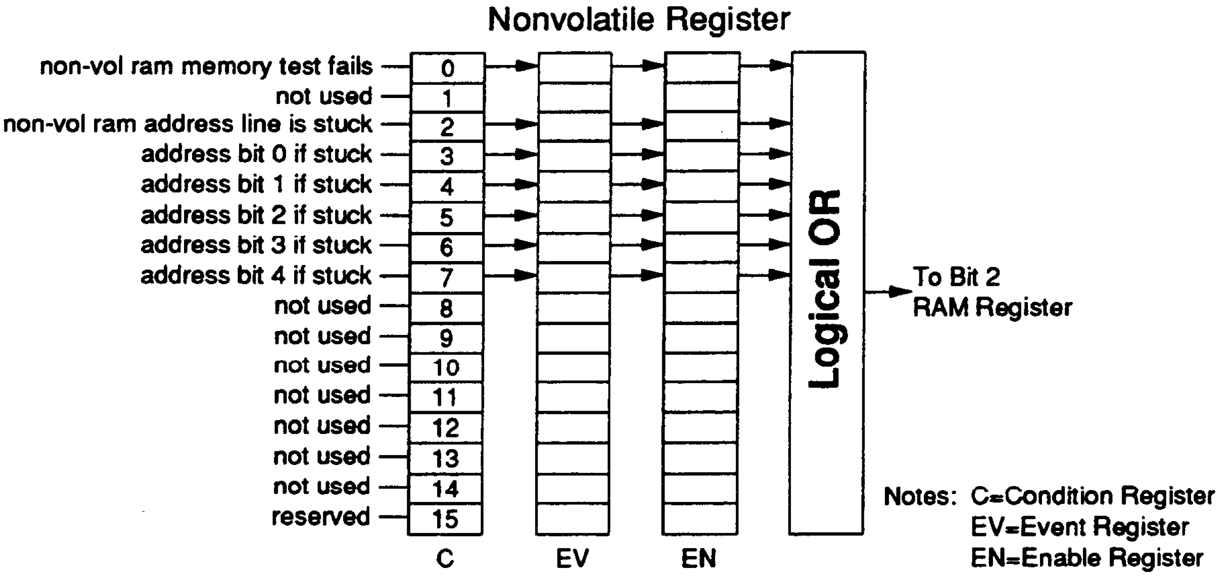


Example **Query acquisition RAM event register**

`SUMM:QUES:TEST:RAM:ACQ?` *Query instrument to return register contents*

:QUEStionable:TEST:RAM: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.



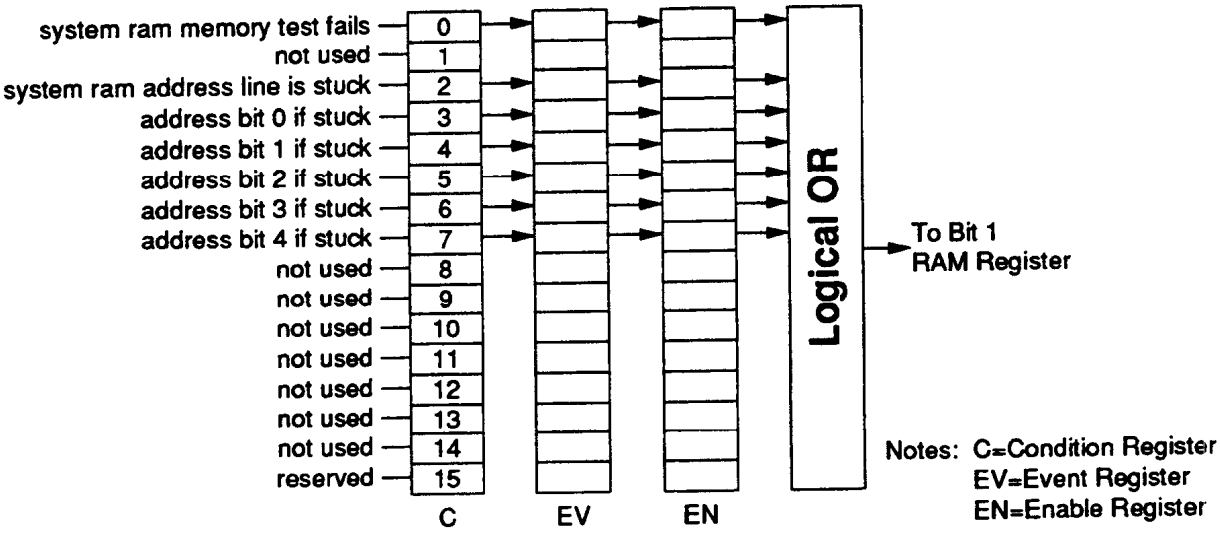
Example **Query nonvolatile RAM event register**

`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.

System Register

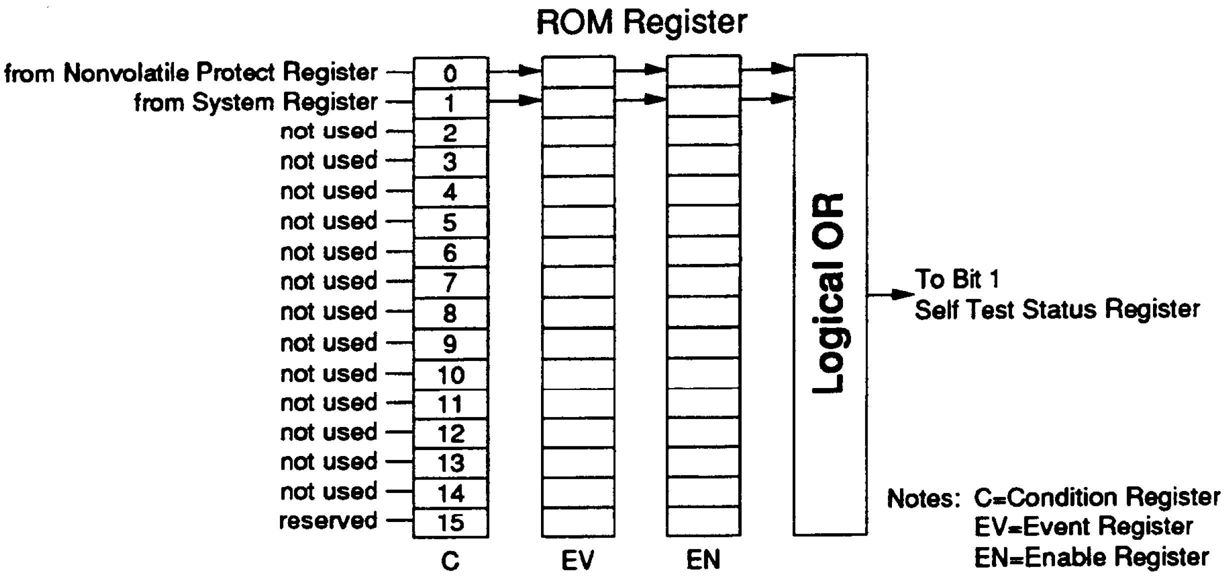


Example **Query system RAM event register**

`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.

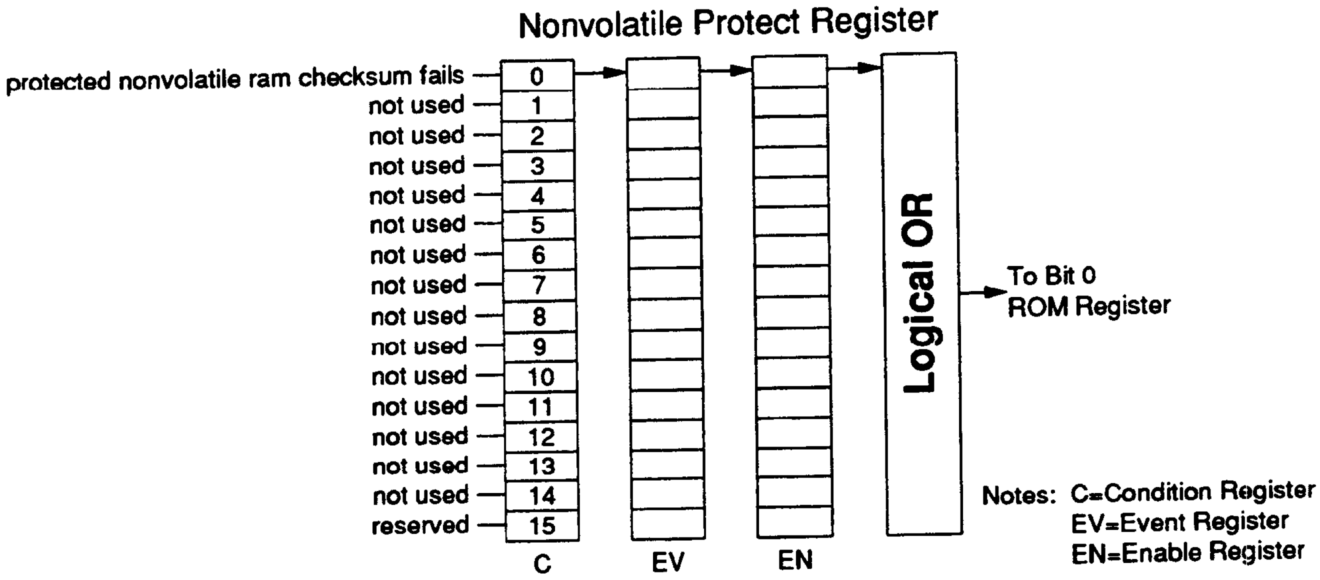


Example **Query ROM event register**

`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.

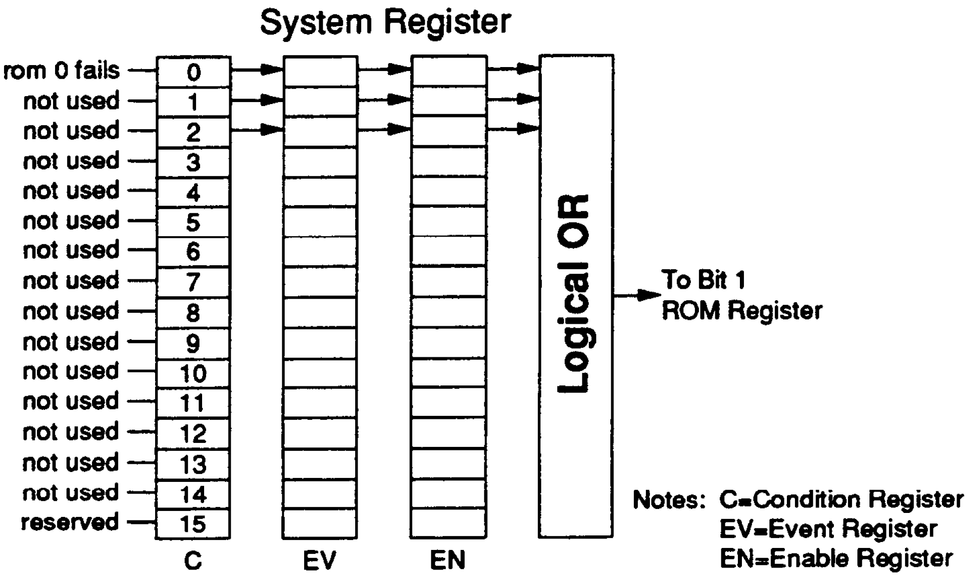


Example **Query non-volatile protected ROM event register**

`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, if calibration data is protected, and define the programming language used.

Subsystem Syntax

```
SYSTem
    :ERRor? [<mode>]
    :HEADer <mode>
    :HEADer?
    :LANGuage <command>
    :LANGuage?
    :LONGform <mode>
    :LONGform?
    :NVPRotect
        :PASSword <old>, <new>
        [:STATe] <mode>, <pass>
        [:STATe?]
    :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 HP54510A 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 54510A compatible language to instruct the Oscilloscope

```
SYST:LANG  COMP        Select compatible language
```

Comments

- **Selecting Command:** When the HP 54510A 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 54510A 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*

:NVPRotect:PASSword

SYSTem:NVPRotect:PASSword ***<old>,<new>*** is used to set a new password.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
old	discrete	Six Characters 0-9, a-z, A-Z	none
new	discrete	Six Characters 0-9, a-z, A-Z	none

Example

Change password from "SYSTEM" to "System"

```
SYST:NVPR:PASS   "SYSTEM","System"
```

Password changed

Comments

- **Password:** The password can be any string of from one to six case-sensitive alpha-numeric characters. Spaces are allowed, but special characters are not. The old password has to be specified to change to a new password.
- **Factory Default:** The oscilloscope is shipped with resistor R208 (CAL PROTECT) installed. This is the non-protected mode and defaults the password to "SYSTEM".
- **Changing the default password:** Remove resistor R208 (CAL PROTECT), then execute the :NVPRotect:PASSword command. With resistor R208 removed, the last entered password and/or protect state are maintained after a power down.
- **Unknown Password:** If resistor R208 has been removed and you do not know the password, place a 0 ohm resistor in R208 (CAL PROTECT) space on the printed circuit board. Now powering up the oscilloscope will set the factory default password to "SYSTEM" and the protect state to non-protected. Change the default password using the procedure in the above paragraph.
- **Related Commands:** SYSTem:NVPRotect[:STATe].
- **Factory Default:** Defaults to "SYSTEM".

:NVPRotect[:STATe]

SYSTem:NVPRotect[:STATe] ***<mode>,<password>*** enables (ON | 1) and/or disables (OFF | 0) the non volatile RAM protect mode. The password has to be specified for both cases.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	boolean	ON|OFF|1|0	none
password	discrete	Six Characters 0-9, a-z, A-Z	none

Example

Disabling Non-Volatile RAM protect

```
SYST:NVPR  0,"SYSTEM"   Non-volatile RAM protect to off
                        (unprotected)
```

Comments

- **Calibration:** When performing any of the firmware calibrations, the protect mode has to be OFF.
- **Password:** If the password is unknown, placing a 0Ω resistor in R208 (CAL PROTECT) space on the printed wiring board, then powering up the module will set the factory default password to "SYSTEM" and protect state to OFF. With R208 removed, the last entered password and/or protect state are maintained after a power down.
- **Related Commands:** SYSTem:NVPRotect:PASSword.
- **Factory Default:** Defaults to OFF.

:NVPRotect[:STATe]?

SYSTem:NVPRotect[:STATe]? queries the present state of the non volatile RAM protect mode. The query returns ON if the protection mode is enabled (protected) or OFF if the protection mode is disabled (unprotected). The value is sent to the output buffer. See SYSTem:NVPRotect[:STATe] command for more information.

Example

Query Non-Volatile RAM protect state

```
dimension  statement    String for data
SYST:NVPR  0,"SYSTEM"   Non-volatile RAM protect to off
                        (unprotected)
SYST:NVPR?              Query instrument to return Non-
                        volatile RAM protect enable 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 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	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.

Subsystem Syntax

```
TIMebase
    :DELay <time>
    :DELay?
    :MODE <mode>
    :MODE?
    :RANGe <range>
    :RANGe?
    :REFerence <position>
    :REFerence?
    :SAMPle
    :SAMPle?
        :CLOCk
            :IMPedance <value>
            :IMPedance?
            :LEVel <value>
            :LEVel?
            [:MODE] <value>
            :MODE?
```

: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:MODE  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:MODE TRIG`	*Set mode to triggered*
`TIM:MODE?`	*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	10 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.
- **Related Commands:** TIMebase:DELay.
- ***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
```

:SAMPle

TIMebase:SAMPle<*value*> is used to select the sampling mode of the oscilloscope. REALtime sample mode causes a complete data record to be collected (or captured) on one trigger event. REPetitive sample mode collects a complete data record over one or more trigger events.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
value	discrete	REALtime|REPetitive	None

Example

Set the Timebase sample mode to repetitive

```
TIM:SAMP  REP           Sample mode to repetitive
```

Comments

- **ACQuire:POINts :** When in the repetitive mode, the only acceptable acquisition points value is 500. When in the realtime mode, acceptable acquisition points values are 500 or 8000.
- **ACQuire:TYPE :** When in the realtime mode, the acceptable acquisition types are NORMal and RAW Data. NORMal, AVERage, and ENVelope are all available in repetitive mode.
- **Related Commands:** ACQuire:POINts, TYPE.
- ***RST Condition:** Defaults to REALtime.

:SAMPle?

TIMebase:SAMPle? returns the currently selected timebase sample mode. The data is sent to the output buffer. Returns REALtime or REPetitive depending on the current mode selected. See TIMebase:SAMPle command for more information.

Example

Querying timebase sample mode

```
dimension  statement    String to hold data
TIM:SAMP  REP           Sample mode to repetitive
TIM:SAMP?               Query instrument to return mode
enter  statement        Enter value into computer
```

:SAMPle:CLOCk:IMPedance

TIMebase:SAMPle:CLOCk:IMPedance ***<value>*** is used to select the input impedance for the external clock input at the EXT TRIG connector. The coupling can be set to DC or DCFifty.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
value	discrete	DC|DCFifty	None

Example **Set EXT TRIG sample clock input impedance to 50Ω**

```
TIM:SAMP:CLOC:IMP  DCF    Input impedance to 50Ω
```

Comments

- **Impedance:** DC is 1MΩ and DCFifty is 50Ω.
- **CLOCk:IMPedance versus TRIGger:COUPling:** Entering clock impedance changes the value currently stored in external TRIGger:COUPling.
- ***RST Condition:** Defaults to DC.
- **Related Commands:** TIMebase:SAMPle:CLOCk:IMPedance?

:SAMPle:CLOCk:IMPedance?

TIMebase:SAMPle:CLOCk:IMPedance? returns the currently selected external clock input (EXT TRIG) impedance. The data is sent to the output buffer. Returns DC for 1MΩ and DCF for 50Ω.

Example **Querying EXT TRIG sample clock input impedance**

```
Dimension  statement     String for data
TIM:SAMP:CLOC:IMP  DCF   Input impedance to 50Ω
TIM:SAMP:CLOC:IMP?       Query instrument to return
                         impedance selected
enter  statement         Enter value into computer
```

:SAMPle:CLOCk:LEVel

TIMebase:SAMPle:CLOCk:LEVel *<level>* is used to set the level voltage of the external sample clock connected to the EXT TRIG connector.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
level	numeric	See Below	V

Example

Set external sample clock level to 1 volt

```
TIM:SAMP:CLOC:LEV  1
```
External clock signal level to 1 volt

Comments

- **Selecting Level:** *level* can be entered to a value that is ±1.5 of the current CHANnel<n>:RANGe setting from the current CHANnel<n>:OFFSet setting.
- **CLOCk:LEVel versus TRIGger:LEVel:** Entering clock level changes the value currently stored in external TRIGger:LEVel.
- **Related Commands:** TIMebase:SAMPle:CLOCk:LEVel?, CHANnel<n>:RANGe , CHANnel<n>:OFFSet.
- ***RST Conditions:** Defaults to 0 volts.

:SAMPle:CLOCk:LEVel?

TIMebase:SAMPle:CLOCk:LEVel? returns the currently selected external clock level (in volts). The value is sent to the output buffer.

Example

Query the external sample clock level

`TIM:SAMP:CLOC:LEV  1`	*External clock signal level to 1 volt*
`TIM:SAMP:CLOC:LEV?`	*Query instrument to return external clock level*
`enter  statement`	*Enter data into computer*

:SAMPle:CLOCk[:MODE]

TIMebase:SAMPle:CLOCk[:MODE] ***<value>*** is used to select the internal or external sample clock

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
value	discrete	INTernal|EXTernal	None

Example **Set the sample clock to external**

```
TIM:SAMP:CLOC  EXT
```
Sample clock is signal connected to the EXT TRIG connector

Comments

- **External Clock:** An external 100 MHz ±0.005% signal must be connected to the EXT TRIG input connector. If a signal is not connected, or not within specified tolerance, then the timing accuracy will be degraded. Signal must have 50% duty cycle.
- **Power Up:** SAMPle:CLOCk is set to INTernal at Power up.
- **Changing Sample Clock:** SAMPle:CLOCk mode can only be changed when the oscilloscope is in STOP mode (not RUNning).
- **TIMebase:SAMPle:CLOCk[:MODE] versus EXTernal TRIGger:** An external sample clock cannot be selected when external trigger is desired (shared connector).
- **Related Commands:** RUN, STOP, TIMebase:SAMPle:CLOCk IMPedance, LEVel.
- ***RST Condition:** Defaults to INTernal.

:SAMPle:CLOCk[:MODE]?

TIMebase:SAMPle:CLOCk[:MODE]? returns the sample clock mode currently selected. INTernal indicates the internal sample clock is selected, and EXTernal indicates a signal connected to the EXT TRIG connector is being used as the sample clock.

Example **Querying the sample clock currently selected**

```
dimension  statement
```
String to hold data

```
TIM:SAMP:CLOC  EXT
```
Sample clock is signal connected to the EXT TRIG connector

```
TIM:SAMP:CLOC?
```
Query instrument to return sample clock selected

```
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?
    :COUPling <type>
    :COUPling ?
    :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.

:TRIGger:COUPling

TRIGger:COUPling ***<type>*** is used to select the input impedance for the EXTernal TRIGger connector. The coupling can be set to DC or DCFifty.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
type	discrete	DC|DCFifty	None

Example

Set EXT TRIG input impedance to 50Ω

```
TRIG:COUP  DCF               Input impedance to 50Ω
```

Comments

- **Impedance:** DC is 1MΩ and DCFifty is 50Ω.
- **TRIGger:COUPling versus CLOCk:IMPedance:** Entering external trigger coupling changes the value currently stored in TIMebase:SAMPle:CLOCk:IMPedance.
- ***RST Condition:** Defaults to DC.
- **Related Commands:** TRIGger:COUPling?

:TRIGger:COUPling?

TRIGger:COUPling? returns the currently selected EXTernal TRIGger connector impedance. The data is sent to the output buffer. Returns DC for 1MΩ and DCF for 50Ω.

Example

Querying EXT TRIG input impedance

```
Dimension  statement         String for data
TRIG:COUP  DCF               Input impedance to 50Ω
TRIG:COUP?                   Query instrument to return
                             impedance selected
enter  statement             Enter value into computer
```

: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 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. Source can be specified as channel 1 or 2, ECL Trigger lines 0 or 1, or the external trigger input. 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 2)| ECLTrg*n* (*n*=0 to 1)|EXTernal	none

Example

Set delay source to channel 2

```
TRIG:DEL:SOUR  CHAN2  Delay source to channel 2
```

Comments

- **Related Commands:** TRIGger:DELay, DELay:SLOPe.

:DELay:SOURce?

TRIGger:DELay:SOURce? returns the currently selected delay source (CHANnel1-2, ECLTrg0-1, or EXTernal). The data is sent to the output buffer.

Example

Query the current delay source selection

```
dimension  statement  String to hold data
TRIG:DEL:SOUR  CHAN1  Delay source to channel 1
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  TIME,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  TIME,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:** Internal trigger *level* can be entered to a value that is ±1.5 of the current CHANnel<n>:RANGe setting **from** the current CHANnel<n>:OFFSet setting. External trigger level can be entered over a range of ±2.0 V (trigger sensitivity is then 100 mVp-p into 50Ω).
- **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.
- **TRIGger:LEVel versus CLOCk:LEVel :** Entering external trigger level changes the value currently stored in TIMebase:SAMPle:CLOCk:LEVel.
- **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:LINE  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:LINE  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:MODE  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 three 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 two 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 two 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:MODE PATT`	*Pattern trigger mode enabled*
`TRIG:MODE?`	*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. Source can be specified as channel 1 or 2, ECL Trigger lines 0 or 1, or the EXTernal trigger input. 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 2)| ECLTrg*n* (*n*=0 to 1)|EXTernal	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 (CHANNel1-2, ECLTrg0-1, or EXTernal). 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. Source can be specified as channel 1 or 2, ECL Trigger lines 0 or 1, or the external trigger input. 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
source	discrete	CHANnel*n* (*n*=1 to 2)| ECLTrg*n* (*n*=0 to 1)|EXTernal	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 (CHANnel1-2, ECLTrg0-1, or EXTernal) 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 2)| ECLTrg*n* (*n*=0 to 1)| EXTernal	none

Example

Set trigger source to ECL trigger bus line 1

```
TRIG:HOLD  TIME,40NS  Holdoff to 40 nsec
TRIG:MODE  EDGE       Trigger mode to edge
TRIG:SOUR  ECLT1      Source to ECL trigger bus line 1
```

Comments

- **Selecting Source:** *source* is specified depending on the TRIGger:MODE currently selected.

  CHANnel1-2: Available on all modes except PATTern. Source is input connectors 1-2 on the instrument panel.

  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.

  EXTernal: Available on all trigger modes. Source is front panel EXT TRIG connector.

- **AUToscale:** Autoscale selects the trigger source from CHANnel1-2 only. ECLTrg lines and EXTernal trigger cannot be used for an autoscale.
- ***RST Condition:** Defaults to CHANnel1.

:SOURce?

TRIGger:SOURce? returns the selected source (CHANnel1-2 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  ECLT1      Source to ECL trigger bus line 1
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?
      :BYTeorder <type>
      :BYTeorder ?
   :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 "#41000...."

```
WAV:SOUR  WMEM1        Select waveform memory 1
WAV:DATA  #41000....   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:DATA?`	*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 (CHAN1-2), or FUNCtions (FUNC1-2) may have waveform data sent from them. Select the desired location using the WAVeform:SOURce command. Envelope data is transferred in two arrays (see WAVeform:TYPE? command for more information).
- **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.

- **RAWData Mode:** When in raw data acquisition mode, WAVeform:FORMat has no effect (data is always transferred in the WORD format).
- **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*

:FORMat:BYTeorder

WAVeform:FORMat:BYTeorder<*type*>is used to set the output sequence of the word data. MSBFirst selects the most significant byte to be transferred first. LSBFirst selects the least significant byte to be transferred first.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
type	discrete	MSBFirst|LSBFirst	None

Example

Set the output sequence to transmit the least significant byte first

```
WAV:FORM:BYT  LSBF
```
Transmit LSB first

Comments

- **WAVeform:FORMat:** BYTeorder is only used when WAVeform FORMat is set to WORD.
- **Raw Data Mode:** This command has no effect in the raw data mode.
- **Related Commands:** WAVeform:FORMat WORD.
- ***RST Condition:** Defaults to MSBFirst.

:FORMat:BYTeorder?

WAVeform:FORMat:BYTeorder? returns the currently selected word data output sequence (MSBFirst or LSBFirst) for transfer of data. Data is sent to the output buffer. See WAVeform:FORMat:BYTeorder command for more information.

Example

Querying word data output sequence

`dimension statement`	*String to hold data*
`WAV:FORM:BYT  LSBF`	*Transmit LSB first*
`WAV:FORM:BYT?`	*Query instrument to return output sequence*
`enter statement`	*Enter value into computer*

:POINts?

WAVeform:POINts? returns the points value (500 or 8000) 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 all cases the number of time buckets actually acquired will be the number of points set with the ACQuire:POINts command (500, 8000, or 4 to 8000 dependent on the TIMebase:SAMPle mode selected).
- **Related Commands:** TIMebase:SAMPle, 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 4 for RAWData 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.

- **RAW Data Mode:** In Raw Data mode, the format is always 2 for WORD, and the number of acquisitions is returned by the count parameter. In all other modes, the count parameter always returns 1.
- **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 preambles may be retrieved from the following sources: Waveform MEMories (WMEM1-4), channel buffers (CHAN1-2), or FUNCtions (FUNC1-2). 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 2)| 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.
- ***RST Condition:** Defaults to CHANnel1.

:SOURce?

WAVeform:SOURce? returns the currently selected source (CHANnel1-2, 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, NORMal, or RAWData) 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:TYPE AVER         Acquisition type to average
WAV:TYPE?             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. When 500 points are selected, t his 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 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.

Rawdata: Rawdata can only be acquired with a digitize operation, and cannot be stored in memories or measured. When all acquisitions are complete, the buffer is translated into unreconstructed, calibrated 16 bit binary data, and sent over the bus in the WORD format. The :WAVeform:FORMat command has no effect in Rawdata mode.

The command has two parameters: length and acquisitions. Length specifies the number of points of each acquisition. Acquisitions specify the number of acquisitions to be taken in a single digitize operation. The data is transferred from the oscilloscope in a single IEEE 488.2 data block consisting of two arrays.

- The first array consists of double precision 64 bit floating point numbers. This array contains the x-origin values of the waveform records that follow. It can read directly into a double precision real array in a controller allocated by the BASIC command ALLOCATE REAL Xorigins (l:Acquisitions). The x-origin values are also available in ASCII format using the WAVeform:XORigin? query.
- The second array consists of 16-bit integer numbers. This data is transmitted in a linear fashion, and starts with sample zero of the first acquisition, and continues through sample length-l of the acquisition. Then it continues in a similar fashion with sample zero through sample length-l of the each following acquisition through the last acquisition. This array can be read directly into a two dimensional integer array allocated by the BASIC command ALLOCATE INTEGER Waveforms (l:Acquisitions, l:Points).

- **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 current x-origin data point in the value in the preamble (WAVeform:PREamble?) for the current specified source. For a 500 point record, the x-origin value is a time of the first data point in the memory with respect to the trigger point. For an 8000 point record, the x-origin value is the point referenced by x-reference.

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

- **RAWData Mode:** The query may return multiple values depending on the number of acquisitions specified. In this mode, the x-origin value is the time of the first data point in each acquisition.
- **Related Commands:** WAVeform:POINts, PREamble?, SOURce.

:XREFerence?

WAVeform:XREFerence? returns the x-reference value currently in the preamble (WAVeform:PREamble?), for the current specified source set using the WAVeform:SOURce command. For a 500 point waveform in the RAW Data mode, the value is always zero. For an 8000 point record, the value represents the number of the first data point of the waveform from 0 to 7999. 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 54510A 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 :TYPE, and [SENSe:]DETector:MODE	In SCPI, average, scalar (normal), envelope, and raw data modes are selected by the average **AND** detector subsystems.
:TYPE?	[SENSe:]AVERage[:STATe]? :TYPE?, and [SENSe:]DETector:MODE?	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.
FUNCtion<n>		
:ADD	CALCulate:MATH<n>[:EXPRession]	Command operates the same in both languages.
:DIFF	CALCulate:MATH<n>[:EXPRession]	Command operates the same in both languages.
:INTegrate	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.
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.
SUMMary	STATus	The Compatible SUMMary and the SCPI STATus subsystems operate the same in both languages.

Command Cross Reference to SCPI Commands—Continued

COMP Command	SCPI Title	Description
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.
:NVPRotect	SYSTem:NVPRotect	Command operates the same in both languages.
:NVPRotect?	SYSTem:NVPRotect?	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.
:SAMPle	[SENSe:]SWEep:DETector:MODE	Command operates the same in both languages, except in realtime.
:SAMPle?	[SENSe:]SWEep:DETector:MODE?	Command operates the same in both languages, except in realtime.
TRIGger		
:COUPling	TRIGger:COUPling	Command operates the same in both languages.
:COUPling?	TRIGger:COUPling?	Command operates the same in both languages.
: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 input1-2, 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.
FORMat:BYT	TRACe:BORDer NORMal|SWAPped	Command operates the same in both languages.
FORMat:BYT?	TRACe:BORDer?	Command operates the same in both languages.
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 54510A Commands

The following table is provided as a quick cross reference of all Hewlett-Packard 54510A commands that are not supported in the HP E1428A COMPatible programming language. Any command not listed is fully functional on both the HP 54510A and the HP E1428A.

HP 54510A 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	None	Not supported by COMP.

Command Cross Reference to HP 54510A Commands—Continued

Any command not listed is fully functional on both the HP 54510A and the HP E1428A.

HP 54510A Command	COMP Command	Description
DISPlay		
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.
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 ECLTrg0-1 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 54510A 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 48) 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 48) 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 54510A Compatible Language (COMP) commands and IEEE 488.2 Common (*) commands for the HP E1428A 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, envelope, or raw data).
	: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 (1-2) for the attenuation calibration.
	:TNULl:CH1TO*<number>* ,*<time>*	Enters a time null value for a specified channel pair (1-2, 1 to EXT).
	: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 (1-2).
	: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-EXT).
	:VERTical	Configures for a vertical calibration on all channels.
	:TNULl *<value>*	Enters time null values for channel pairs 1-2.
	:TNULl?	Returns the current time null values for channel pairs 1-2.
CHANnel*<number>*	:COUPling *<type>*	Selects the specified channel's (1-2) coupling and impedance (AC 1MΩ, DC 1MΩ, or DC 50Ω).
	:COUPling?	Returns a specific channel's (1-2) current coupling selection.
	:ECL	Configures the specified channel's (1-2) vertical controls for an ECL measurement.
	:HFReject *<mode>*	Enables or disables the specified channel's (1-2) low-pass filter.
	:HFReject?	Returns a specific channel's (1-2) current low-pass filter state.
	:LFReject *<mode>*	Enables or disables the specified channel's (1-2) high-pass filter.
	:LFReject?	Returns a specific channel's (1-2) current high-pass filter state.
	:OFFSet *<value>*	Enters the specified channel's (1-2) offset.
	:OFFSet?	Returns a specific channel's (1-2) current offset value.
	:PROBe *<atten>*	Enters the specified channel's (1-2) probe attenuation factor.
	:PROBe?	Returns a specific channel's (1-2) current probe attenuation factor.
	:RANGe *<range>*	Enters the specified channel's (1-2) full scale vertical range.
	:RANGe?	Returns a specific channel's (1-2) current full scale vertical range value.
	:TTL	Configures the specified channel's (1-2) vertical controls for a TTL measurement.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
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.
	:DIFF <*source*>,<*source*>	Differentiates two specified sources and retains the result in the function number specified.
	:INTegrate <*source*>,<*source*>	Integrates 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.
	: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.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
MEASure	: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.
	: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.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
MEASure	: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.
	:WCOMpare	Waveform comparison test parameters.
	:ALLowance *<value>*	Sets the allowable comparison test deviation (in divisions).
	:ALLowance?	Returns the current allowable comparison deviation.
	:COMPare *<channel>*, *<memory>*	Selects the input channel and waveform memory pair for comparison testing.
	:COMPare?	Returns the current input channel and waveform memory pair selected for testing/comparison.
	:DESTination *<location>*	Selects the destination for comparison test violations.
	:DESTination?	Returns the waveform memory containing test violations.
	:POSTfailure *<mode>*	Specifies wether to continue or stop after a test violation.
	:POSTfailure?	Returns the current post test violation action.
	:WTESt *<mode>*	Enables or disables the waveform comparison test.

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.
	:MAP? *<source>*	Returns the address assignment and size for measurement results and internal buffers of the source specified.
	:SIZE *<bytes>*|MIN	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 Trigger Output BNC connector on the front panel.
	[:STATe] *<mode>*	Enables or disables the Trigger Output connector.
	[:STATe]?	Returns the current 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.
ROOT Commands	AUToscale	Evaluates all inputs, then sets conditions to present the signals.
	BLANk *<source>*[,*<source>* [,*<source>*[,*<source>*]]]	Turns off the specified source (channels, 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) (1-2).
	LTER?	Returns the limit test event register value. 1=violation.
	POWerup *<state>*	Selects if the oscilloscope will power up running or stopped.
	POWerup?	Returns current power up state.
	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, 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, 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 1-2, 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*>|EXTernal	Reports the specified channel's (1-2), or external trigger's 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 (1-2) 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 (1-2) 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 (1-2) 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 (1-2) or external trigger's 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*>?EXTernal	
	:OFFSet	Reports the specified channel's (1-2) 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 channel 2's or external trigger's 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 (1-2) or external trigger's 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.
	: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.

COMP Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
SUMMary	:QUEStionable	
	:TEST	
	:RAM	
	: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.
	:NVPRotect	Non-volatile RAM state and password commands.
	:PASSword *<old>*, *<new>*	Changes the current password.
	[:STATe] *<mode>*, *<pass>*	Selects the state of non-volatile RAM.
	[:STATe?]	Queries the current state of non-volatile RAM.
	: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.
	:SAMPle	Selects realtime or repetitive as the sample timebase mode.
	:SAMPle?	Returns the current sample timebase mode.
	:CLOCk	Commands for the internal or external sample clock.
	:IMPedance *<value>*	Selects the impedance for the external sample clock signal.
	:IMPedance?	Returns external sample clock impedance currently selected.
	:LEVel *<value>*	Selects the level for the external sample clock signal.
	:LEVel?	Returns external sample clock level currently selected.
	[:MODE] *<value>*	Selects internal or external sample clock.
	:MODE?	Returns sample clock currently selected.
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.
	:COUPling	Defines input trigger coupling.
	:COUPling?	Return trigger coupling 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 1-2, EXTernal, ECLTrg0-1) 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 1-2, EXTernal, ECLTrg0-1) that will be counted by the occurrence command.
	:SOURce?	Returns the occurrence source currently selected.
	:PATH *<channel>*	Selects the path (channel 1-2, EXTernal, ECLTrg0-1) 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 1-2, EXTernal, ECLTrg0-1).
	: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.
	:DATA?	Reads a binary block of waveform data from source specified.
	:DATA *<data>*	Reads a binary block of waveform data from channel 1-2, 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.
	:BYTeorder*<type>*	Selects word data output sequence.
	:BYTeorder?	Returns word data output sequence 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 48) 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 48) 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 1-2.
:AFACtor	Select the input probe attenuation factor.
:INPut<n>[:STATe]	Enable or disable inputs.<n> is the input number 1-2.
: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 1-2.
: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	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.
BORDer	NORMal	Data read MSB to LSB.
INITiate		
:CONTinuous	AUTO	Automatically sweeps.
MEASure	NULL	Reset all configurations, last measurement is invalid.
MEMory:VME		
:ADDRess	200000H	External memory address space (hex).
:SIZE	6AD14H	External memory size in bytes (hex).
:STATe	OFF	External memory disabled.
OUTPut	OFF	Output trigger 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.
DETector		
[:FUNCtion]	SAMPle	Sample mode.
:MODE	RTIMe	Realtime sample mode.
INPut	1	Input 1 on, input 2 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		
COUPling	DC	Input at DC 1MΩ.
: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, two, and external using the following procedure:

1. Set the Non-volatile RAM protection mode to OFF (if the CAL PROTECT resistor (R208) has been removed).

Note

If R208 has been removed, a <password> parameter is required when changing Non-volatile RAM protection mode. <password> default is "SYSTEM" from factory. See Chapter 6, SYSTem Subsystem for more information.

2. Connect the Oscilloscope Probe Comp/Cal/Trig Output connector to the Input 1, 2, and EXT TRIG 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 no more than 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 6, CALibrate Subsystem for more information.
7. Disconnect cables and perform Delay Calibration Procedure.

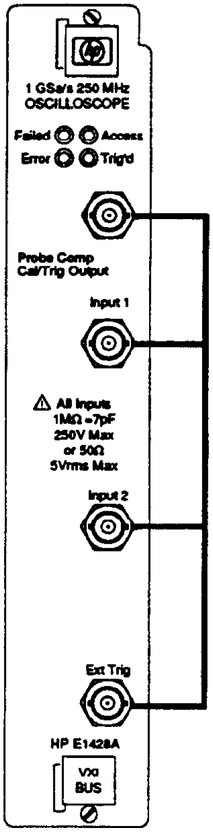


Figure 5-1. Example: Vertical Calibration Setup

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 and two (one at a time) using the following procedure:

1. Verify the Vertical Cal Procedure has been performed.
2. Connect the Oscilloscope Probe Comp/Cal/Trig 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 6, 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. When complete, disconnect BNC cable and perform Time Null Calibration Procedure.

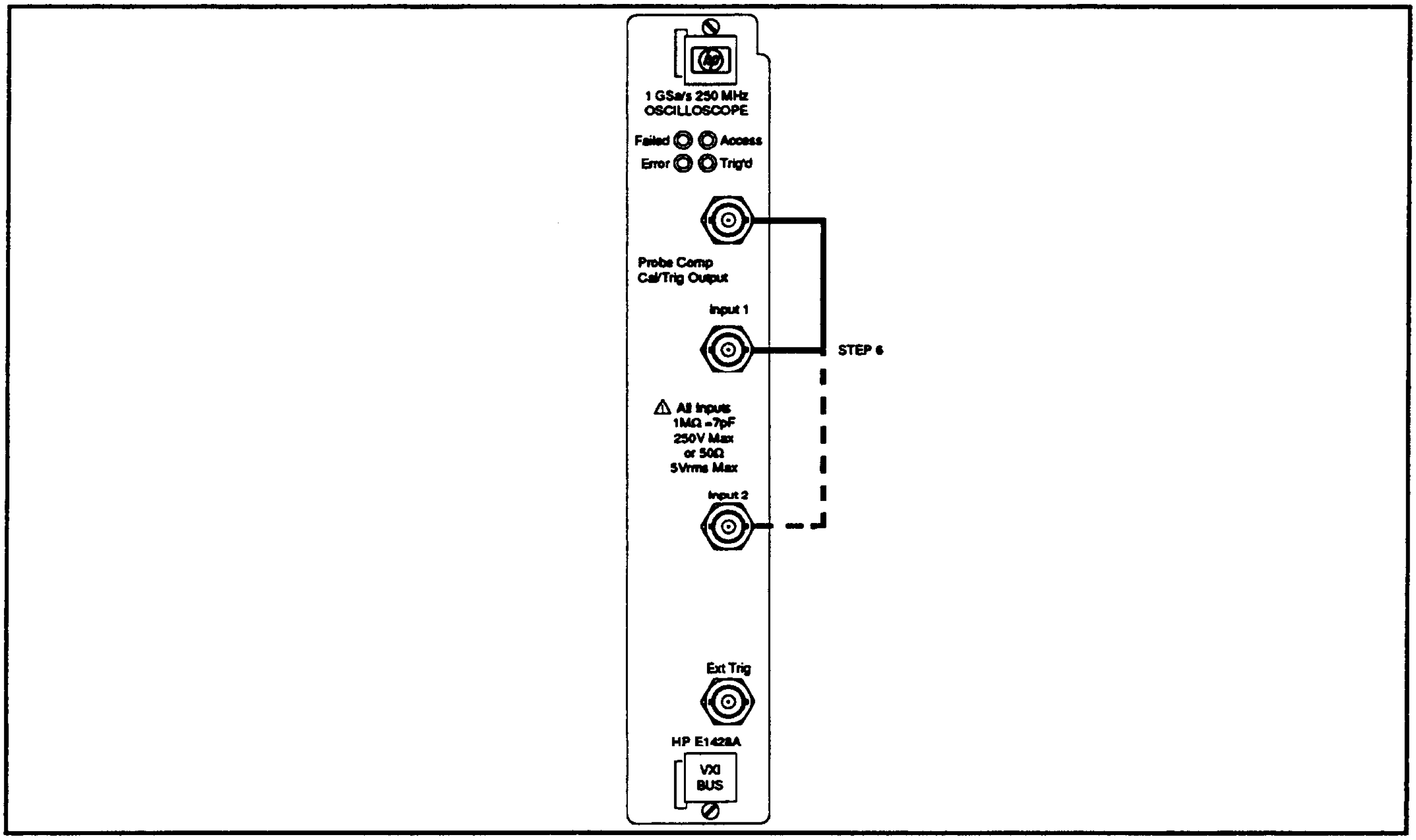


Figure 5-2. Example: Delay Calibration Setup

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 2                         Input testing loop.
 50  OUTPUT 70905;"CAL:SCAL:DEL INP"&VAL$(I)
                                          Selects input n delay calibration
                                          routine (n=input 1-2).
 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 input 2.
120  END                                  Terminate program.
```

Time Null Cal Procedure

Time Null calibration is performed on inputs one to two, and one to EXT TRIG using the following procedure:

1. Verify the Vertical Cal and Delay Cal Procedures have been performed.
2. Connect the Oscilloscope Probe Comp/Cal/Trig 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 6, CALibrate Subsystem for more information.
6. Disconnect cable from input 2 and reconnect to EXT TRIG. Repeat steps 4 and 5 for external trigger.
7. When complete, disconnect BNC cable and perform the Logic Trigger Calibration Procedure (if necessary).

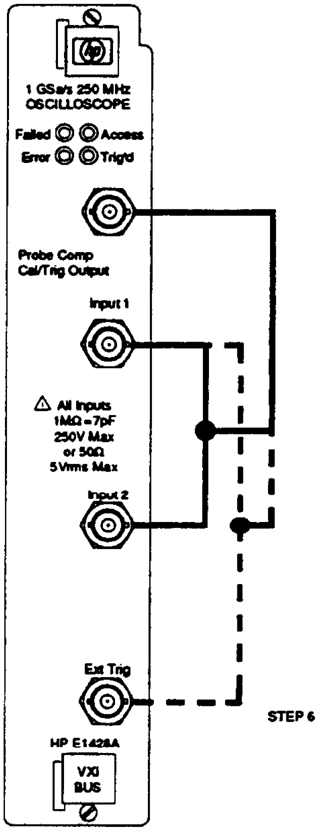


Figure 5-3. Example: Time Null Calibration Setup

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  OUTPUT 70905;"CAL:SCAL:TNUL CH1TO2"
                                         Selects input 1 to 2 time null
                                         calibration routine.
50  OUTPUT 70905;"CAL:SCAL:BCAL"         Starts time null calibration routine.
60  OUTPUT 70905;"STAT:QUES:CAL?"        Read calibration event register.
70  ENTER 70905;C                        Enter calibration event register results.
80  PRINT C                              Print calibration event register results.
90  PAUSE                                Connect cable to EXT TRIG.
100 OUTPUT 70905;"CAL:SCAL:TNUL CH1TOEXT"
                                         Selects input 1 to EXT time null
                                         calibration routine.
100 OUTPUT 70905;"CAL:SCAL:BCAL"         Starts time null calibration routine.
110 OUTPUT 70905;"SUMM:QUES:CAL?"        Read calibration event register.
120 ENTER 70905;D                        Enter calibration event register results.
130 PRINT D                              Print calibration event register results.
140. 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, and protection of the calibration data in non-volatile RAM is desired, set the Non-volatile RAM protection mode to ON (SYST:NVPR ON,<password>).

1. Verify the Vertical Cal, Delay Cal, and Time Null Cal procedures have been performed.
2. Connect the Oscilloscope Probe Comp/Cal/Trig 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 6, CALibrate Subsystem for more information.
4. Perform clear status, reset, then preset the Oscilloscope.
5. Select and start the logic trigger calibration routine. Verify that the TRIG'D LED flashes during calibration.
6. 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 6, CALibrate Subsystem for more information.
7. Disconnect BNC cable and set the Non-volatile RAM protection mode to ON (if desired and resistor R208 has been removed).

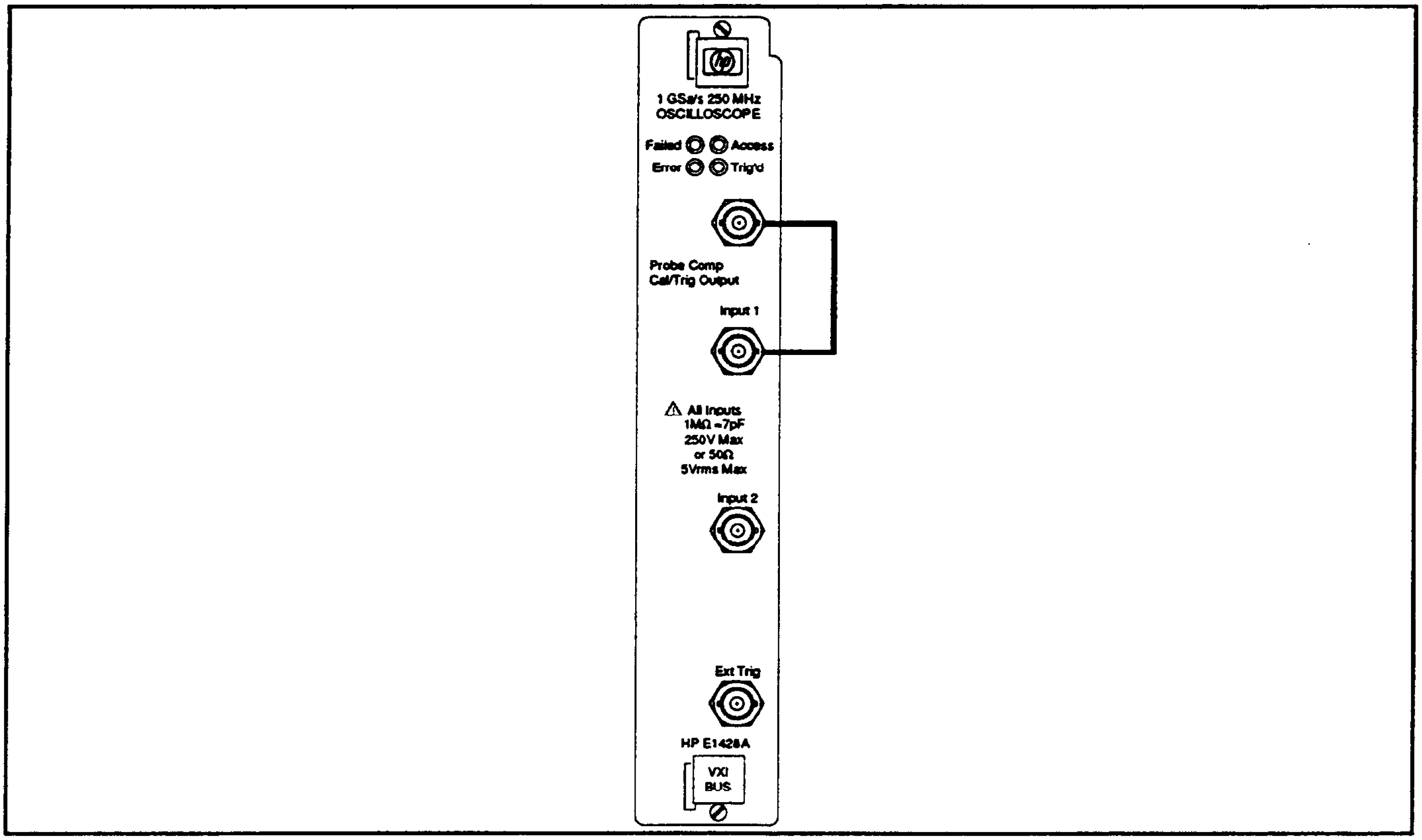


Figure 5-4. Example: Logic Trigger Calibration Setup

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;"CAL:SCAL:LTC"`	*Selects logic trigger calibration routine.*
`50 OUTPUT 70905;"CAL:SCAL:BCAL"`	*Starts logic trigger calibration routine.*
`60 OUTPUT 70905;"STAT:QUES:CAL?"`	*Read calibration event register.*
`70 ENTER 70905;E`	*Enter calibration event register results.*
`80 PRINT E`	*Print calibration event register results.*
`90 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:

```
5  DIM Results$[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: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 1-2, 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 input 2 is 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 (must be in AC coupled mode (INPn:COUP AC):

```
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 10 ns 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 1-2), from the external trigger signal (EXT TRIG), or from one of two bus lines (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, EXT, 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

Four different modes define the type of acquisition that will take place when the INITiate[:IMMediate] command is executed. See Chapter 6, [SENSe:]AVERage[:STATe], [SENSe:]AVERage:TYPE, and [SENSe:]DETector:MODE, for information on SCALar, AVERage, ENVelope, and RAWData 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)
```

AND

```
DET:MODE xxx    (xxx = RTIMe or REPetitive)
```

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 (from 1 to 2048) to be averaged for each acquisition can be entered (Average mode), or the number of times a bucket is hit (Envelope mode). Count is not effective in Raw Data mode. 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 depending on the sample mode. During repetitive mode, 500 is the only acceptable value. During realtime mode, 500 and 8000 are acceptable. 500 is selected at reset. You enter points with the following command:

```
SWE:POIN xxx    (xxx = 500 or 8000)
```

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=INP1-2, 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=INP1-2, 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 500 points have been averaged at least four times. The digitized data sent to the controller is to be in 8 bit format.

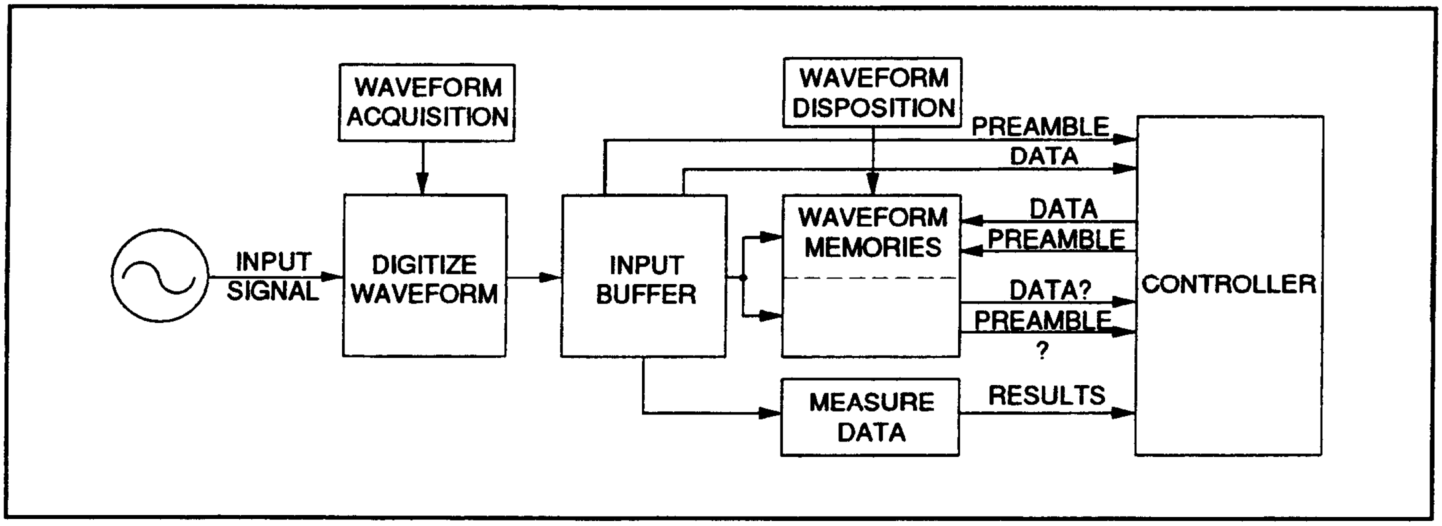


Figure 5-5. Example: Digitizing Waveforms

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.*
`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 500"`	*Data record to 500 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, followed by 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., TRACe[:DATA]?.

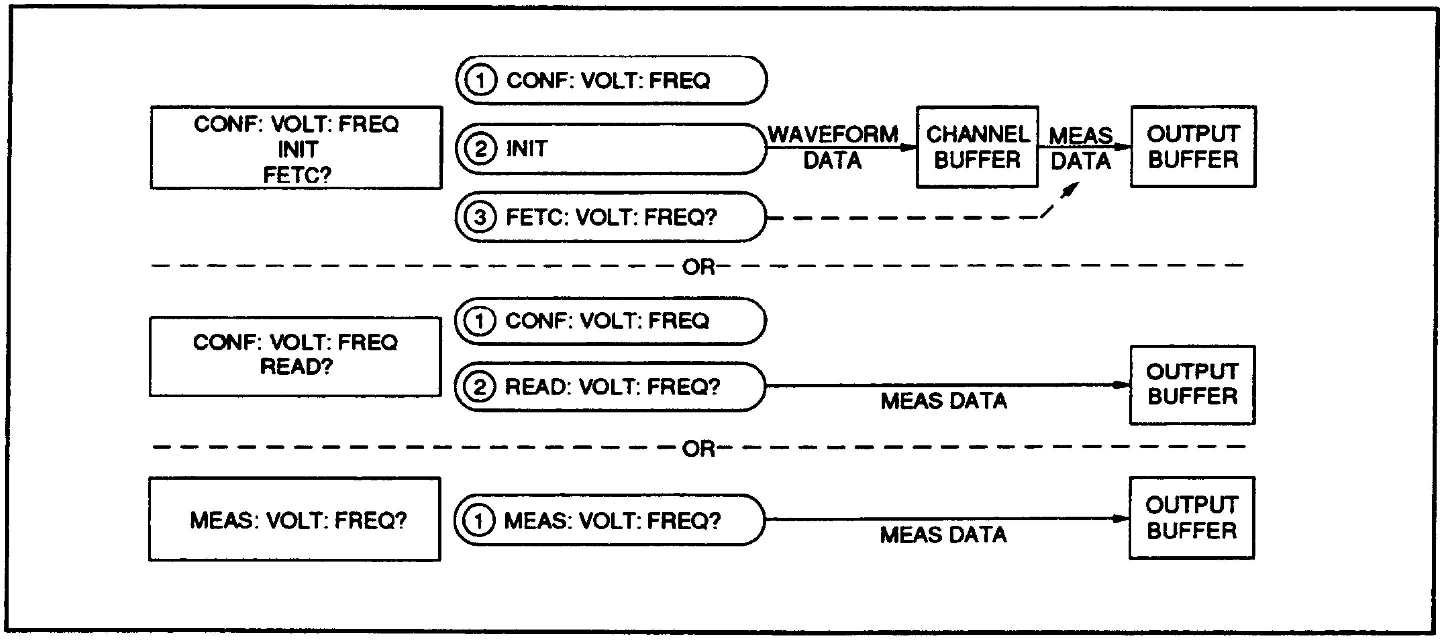
When to Read Preamble. The preamble should be read prior to the waveform data.

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 two 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 two 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 input 1.

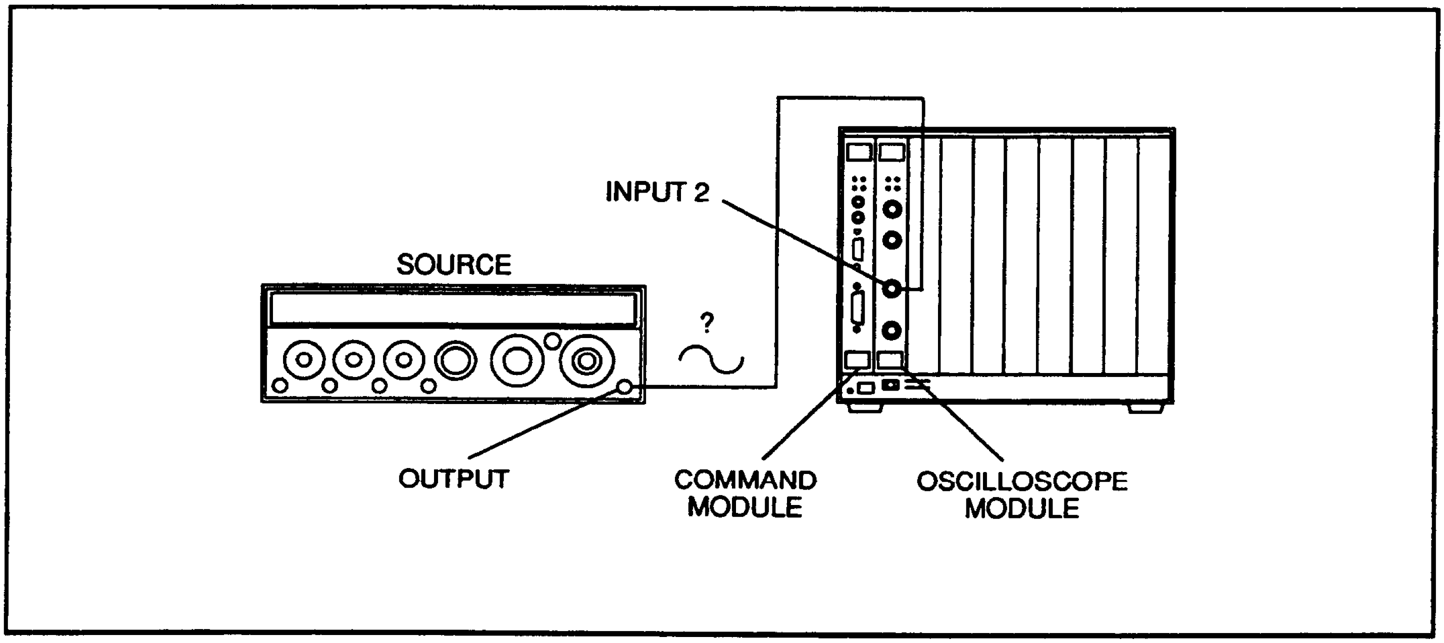


Figure 5-6. Example: Autoscale Period Measurement

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.

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 1 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.

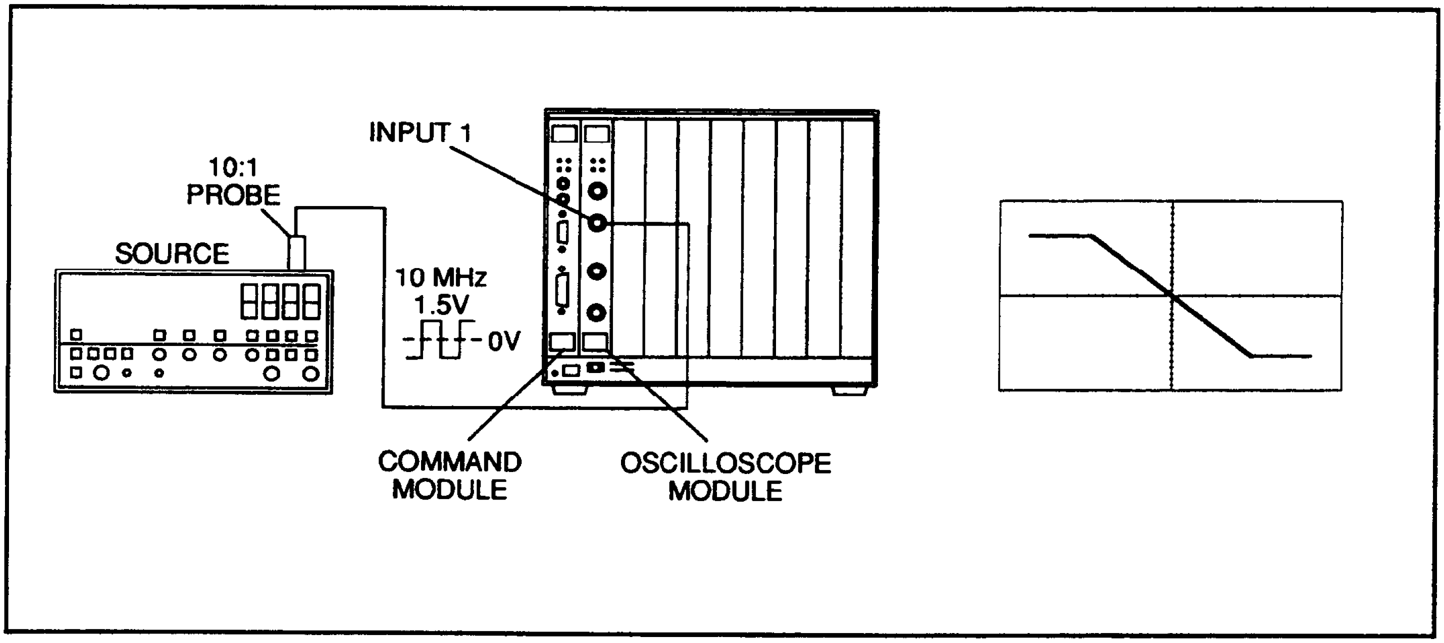


Figure 5-7. Example: Manual Rise Time Measurement

Execute:

```
 10 OUTPUT 70905;"*CLS"                    Clear status.
 20 OUTPUT 70905;"*RST"                    Resets the Oscilloscope to its
                                           default state (table 5-2).
 30 OUTPUT 70905;"CORR1:AFAC 10"           Set input 1 probe attenuation
                                           factor at 10:1.
 40 OUTPUT 70905;"VOLT1:RANG PTP 2"        Set input 1 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 INP1"          Trigger source to input 1.
 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 (@INP1)"
                                           Configure input 1 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 140              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 2 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.

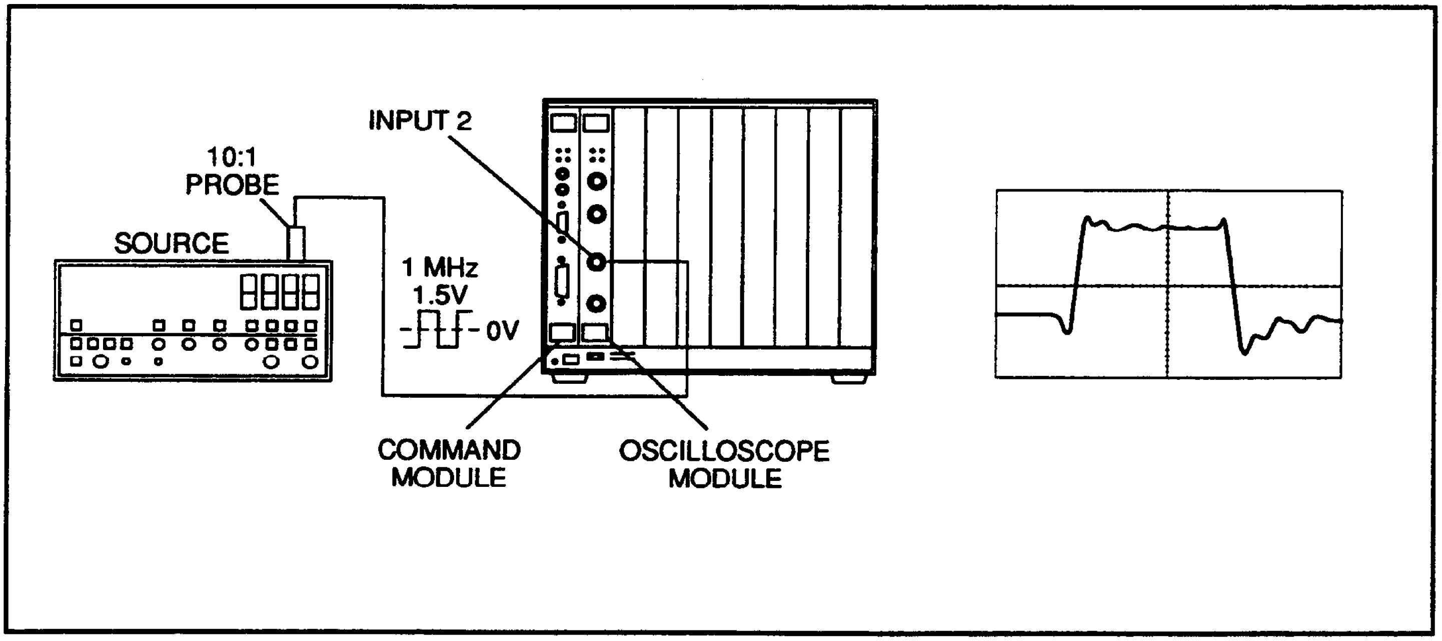


Figure 5-8. Example: Manual Rise Time and Fall Time Measurement

Execute:

```
 10 OUTPUT 70905;"*CLS"                    Clear status.
 20 OUTPUT 70905;"*RST"                    Resets the Oscilloscope to its
                                           default state (table 5-2).
 30 OUTPUT 70905;"CORR2:AFAC 10"           Set input 2 probe attenuation
                                           factor at 10:1.
 40 OUTPUT 70905;"VOLT2:RANG:PTP 2"        Set input 2 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 INP2"          Trigger source to input 2.
 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,(@INP2)"
                                           Configure input 2 for rise time
                                           measurement, with thresholds of
                                           20/80%.
 90 OUTPUT 70905;"CONF:VOLT:FTIM 20,80,(@INP2)"
                                           Configure input 2 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 1. 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 two 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 input 2.

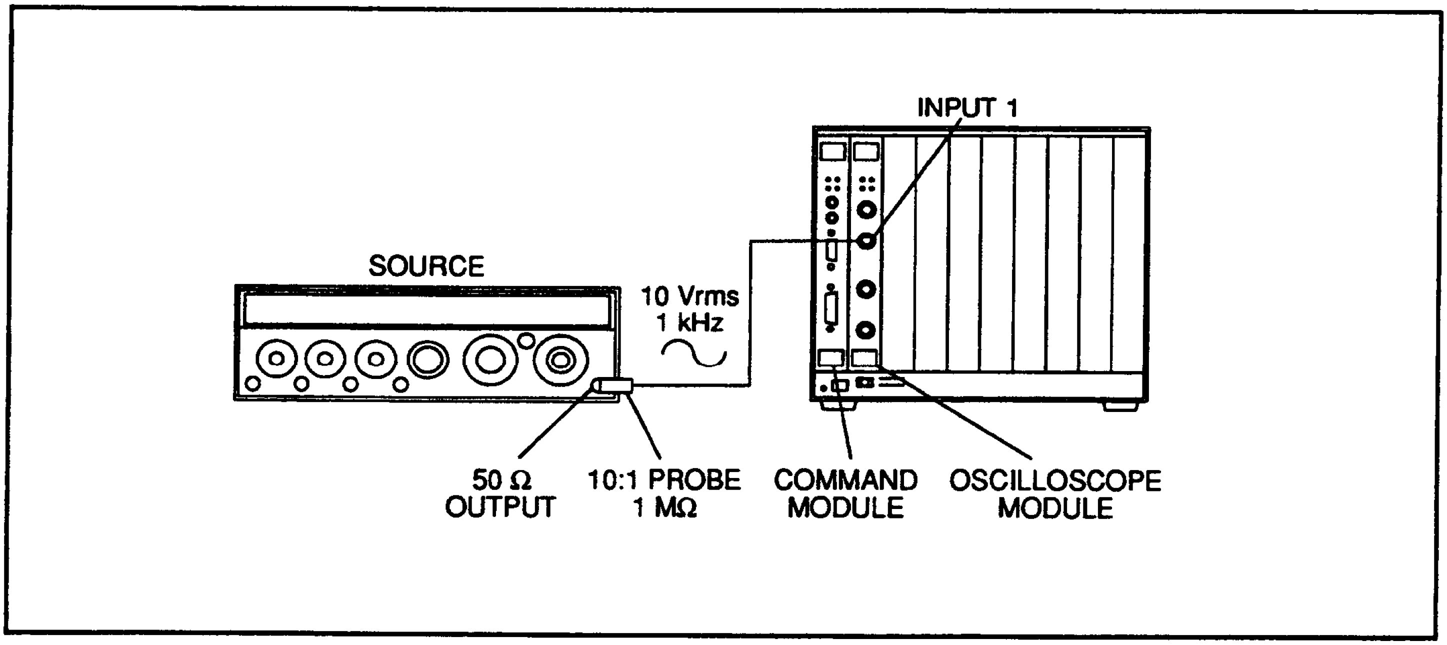


Figure 5-9. Example: AC Voltage Measurement

Execute:

```
 10  OUTPUT 70905;"*CLS"                  Clear status.
 20  OUTPUT 70905;"*RST"                  Resets the Oscilloscope to its
                                          default state (table 5-2).
 30  OUTPUT 70905;"CORR1:AFAC 10"         Set input 1 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 (@INP1)"
                                          Configure input 1 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 ECL Trigger bus line 1. The input signal is unknown.

Caution

MAXIMUM INPUT VOLTAGE. The maximum voltage that can be applied to any of the two 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 input 2.

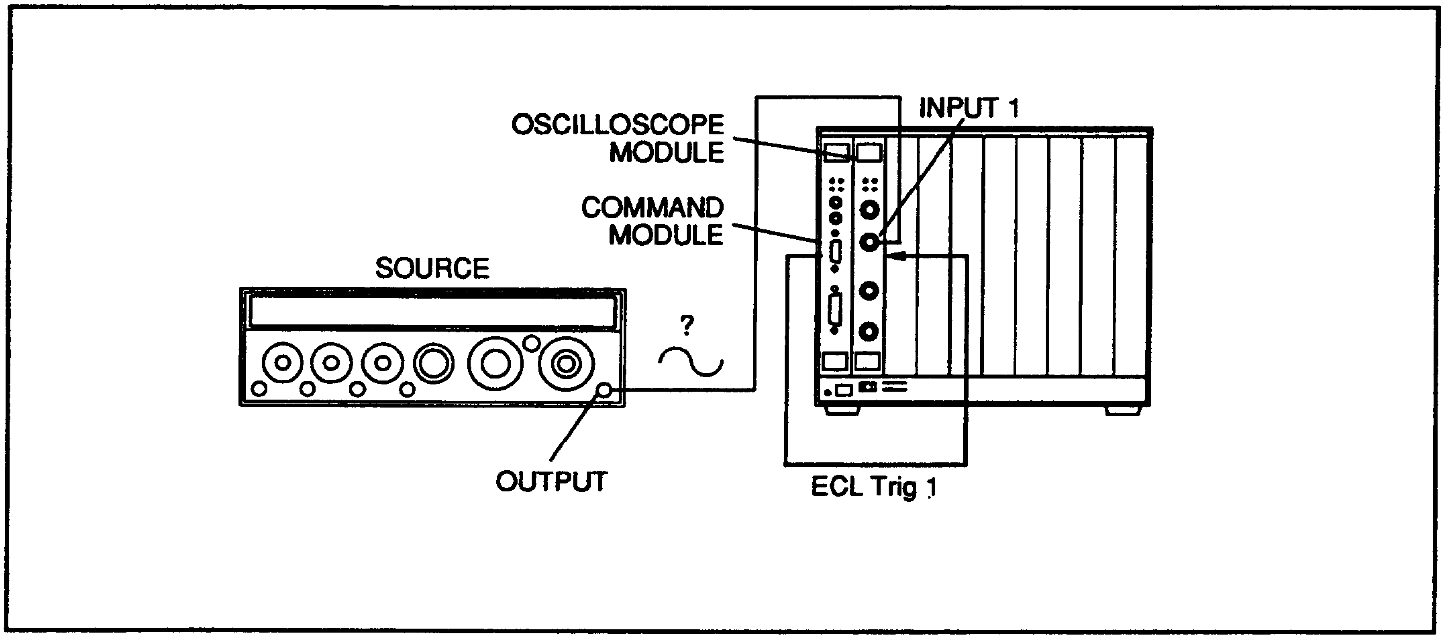


Figure 5-10. Example: Using the Backplane ECL Trigger

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 ECLT1"`	*Trigger on ECL trigger bus line 1.*
`50 OUTPUT 70905;"INIT:CONT ON"`	*Set to initiate a measurement when a trigger is received.*
`60 OUTPUT 70900;"OUTP:ECLT1:STAT ON"`	*Enables the command module to output a trigger on ECLTrg line 1.*
`70 OUTPUT 70900;"OUTP:ECLT1:SOUR INT"`	*Set the command module trigger source.*
`80 OUTPUT 70900;OUTP:ECLT1:IMM"`	*Cause a Trigger (ECL 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.

Triggering ECL Trigger line 1. Any instrument in the mainframe can send a trigger on ECLTrg line 1 using the OUTPut:ECLTrg 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 (1-48) 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 (1-48) 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

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 1-2 - active waveform from input 1-2

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 INP1-2, 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.

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 6 for more information.

Unless otherwise specified, <n> is the input number (1-2).

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.
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
:BORDer?	Order bytes are read
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

Querying the Oscilloscope — Continued

Unless otherwise specified, <n> is the input number (1-2).

Query	Description
MEASure[:SCALar]:VOLTage	
: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
: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
:MAP?	Measurement result address and size
:SIZE?	Memory size
:STATe?	Overall VME memory state

Querying the Oscilloscope — Continued

Unless otherwise specified, <n> is the input number (1-2).

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
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
DETector	Detector subsystem queries
[:FUNCtion]?	Always returns SAMPle
:MODE?	Sample mode selected
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 (1-2).

Query	Description
STATus	Status subsystem commands
:OPERation?	Operation event register value (trigger)
:QUEStionable?	Questionable event register value
:CALibration?	Calibration event register value
:INPut*<n>*|EXTernal?	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
: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
:NVPRotect?	Non-volatile RAM protect mode
: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

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?
```

Contents

Chapter 6 — Oscilloscope SCPI Command Reference

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.

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.

CALCulate

The CALCulate command subsystem defines three functions that use signals acquired on INPuts 1-2 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.

Note Calculations are only performed on 500 point records.

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 (–), multiply (*), differentiate (DIFF), or integrate (INT) two defined operands (sources) and retain the result. 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*>)| (DIFF(<*source*>)) (INT(<*source*>))	none
source	discrete	INPut*n* (*n*=1 to 2) 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.

  (DIFF (*source* | *source*)) is used to calculate the voltage differences between consecutive points in time divided by the time bucket width Δt on two defined operands.

  (INT (*source* | *source*)) is used to calculate the integral of two defined operands.
- **Specifying Source:** Inputs 1-2 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 <time>
    :REPort? <input>
    :SCALibration
        :BCALibration
        :DCALibration
        :DELay <input>
        :DOUTput <level>
        :LTCalibration
        :TNULl <input_skew>
        :VERTical
    :TNULl <value>
    :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 Probe Comp/Cal/Trig 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 2

This example calibrates the input gain on input 2. For example, a 10:1 attenuator probe is connected to the Probe Comp/Cal/Trig Output connector from the Input 2 connector.

`CAL:PCAL:ATT:INP2`	*Attenuation calibration input 2*
`pause`	*To connect probe to Probe Comp / Cal / Trig Output from Input 2 connector*
`CAL:PCAL:ATT:BCAL`	*Perform attenuation calibration. Correction automatically stored in INP2: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 2	none

Example **Set attenuation calibration input to 2**

```
CAL:PCAL:ATT:INP2
```

Attenuation calibration input to 2

Comments

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

:PCALibration:TNULl

CALibration:PCALibration:TNULl <*time*> is used to set the timing of input 2 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
time	numeric	−70NS to +70NS	S

Example **Set time null from input 1 to 2 to 25 ns.**

```
CAL:PCAL:TNUL 25E-9
```

Input 1 to 2 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 identical to the CALibration:TNULl command.
- **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 2)|EXTernal	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	**A/D X, Gain X, Offset X, Hysteresis X, Trigger X, Delay X, Time Null X**
EXTernal	**Hysteresis 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.
- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing an SCALibration routine (see SYSTem:NVPRotect).
- **Related Commands:** CAL:SCAL:VERTical, CAL:SCAL:DCALibration, CAL:SCAL:DELay, CAL:SCAL:LTCalibration, CAL:SCAL:TNULl, CAL:TNULl, STATus:QUEStionable, SYSTem:NVPRotect.

: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

- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a default calibration routine (see SYSTem:NVPRotect).
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?, SYSTem:NVPRotect.

:SCALibration:DELay

CALibration:SCALibration:DELay *<input>* performs a delay calibration on both inputs, one at a time. Each input must be connected to the front panel Probe Comp/CAL/TRIG 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 2)	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.
- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a calibration routine (see SYSTem:NVPRotect).
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?, SYSTem:NVPRotect.

:SCALibration:DOUTput

CALibration:SCALibration:DOUTput *<level>* is used to set the output level of the Probe Comp/Cal/Trig 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 Probe Comp/Cal/Trig Output connector to 5 V**

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

Comments

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

: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 two input calibration results must indicate "P" before the logic trigger calibration can be executed.
- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a calibration routine (see SYSTem:NVPRotect).
- **Calibration Results:** Logic trigger calibration results can be reviewed using the CALibration:REPort? query.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?, SYSTem:NVPRotect.

: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|INP1TOEXT	none

Example

Chapter 5 contains an example on performing a time null calibration

Comments

- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a calibration routine (see SYSTem:NVPRotect).
- **Calibration Results:** Time null calibration results can be reviewed using the CALibration:REPort? query.
- **Related Commands:** CAL:SCAL:BCALibration, CAL:REPort?, SYSTem:NVPRotect.

:SCALibration:VERTical

CALibration:SCALibration:VERTical performs a vertical calibration on both 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.
- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to performing a calibration routine (see SYSTem:NVPRotect).
- **Related Commands:** CALibration:SCALibration:BCALibration, CALibration:REPort?, SYSTem:NVPRotect.

:TNULl

CALibration:TNULl ***<value>*** is used to set the timing of input 2 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
value	numeric	−70NS to +70NS	S

Example

Set time null to 25 ns from input 1 to 2

```
CAL:TNUL  25E-9          Input 1 to 2 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 identical to the CALibration:PCALibration:ATTenuation:TNULl command.
- **Related Commands:** CAL:PCAL:TNULl.

:TNULl?

CALibration:TNULl? returns the currently selected time nulls (in seconds) for inputs 1 to 2. The data is sent to the output buffer. See CALibration:TNULl command for more information.

Example

Querying time nulls

```
CAL:TNUL  25E-9          Set input 1 to 2 time null to 25 nsec
CAL:TNUL?                Query instrument to return time nulls
enter  statement         Enter value into computer
```

Comments

- **Related Commands:** CAL:PCAL:TNULl, 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.

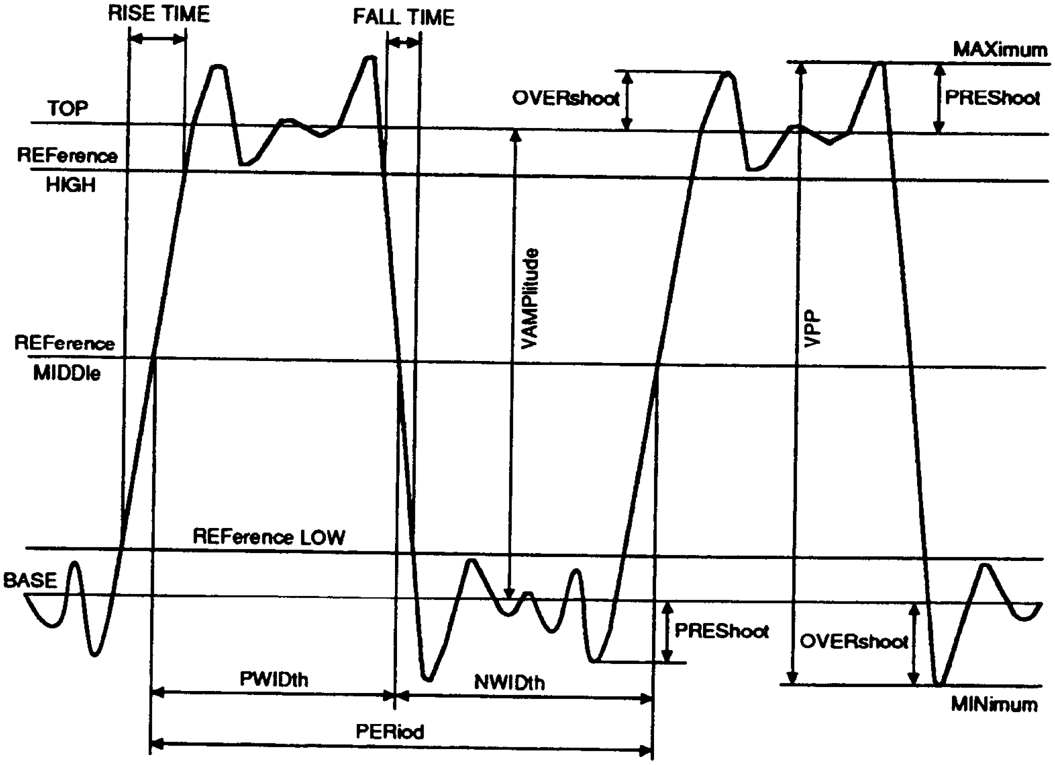
All measurements can be performed using the following methods:

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 2)| 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 2)| 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 1 for a DC voltage measurement

```
CONF:VOLT  (@INP1)
```
Configure input 1 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example **Configure input 2 for a duty cycle measurement at 40%**

```
CONF:VOLT:DCYC  40,(@INP2)
```

Configure input 2 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 2)| 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 1 for a preshoot measurement on the falling edge of the waveform

```
CONF:VOLT:FALL:PRES   (@INP1)
```
Configure input 1 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 2)| 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 2)| 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 2)| 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 2)| 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 1 for a low voltage measurement

```
CONF:VOLT:LOW  (@INP1)
```
Configure input 1 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 2 for a maximum voltage measurement

```
CONF:VOLT:MAX   (@INP2)
```

Configure input 2 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 2)| 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 2)| 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example **Configure input 1 for a negative pulse width measurement**

```
CONF:VOLT:NWID   (@INP1)
```
Configure input 1 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 2)| 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 2)| 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 2)| 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 1 for an overshoot measurement on the rising edge of the waveform

```
CONF:VOLT:RISE:OVER   (@INP1)
```
Configure input 1 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 2 for a preshoot measurement on the rising edge of the waveform

```
CONF:VOLT:RISE:PRES    (@INP2)
```

Configure input 2 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 2)| 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 2)| 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 1 for a first maximum voltage time measurement

```
CONF:VOLT:TMAX   (@INP1)  Configure input 1 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Configure input 2 for a first minimum voltage time measurement

```
CONF:VOLT:TMIN  (@INP2)
```
Configure input 2 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 (@1)`	*Configure input 1 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
    :BORDer <type>
    :BORDer?
    [:DATA] <type>,<format>
    [:DATA]?
```

:BORDer

WAVeform:FORMat:BORDer*<type>*is used to set the output sequence of the word data. NORMal selects the most significant byte to be transferred first. SWAPped selects the least significant byte to be transferred first.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
type	discrete	NORMal|SWAPped	None

Example

Set the output sequence to transmit the least significant byte first

```
FORM:BORD   SWAP
```
Transmit LSB first

Comments

- **WAVeform:FORMat:** BORDer is only used when FORMat[:DATA] is set to 16.
- **Raw Data Mode:** This command has no effect in the raw data mode.
- **Related Commands:** FORMat[:DATA].
- ***RST Condition:** Defaults to NORMal.

:BORDer?

WAVeform:FORMat:BORDer? returns the currently selected word data output sequence (NORMal or SWAPped) for transfer of data. Data is sent to the output buffer. See FORMat:BORDer command for more information.

Example

Querying word data output sequence

`dimension statement`	*String to hold data*
`FORM:BORD SWAP`	*Transmit LSB first*
`FORM:BORD?`	*Query instrument to return output sequence*
`enter statement`	*Enter value into computer*

[: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	discrete	INTeger	none
format	numeric	8 or 16	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 WORD format is 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.

- **RAWData Mode:** When in raw data acquisition mode, FORMat[:DATA] has no effect (data is always transferred in the 16 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).
- **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 1 one time and save in the input 1 buffer**

```
INP1  ON                Enable input 1
INIT:CONT  OFF          Acquire data once
INIT                    Input 1 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.

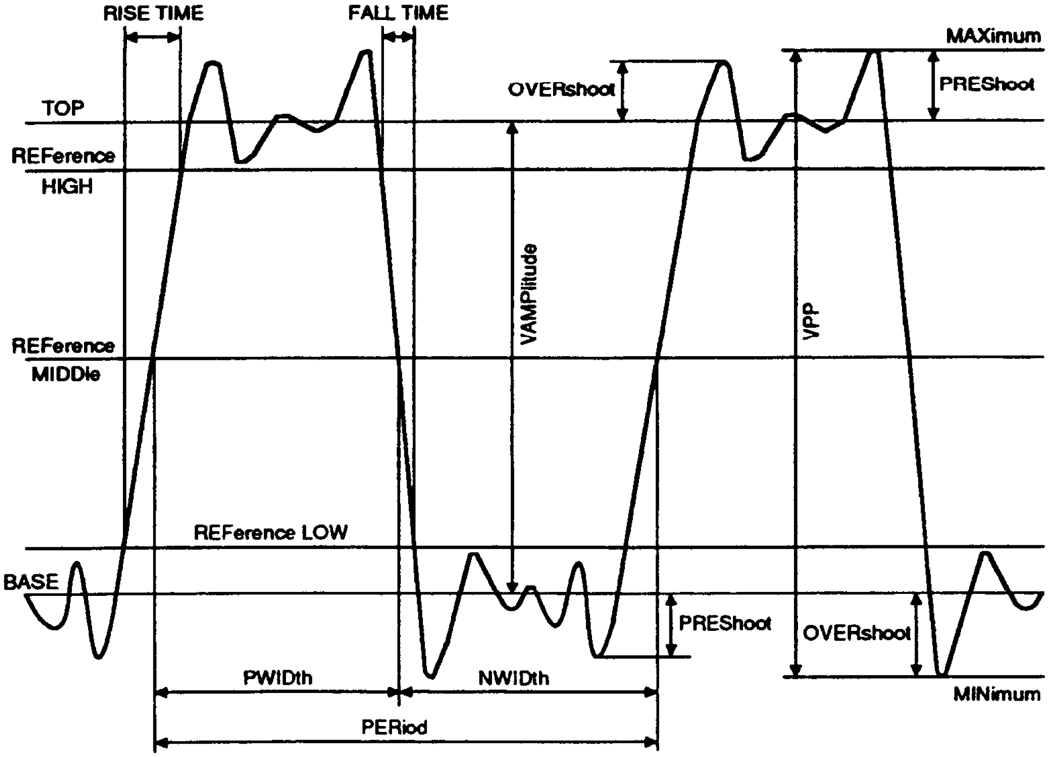
All measurements can be performed using the following methods:

> 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 2)| 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 2)| 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a DC voltage measurement on input 1

`MEAS:VOLT? (@INP1)`	*Configure input 1 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 *channel_list*	numeric numeric	0 to 100 INPut*n* (*n*=1 to 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	PCT 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 2)| 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 2)| 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 1

```
MEAS:VOLT:FALL:PRES?  (@INP1)
enter  statement
```

`MEAS:VOLT:FALL:PRES?  (@INP1)`	*Configure input 1 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 2)| 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 2)| 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 2)| 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 2)| 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 2)| 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a maximum voltage measurement on input 2

```
MEAS:VOLT:MAX  (@INP2)
```

Configure input 2 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 2)| 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?         Configure input 1 for a Minimum
                       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:** 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 2)| 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	numeric	0 to 100	PCT
channel_list	numeric	INPut*n* (*n*=1 to 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a negative pulse width measurement on input 1

```
MEAS:VOLT:NWID?  (@INP1)
```

Configure input 1 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 2)| WMEMory*n* (*n*=1 to 4)| MATH*n* (*n*=1 to 2)	none

Example

Make a duty cycle measurement on input 2

```
MEAS:VOLT:PDUT?  (@INP2)   Configure input 2 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 2)| 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 2)| 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 2)| 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 2)| 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 2

```
MEAS:VOLT:RISE:PRES?  (@INP2)
```

Configure input 2 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 2)| 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 2)| 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 2)| 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 1

```
MEAS:VOLT:TMIN?   (@INP1)
```

Configure input 1 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.

Subsystem Syntax

```
MEMory
    :VME
        :ADDRess <address>
        :ADDRess? [MINimum|MAXimum]
        :MAP?
        :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 sample mode, acquisition type, and format selected using the following guidelines:

- **DETector:MODE REPetitive / 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 (CHANnel1) = MEM:VME:ADDR + #H016C (364 dec)
  INPUT2 (CHANnel2) = MEM:VME:ADDR + #H0940 (2368 dec)

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 = data bits. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the WAVeform:PREamble? query.

- **DETector:MODE REPetitive / Acquisition TYPE AVERage** — The acquired data is 16 bits in length (stored in a 32 bit word). The starting location, in VME space, of each INPUT buffer is to be calculated as follows:

  INPUT1 (CHANnel1) = MEM:VME:ADDR + #H016C (364 dec)
  INPUT2 (CHANnel2) = MEM:VME:ADDR + #H0940 (2368 dec)

Data — 32 bits

```
                    33222222222211111111110000000000
                    10987654321098765432109876543210
                            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 = data bits. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the WAVeform:PREamble? query.

- **DETector:MODE REPetitive / 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 (CHANnel1) = MEM:VME:ADDR + #H016C (364 dec)
  INPUT2 (CHANnel2) = MEM:VME:ADDR + #H0940 (2368 dec)

  MAXIMUM

  INPUT1 (CHANnel1) = MEM:VME:ADDR + #H0556 (1366 dec)
  INPUT2 (CHANnel2) = MEM:VME:ADDR + #H0DA2 (3490 dec)

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 = data bits. The external buffers only contain the acquired waveform data, not preamble information. Preamble information can be read using the WAVeform:PREamble? query.

- **DETector:MODE RTIMe,<points>,<acquisition>** (or real-time mode and raw data acquisition type) – The acquired data is 16 bits length. The starting location, in VME space, of each INPUT buffer is calculated as follows:

  SEQUential1 (CHANnel1) = MEM:VME:ADDR + #H9294 (37524 dec)
  SEQUential2 (CHANnel2) = MEM:VME:ADDR + #H39FD4 (237524 dec)

Note: The above calculations are based on the default size of #H6AD14 (437524 dec). If the size is increased the location for SEQUential2 will change due the fact that both SEQ1 and SEQ2 are equally incremented based on available size.

It is recommended for any size to use the MEM:VME:MAP? query to determine exact addresses in VME space.

The data block consists of two arrays. The first array consists of double precision 64-bit floating point numbers. These are the xorigin values of the waveform records to follow. The second array consists of 16-bit integers and represents the actual data for each acquistion.

Double precision format for xorgs:

The most significant bit is the sign bit, the next 11 most significant bits are the exponent field, and the remaining 52 bits are the fraction field. The bias of the exponent is 1023.

```
63      62----------52          51---------------0
S       Exp.+Bias               Fraction
```

Data — 16 Bits

```
1111110000000000
5432109876543210
xxxxxxxxxxxxxxxx
```

```
      WORD
15                   0
x x x x x x x x x x x x x x x x >
      shift right 1 bit
```

*Note: x = data bits.

- **DETector:MODE RTIMe** (or real-time mode and normal acquisition type) – — The acquired data is 16 bits in length. The starting location, in VME space, of each INPUT buffer is to be calculated as follows:

Length = 500 points

```
INPUT1(CHANnel1) = MEM:VME:ADDR + #H016C (364 dec)
INPUT2(CHANnel2) = MEM:VME:ADDR + #H0940 (2368 dec)
```

Length = 8000 points

```
INPUT1(CHANnel1) = MEM:VME:ADDR + Dynamic
INPUT2(CHANnel2) = MEM:VME:ADDR + Dynamic
```

For realtime sampling (DETector:MODE RTIMe), 8000 points, it is necessary to query with MEMory:VME:MAP? every time the time/div setting is changed.

Through a query, the starting address of the memory space can be determined for real time, 8000 points at a particular time/division (use BUF8000_1 for CHANnel1, BUF8000_2 for CHANnel2.)

Example in HP BASIC to find memory location for CHANnel1:

```
OUTPUT @Scope; "MEMory:VME:MAP? BUF8000_1"
ENTER @Scope; Memory_loc
PRINT "Memory location is "; Memory_loc
```

Data — 16 Bits

```
1111110000000000
5432109876543210
xxxxxxxxxxxxxxxx
```

```
        BYTE                      WORD                     COMPRESSED

15                    0  15                    0  15                    0
x x x x x x x - - - - - - - - >  x x x x x x x x x x x x x x x x >  x x x x x x x x x x - - - - - >
    shift right 9 bits          shift right 1 bit          shift right 8 bits
```

*NOTE: x = data bits. 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:MAP?

MEMory:VME:MAP?[<*source*>] returns the address assignment and size for measurement results and internal buffers, of the source specified, to the output buffer.

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	BUF500_*n* (n=1 to 2)| BUF8000_*n* (n=1 to 2)| SEQUential*n* (n=1 to 2)| MRESults	none

Example **Querying the VME memory map**

`dimension statement`	*Dimension computer string array*
`MEM:VME:MAP?`	*Query instrument to return size (see comments below)*
`enter statement`	*Enter string into computer*

Comments

- **Returned Formats:** The following responses are based on VME address set to #H00200000 and the size set to #H06AD14 (default):

MEMory:VME:MAP? MRES

after :MEAS:SCRATCH	*"MRES no measurements"*
after :MEAS:FREQ?	*"MRES frequency at #H0020000C"*
after: :MEAS:FREQ?;PER?	*"MRES frequency at #H0020000C;PER at #H00200030"*

MEMory:VME:MAP? BUFF500_1

"BUFF500_,#H0020016C,#000007D4"

MEMory:VME:MAP? SEQUential1

"SEQU1,#H00209294,#00030D40"

MEMory:VME:MAP?

after: :MEAS:SCRATCH

```
"MRES no measurements;
                              BUFF500_1,#H0020016C,#000007D4;
                              BUFF500_2,#H00200940,#000007D4;
                              BUFF8000_1,#H00201114,#000040C0;
                              BUFF8000_2,#H0020514D,#000040C0;
                              SEQU1,#H00209294,#00030D40;
                              SEQU2,#H00239FD4,#00030D40;"
```

after: :MEAS:FREQ?;PER?

```
"MRES frequency at #H0020000C;PER at #H00200030;
                              BUFF500_1,#H0020016C,#000007D4;
                              BUFF500_2,#H00200940,#000007D4;
                              BUFF8000_1,#H00201114,#000040C0;
                              BUFF8000_2,#H0020514D,#000040C0;
                              SEQU1,#H00209294,#00030D40;
                              SEQU2,#H00239FD4,#00030D40;"
```

after: :MEAS:FREQ?;PER? and MEAS:STAT ON

```
"MRES frequency at #H0020000C, MIN at #H00200010,
MAX at #H00200014;PER at #H00200030 MIN at
#H00200034, MAX at #H00200038;
                              BUFF500_1,#H0020016C,#000007D4;
                              BUFF500_2,#H00200940,#000007D4;
                              BUFF8000_1,#H00201114,#000040C0;
                              BUFF8000_2,#H0020514D,#000040C0;
                              SEQU1,#H00209294,#00030D40;
                              SEQU2,#H00239FD4,#00030D40;"
```

If any values either from the measurements or the buffers are read directly thru the VXI backplane, they have to be accessed at these addresses.

:VME:SIZE

MEMory:VME:SIZE <*bytes*> sets the size, in bytes, of the available VME address space. The oscilloscope has 1 Mbyte of internal shared memory.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
bytes	numeric	#H92B4 through #HC00000|MIN	bytes

Example **Setting the VME memory size**

```
MEM:VME:SIZE  64000    Set memory size to 64000
```

Comments

- **Entering Size:** Memory size can be specified as a value in decimal or hexadecimal (#H....), or as MINimum (#H3D090 or 250,000 dec).

  The absolute minimum size of VME memory required is based on the minimum data size in the in the RAW Data mode (4 points and 1 acquisition) as follows:.

meas_queue	= #H016C (364 dec)
buff_500_1	= #H07D4 (2004 dec)
buff_500_2 =	#H07D4 (2004 dec)
buff_8000_1 =	#H40C0 (16576 dec)
buff_8000_2 =	#H40C0 (16576 dec)
sequ1 =	#H0010 (16 dec)
sequ2 =	#H0010 (16 dec)
Total =	#H92B4 (37556 dec)

- **Internal Buffer Size:** The internal buffer size of the oscilloscope is #H6AD14 (default value).
- **Maximum Memory Size:** The maximum memory size available is DF7FFE (14647294 dec).
- ***RST Condition:** MEM:VME:SIZE defaults to #H6AD14.

: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 64,000
                             Bytes
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 64,000 Bytes
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.
- ***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), or the "Probe Comp/Cal/Trig" BNC port on the Oscilloscope front panel.

OUTPut[:STATe] acts like the master switch for the OUTPut subsystem. If the ECLTrg, 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]?
```

: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:** Both trigger outputs, ECLTrg (0 or 1) and EXTernal, 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 EXTernal and ECLTrg2, send the following commands:

  ```
  OUTP:EXT 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 "Probe Comp/Cal/Trig 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 Probe Comp/Cal/Trig output connector

`OUTP:EXT 1`	*Enable "Probe Comp / Cal / Trig Output" BNC port to output pulse*
`OUTP 1`	*Enable output subsystem*

Comments

- **Enabling Trig Out Port:** When enabled, a pulse is output from the "Probe Comp/Cal/Trig 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:** Both trigger outputs, ECLTrg (0 or 1) and EXTernal, 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 EXTernal and ECLTrg2, send the following commands:

  ```
  OUTP:EXT 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 "Probe Comp/Cal/Trig 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 Probe Comp/Cal/Trig output connector enable state

`dimension statement`	*String for data*
`OUTP:EXT ON`	*Enable "Probe Comp / Cal / Trig 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 ECLTrg and EXTernal outputs.

Parameters

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

Example

Enabling Probe Comp/Cal/Trig output connector

`OUTP:EXT 1`	*Enable "Probe Comp / Cal / Trig Output" BNC port to output pulse*
`OUTP 1`	*Enable output subsystem*

Comments

- **Selecting Outputs:** Use the 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*

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 (@1)`	*Configure input 1 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?
    DETector
        [:FUNCtion] <type>
        [:FUNCtion]?
        :MODE <mode> [,<points>][,<acquisition>]]
        :MODE?
    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>
        :POINts?
            :COMPlete <complete>
            :COMPlete?
        :TIME:
            :CENTer <center_time>
            :CENTer?
            :DELay <time>
            :DELay?
                :LINK <reference>
                :LINK?
            :RANGe <range>
            :RANGe?
```

Subsystem Syntax

```
[SENSe:]—Continued
    SWEep—Continued
        :TIME:—Continued
            :SPAN <span>
            :SPAN?
            :STARt <start_time>
            :STARt?
            :STOP <stop_time>
            :STOP?
    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, in AVERage mode, for each time bucket before the acquisition (for that time bucket) is considered complete. In ENVelope mode, count selects the number of times the time bucket is hit.

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. The acceptable values are from 1 to 2048, however entry will be rounded to the nearest power of 2. Query will return entered value.
- **[SENSe:]DETector:MODE:** AVERage:COUNt is only used in REPetitive mode.
- **Related Commands:** [SENSe:]AVERage:TYPE, DETector:MODE, [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 average mode is enabled or OFF if 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, AVERage[:STATe], and DETector:MODE 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. DETector:MODE must be set to or repetitive (REP).

  SCALar Mode: The Scalar acquisition mode is selected when AVERage[:STATe] is disabled (OFF | 0), and AVERage:TYPE is SCALar. DETector:MODE can be set to realtime (RTIMe) or repetitive (REP) as required.

  ENVelope Mode: The Envelope acquisition mode is selected when AVERage[:STATe] is disabled (OFF | 0), and AVERage:TYPE is ENVelope. DETector:MODE must be set to repetitive (REP).

  RAWData Mode: The RAWData acquisition mode is selected when DETector:MODE is in realtime (RTIMe) mode, and <length>,<acquisitions> parameters as set to 500,1. AVERage:TYPE and AVERage[:STATe] have no effect in this mode.

- **Why Four 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.

RAWData: Used to return 16-bit binary data over the bus (in WORD format). See DETector:MODE command for more information.

- **Related Commands:** AVERage:COUNt, INITiate[:IMMediate], DETector:MODE.
- ***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 and 2 are independently programmable.

Parameters

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

Comments

- **Entering Number:** The desired input *number* from 1 to 2 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 2	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 2	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
```

DETector

[SENSe:]DETector subsystem is used to select the sampling mode of the oscilloscope, and the RAW data mode.

DETector[:FUNCtion]

[SENSe:]DETector[:FUNCtion] SAMPle is always set to SAMPle. This command has no effect on instrument operations, and is only included for compatibility with other instruments.

DETector[:FUNCtion]?

[SENSe:]DETector[:FUNCtion]? always returns SAMPle. This query has no effect on instrument operations, and is only included for compatibility with other instruments.

DETector:MODE

[SENSe:]DETector:MODE *<mode>*[,*<length>*][,*<acquisitions>*] is used to select the timebase sampling mode of the oscilloscope. Also used to select RAWData acquisition type when length and number of acquisitions are specified.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode	discrete	RREPetitive|RTIMe	none
length	numeric	4 to 8000	none
acquisitions	numeric	See Determining Acquisitions below	none

Example

Set timebase sampling mode to realtime and acquisition type to RAWData

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

```
DET:MODE  RTIM,500,1
```
Timebase sample is realtime, acquisition type is RAWData

Comments

- **Selecting Mode:** Realtime sample mode causes a complete data record to be collected (or captured) on one trigger event. RandomREPetitive sample mode collects a complete data record over multiple trigger events.
- **Selecting Acquisition Mode:** Acquisition mode is selected using the AVERage:TYPE, AVERage[:STATe], and DETector:MODE 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. DETector:MODE must be set to or repetitive (REP).

SCALar Mode: The Scalar acquisition mode is selected when AVERage[:STATe] is disabled (OFF | 0), and AVERage:TYPE is SCALar. DETector:MODE can be set to realtime (RTIMe) or repetitive (REP) as required.

ENVelope Mode: The Envelope acquisition mode is selected when AVERage[:STATe] is disabled (OFF | 0), and AVERage:TYPE is ENVelope. DETector:MODE must be set to repetitive (REP).

RAWData Mode: The RAWData acquisition mode is selected when DETector:MODE is in realtime (RTIMe) mode, and (as a minimum) the <length> parameter is set from 4 to 8000. Optional <acquisitions> parameter can be specified if desired. AVERage:TYPE and AVERage[:STATe] have no effect in this mode.

- **Why Four 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. Refer AVERage:TYPE command for more information.

  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. Refer AVERage:TYPE command for more information.

  ENVelope: Used when measuring voltage or time jitter. The waveform reflects the minimum and maximum data points (hit) in each time bucket. Refer AVERage:TYPE command for more information.

  RAWData: Used to return 16-bit binary data over the bus (in WORD format). When selected, the number of points for each acquisition (*length*), and number of acquisitions to be taken in a single digitize operation (*acquisition*) are specified. Data cannot be stored in waveform memories. WAVeform:FORMat have no effect in this mode. This mode is exited by selecting another AVERage:TYPE.

- **Entering Length:** In RAWData mode, the number of points for each acquisition (from 4 to 8000) can be specified as *length*. This optional parameter can be entered with or without specifying *acquisition*.

  To receive a correct TRACe:PREamble? with the corresponding 8000 point realtime acquisition, send the INITiate[:IMMediate] or ABORt command before sending the TRACe[:DATA]? command

- **Determining Acquisition:** In RAWData mode, the maximum number of acquisitions taken during a digitize operation INITiate[:IMMediate] are determined by the *length* specified, and the limited buffer size available (200,000 bytes per channel). The number of acquisitions must satisfy the following condition:

  buffer_size = [(*length2) +8]****acquisitions***

  Where:
  length (entered) is from 4 to 8000
  buffer_size = 200,000 (maximum)

Example

Determine how many acquisitions can be made on channel 1 at 500 points (length) as follows:

200,000 = [(2*500) + 8] * acquisitions

$$\text{acquisitions} = \frac{200\,000}{1008} = 198$$

Each data point takes 2 bytes, and the time value associated with each acquisition takes 8 bytes. If this optional parameter is used, *length* must also be specified.

Note

In shared memory, the buffer size is determined by MEMory:VME:SIZE.

- **Related Commands:** ABORt, AVERage:TYPE, INITiate[:IMMediate], DETector:MODE?.
- ***RST Condition:** Defaults to RTIMe.

:MODE?

DETector:MODE? returns the currently selected sampling mode, and if applicable, length and number of acquisitions. The data is sent to the output buffer. See DETector MODE command for more information.

Example

Querying acquire type

`Dimension statement`	*String for data*
`DET:MODE RTIM,500,1`	*Timebase sample is realtime, acquisition type is RAWData*
`DET:MODE?`	*Query instrument to return sample mode, length, and number of acquisitions (as required)*
`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 and 2 are independently programmable.

Parameters

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

Comments

- **Entering Number:** The desired input *number* from 1 to 2 is specified for the input commands listed in this section.

INPut:COUPling

[SENSe:]INPut<*number*>:COUPling <*type*> is used to select 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 2 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.
- **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 2	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 *mode*	numeric boolean	1 to 2 OFF|0|ON|1	none none

Example **Enable high pass filter on input 1**

```
INP1:FILT:HPAS  ON        Input 1 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 2	none

Example **Querying input 1 high pass filter state**

```
dimension  statement      String for data
INP1:FILT:HPAS  ON        Input 1 high pass filter to on
INP1:FILT:HPAS?           Query instrument to return input 1
                          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 2	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 2	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 2	None
value	numeric	50|1E6	OHM

Example

Set input 1 impedance to 50Ω

```
INP1:IMP  50
```
Input 1 impedance to 50Ω

Comments

- ***RST Condition:** Defaults to 1MΩ on all inputs.
- **Related Commands:** [SENSe:]INPut<n>:COUPling.

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 2	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 2	none
mode	boolean	ON|OFF|1|0	none

Example

Enabling input 2

```
INP2:STAT 1
```
Enable input 2

Comments

- **Related Commands:** [SENSe:]INPut<n>[:STATe]?
- ***RST Condition:** Defaults to input 1 on, input 2 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 2	none

Example

Query input 2 state

`INP2:STAT 1`	*Enable input 2*
`INP2?`	*Query instrument to return input 2 state*
`enter statement`	*Enter value into computer*

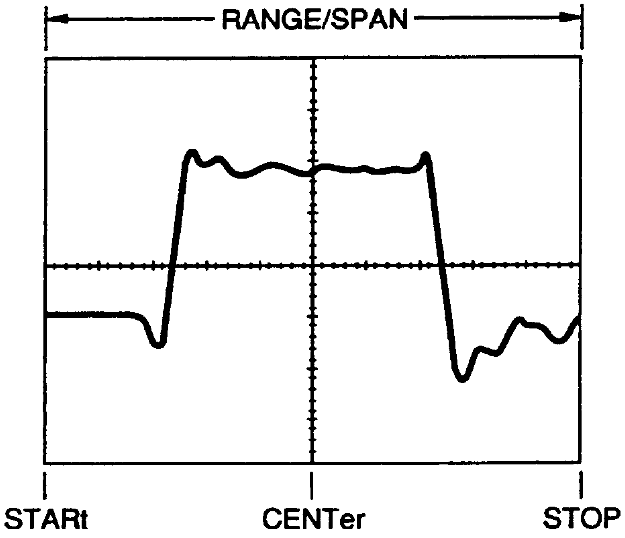
SWEep

[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 record. When operating in the repetitive mode, the only acceptable value is 500. When operating in the real-time mode, acceptable values are 500 or 8000. DETector:MODE sets mode.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
points	numeric	500|8000	none

Example

Set acquisition points to 500

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  500              Acquisition points are 500
```

Comments

- **Entering Points:** Any value between 0 and 1023 can be entered, however entry will be adjusted to 500. Any value entered that is >1024 will be adjusted to 8000. Values entered >1024 in repetitive mode will be adjusted to 500.
- **Waveform Points:** To determine the ACTUAL number of time buckets acquired, send the TRACe:POINts? query.
- **Related Commands:** DETector:MODE, 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  500              Acquisition points are 500
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 for repetitive acquisition. 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	10NS 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	10NS 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 <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 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 and 2 are independently programmable.

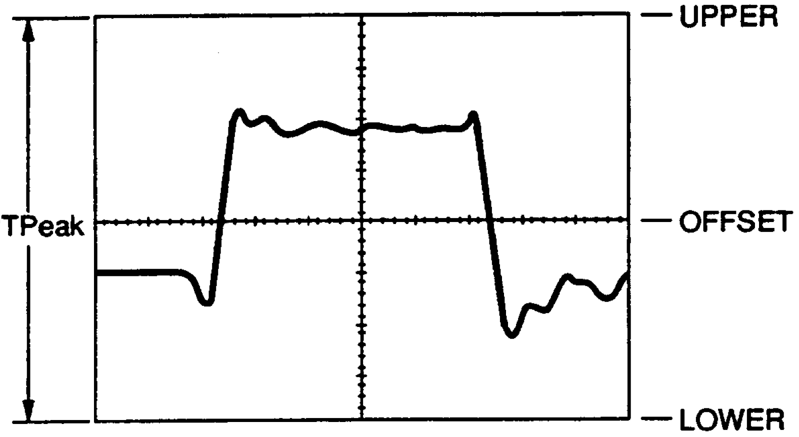
Parameters

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

Comments

- **Entering Number:** The desired input *number* from 1 to 2 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 2 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 2	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 2	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 2	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 1 upper range limit to 10 volts

```
VOLT1: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 1 current upper value**

`VOLT1:RANG:UPP 10`	*Upper limit to 10 volts*
`VOLT1: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.

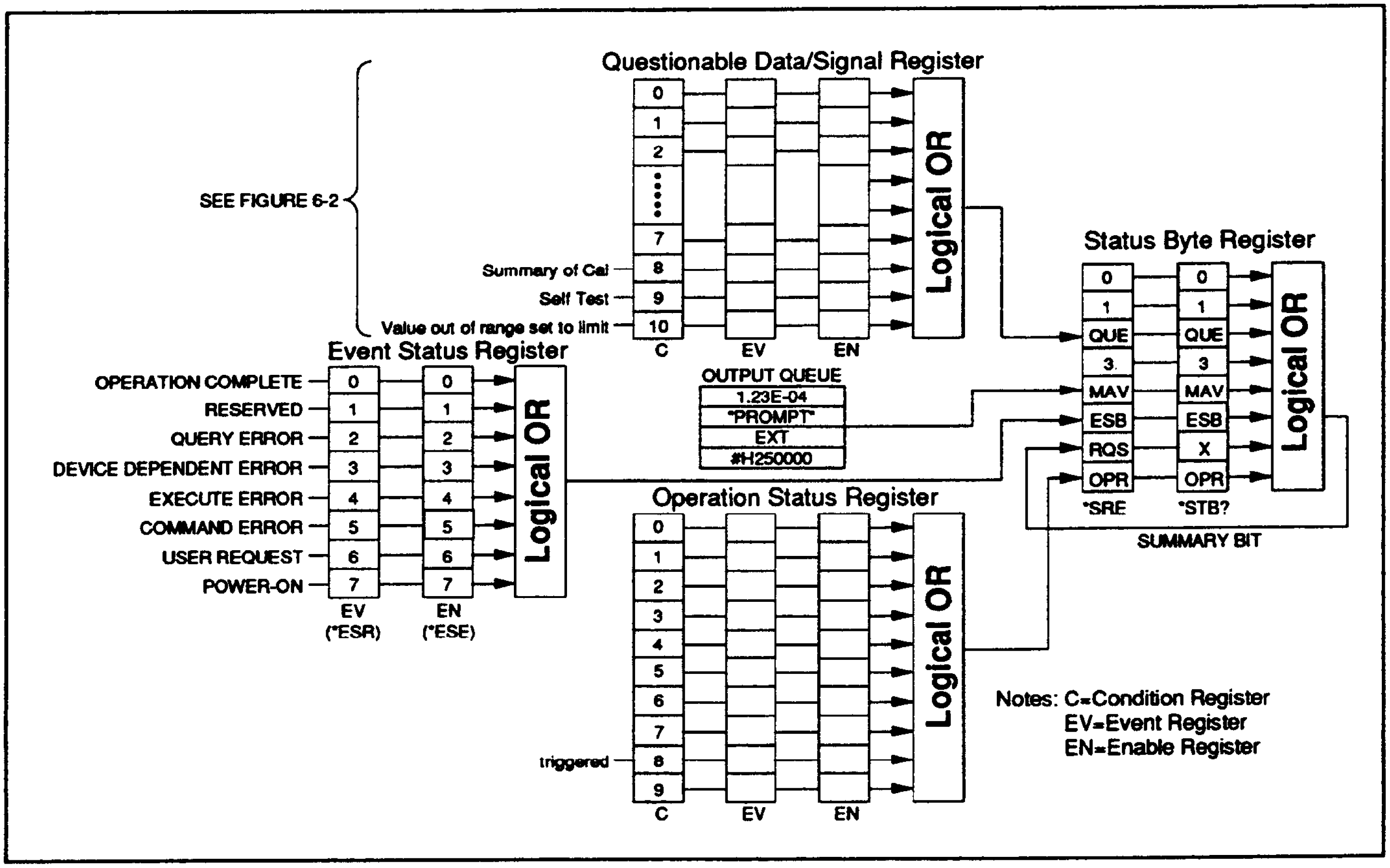


Figure 6-1. Oscilloscope STATus Registers

Subsystem Syntax

```
STATus
    :OPERation
            :CONDition?
            :ENABle
            [:EVENt]?
    :PRESet
    :QUEStionable
        :CONDition?
        :ENABle
        [:EVENt]?
        :CALibration
            :CONDition?
            :ENABle
            [:EVENt]?
            :INPut<number> | EXTernal
                :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]?
```

Subsystem Syntax

```
STATus — Continued
    :QUEStionable — Continued
        :CALibration — Continued
            :DCALibration
                :CONDition?
                :ENABle
                [:EVENt]?
            :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]?
```

Subsystem Syntax

```
STATus — Continued
    :QUEStionable — Continued
        :TEST — Continued
            :RAM — Continued
                :NVOLatile
                    :CONDition?
                    :ENABle
                    [:EVENt]?
                :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.

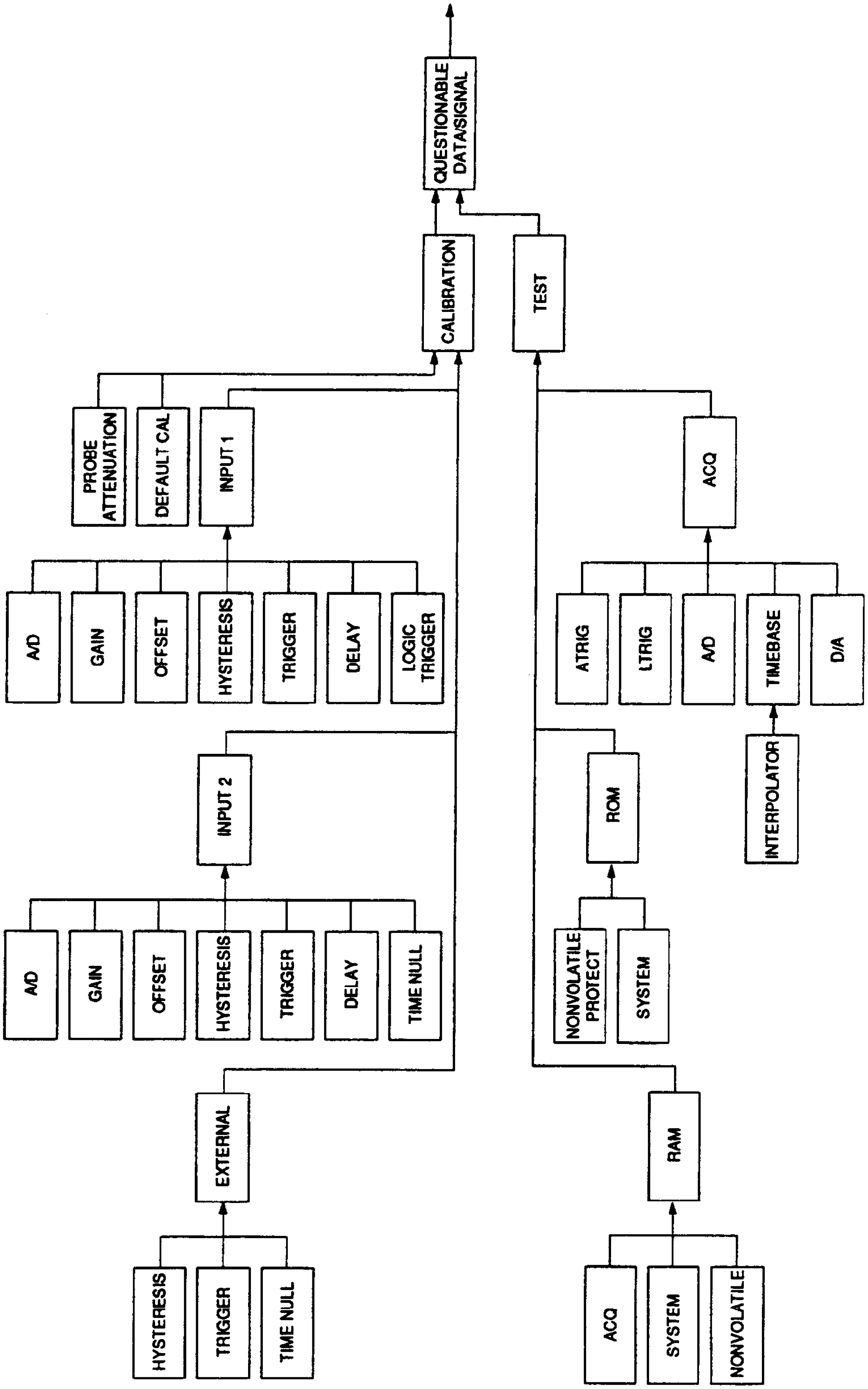
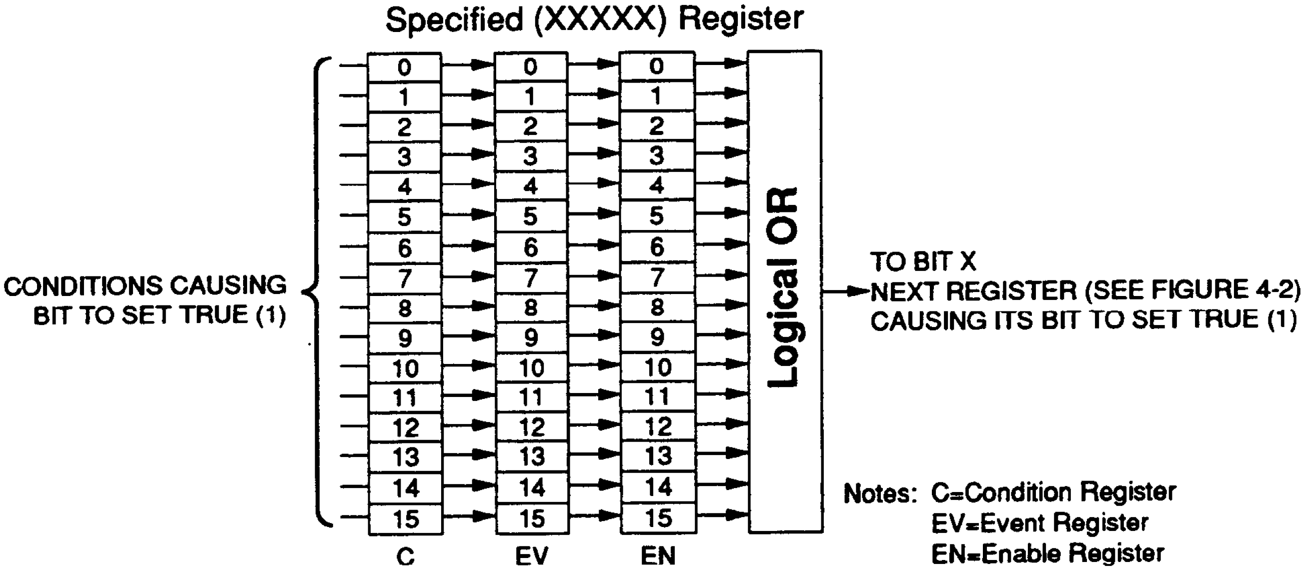


Figure 6-2. Oscilloscope Status Questionable Data/Signal Register Subsystem

: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 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 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.

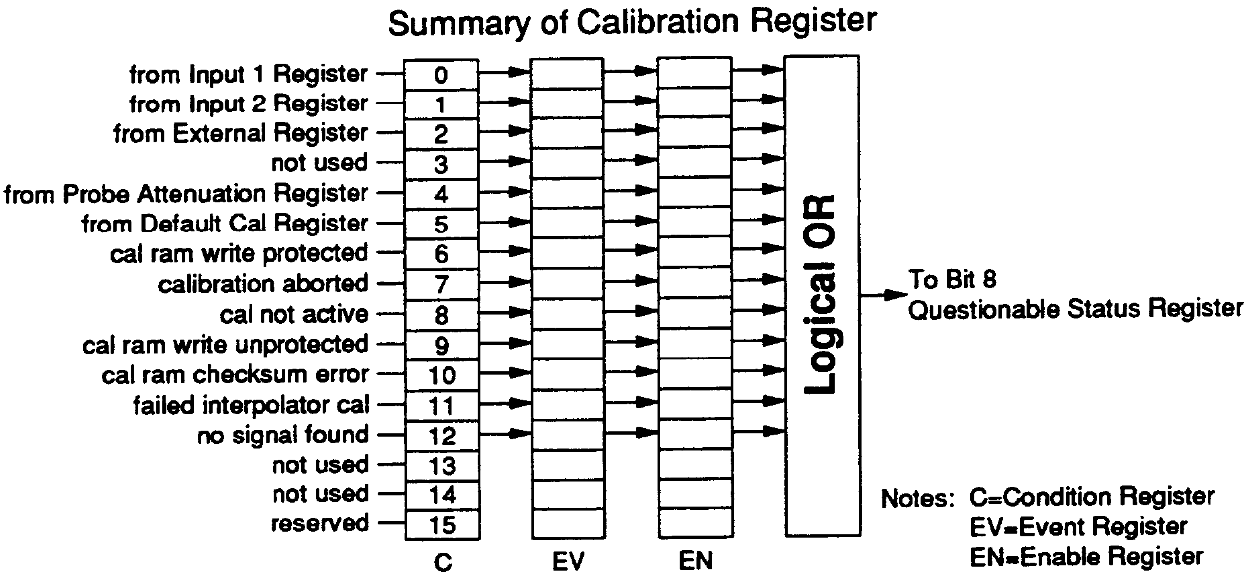
Example **Read the event register**

```
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?
```

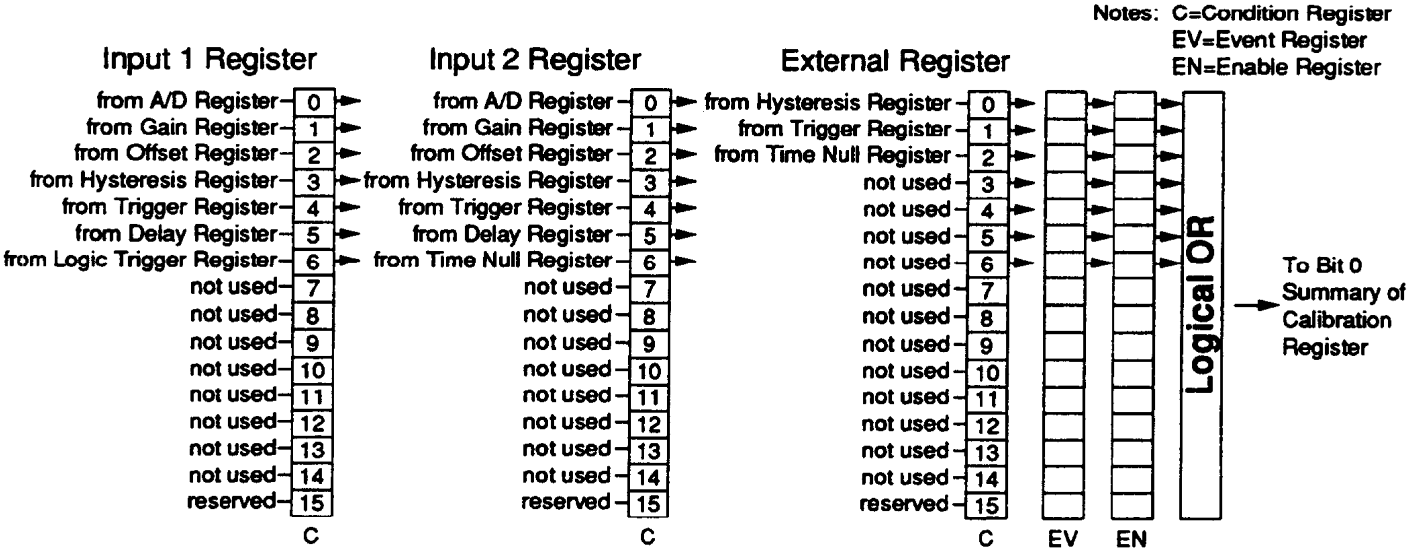
Query instrument to return register contents

:QUEStionable:CALibration:INPut|EXTernal

STATus:QUEStionable:CALibration:INPut<*number*> | EXTernal register reports the status of calibration data for the input specified. *number* (1 to 2) or EXTernal 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 2	none



Example **Query input 1 event register**

`STAT:QUES:CAL:INP1?`	*Query instrument to return register contents*

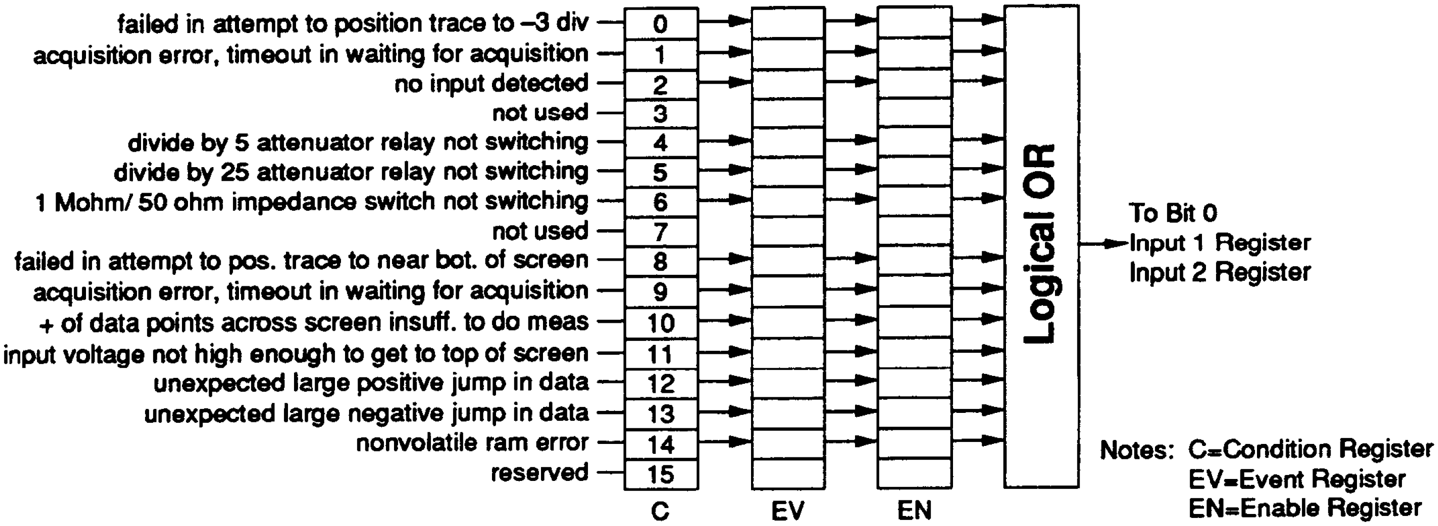
:QUEStionable:CALibration:INPut:AD

STATus:QUEStionable:CALibration:INPut<*number*>:AD register reports the status of the A/D calibration data for the input specified. *number* (1 to 2) 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 2	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
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query input 2 A/D event register**

```
STAT:QUES:CAL:INP2:AD?
```

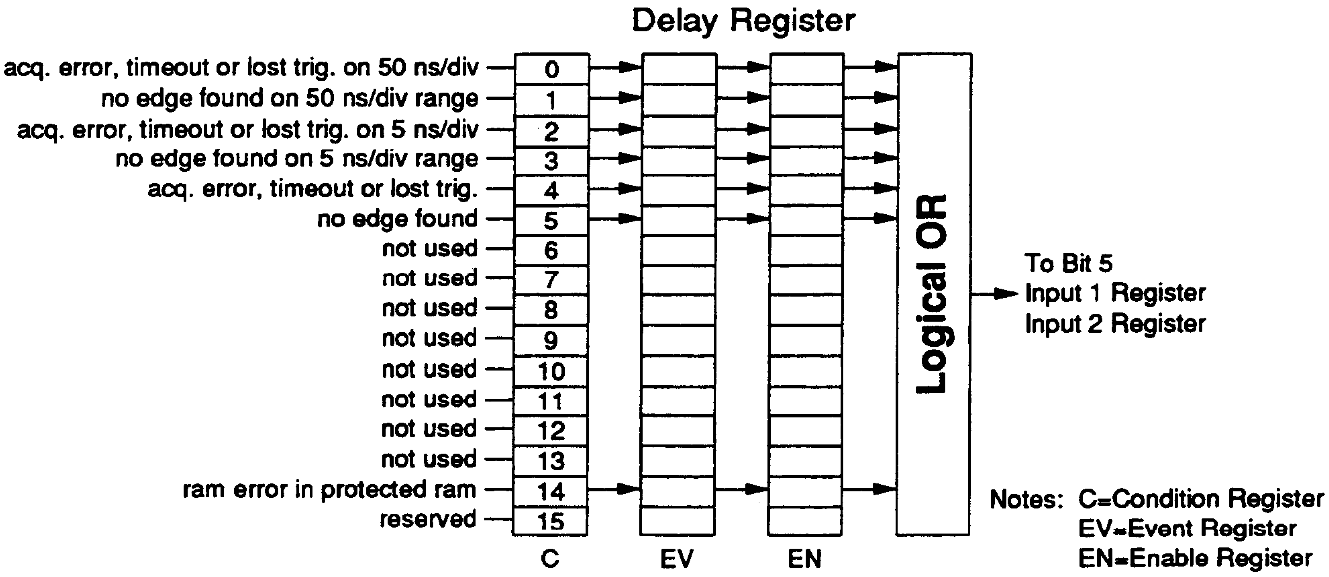
Query instrument to return register contents

: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 2) 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 2	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.
no edge found
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
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example **Query input 1 delay event register**

```
STAT:QUES:CAL:INP1:DEL?
```

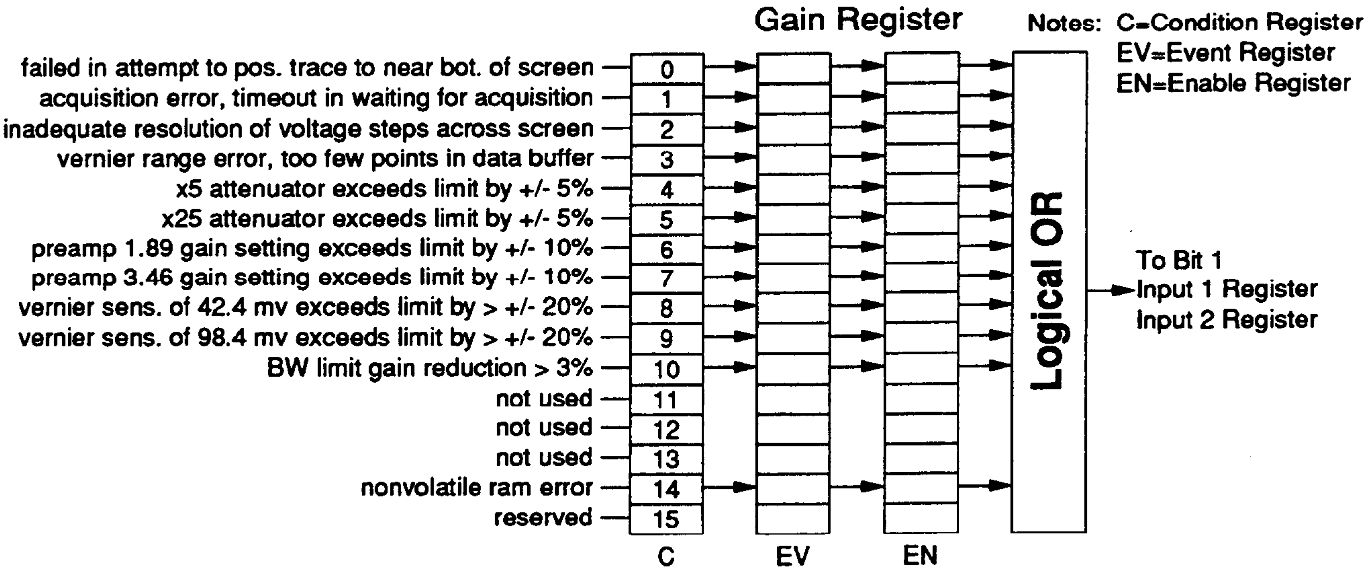
Query instrument to return register contents

: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 2) 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 2	none


Gain Register
Notes: C=Condition Register
EV=Event Register
EN=Enable 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%
BW 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
To Bit 1
Input 1 Register
Input 2 Register

Example **Query input 2 gain event register**

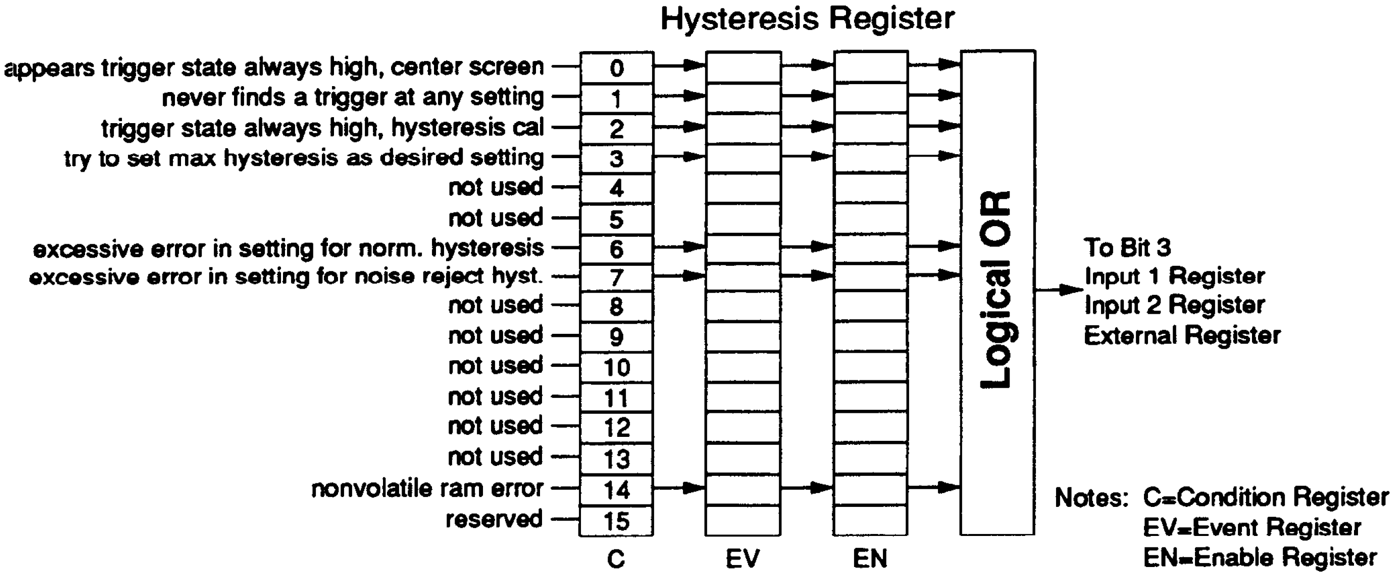
`STAT:QUES:CAL:INP2:GAIN?` *Query instrument to return register contents*

:QUEStionable:CALibration:INPut|EXTernal:HYSTeresis

STATus:QUEStionable:CALibration:INPut<*number*> | EXTernal:HYSTeresis register reports the status of hysteresis calibration data for the input specified. *number* (1 to 2) or EXTernal 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 2	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
External Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

Example

Query input 1 hysteresis event register

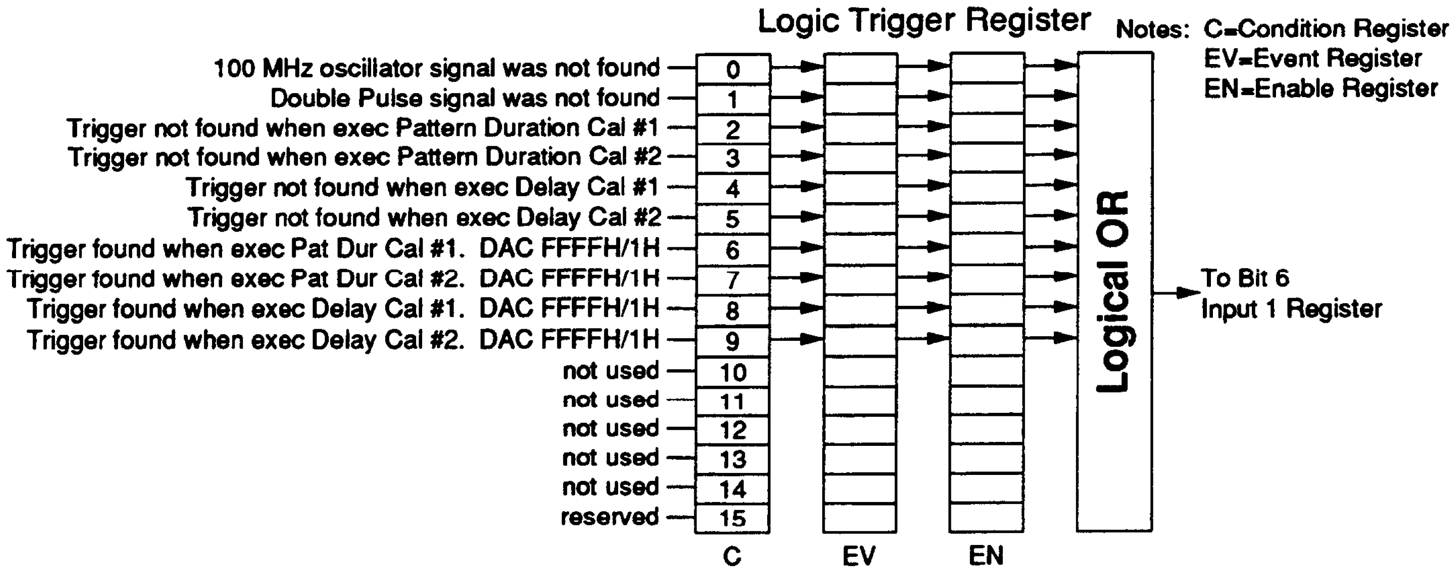
`STAT:QUES:CAL:INP1:HYST?`	*Query instrument to return register contents*

: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
Logical OR
To Bit 6
Input 1 Register
C
EV
EN

Example **Query input 1 logic trigger event register**

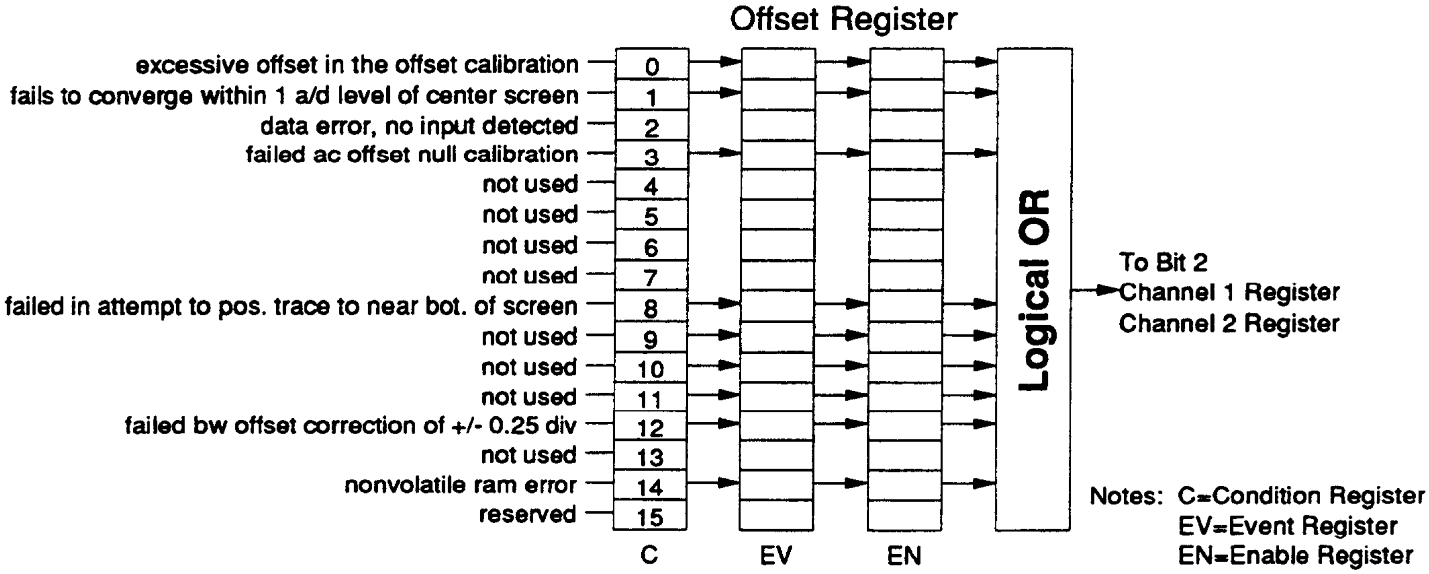
`STAT:QUES:CAL:INP1:LTR?` *Query instrument to return register contents*

: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 2) 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 2	none

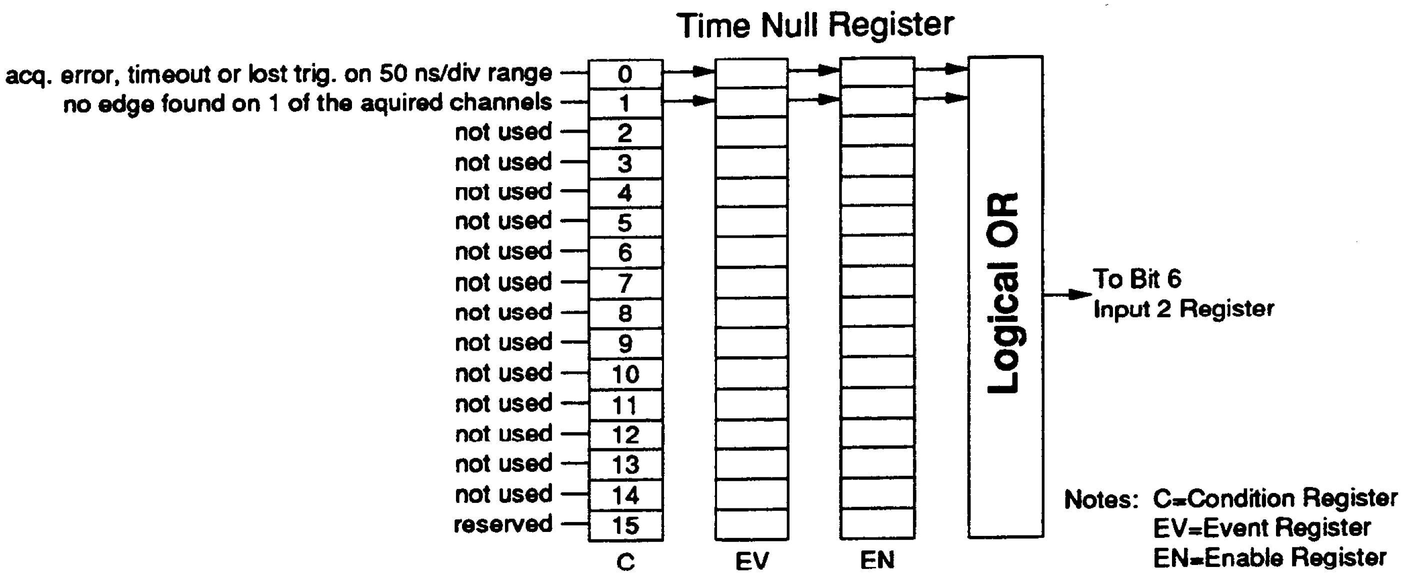

Offset Register
excessive offset in the offset calibration
fails to converge within 1 a/d level of center screen
data error, no input detected
failed ac offset null calibration
not used
not used
not used
not used
failed in attempt to pos. trace to near bot. of screen
not used
not used
not used
failed bw offset correction of +/- 0.25 div
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 2
Channel 1 Register
Channel 2 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:INPut2|EXTernal:TNULI

STATus:QUEStionable:CALibration:INPut2 | EXTernal:TNUL1 register reports the status of time null calibration data for the input specified. Input2 or EXTernal 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.



Example **Query input 1 time null event register**

`STAT:QUES:CAL:INP1:TNUL?` *Query instrument to return register contents*

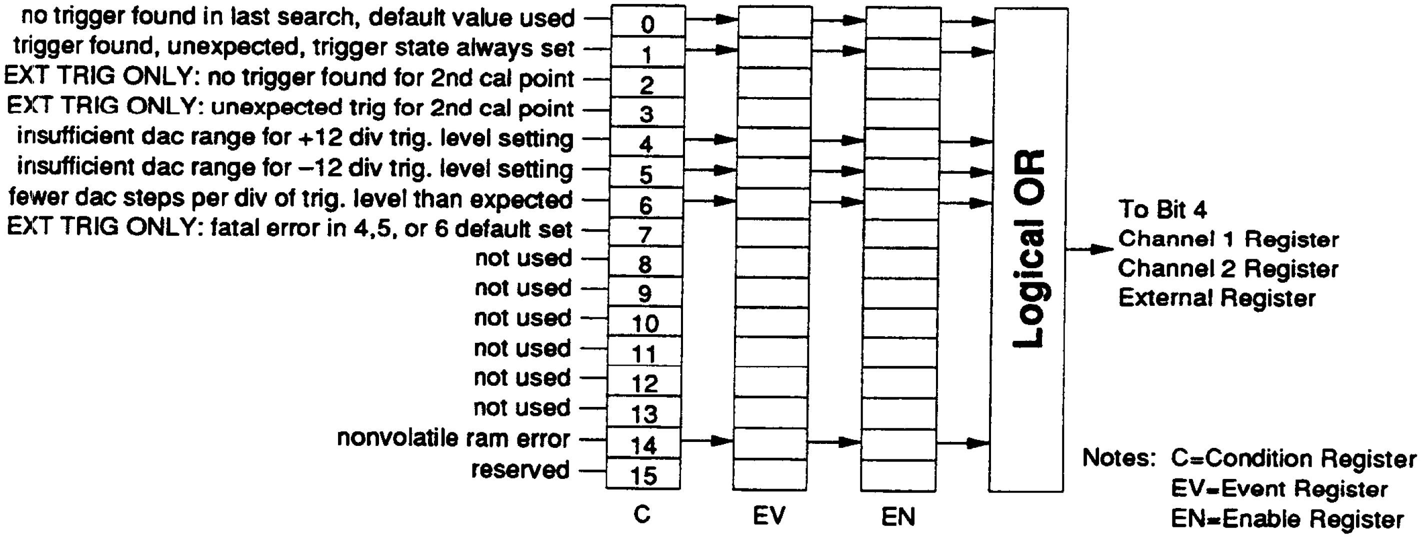
:QUEStionable:CALibration:INPut|EXTernal:TRIGger

STATus:QUEStionable:CALibration:INPut<*number*>|EXTernal:TRIGger register reports the status of trigger calibration data for the input specified. ***number*** (1 to 2) or EXTernal 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 2	none

Trigger Register


no trigger found in last search, default value used
trigger found, unexpected, trigger state always set
EXT TRIG ONLY: no trigger found for 2nd cal point
EXT TRIG ONLY: unexpected trig for 2nd cal point
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
EXT TRIG ONLY: fatal error in 4,5, or 6 default set
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
External Register
Notes: C=Condition Register
EV=Event Register
EN=Enable Register

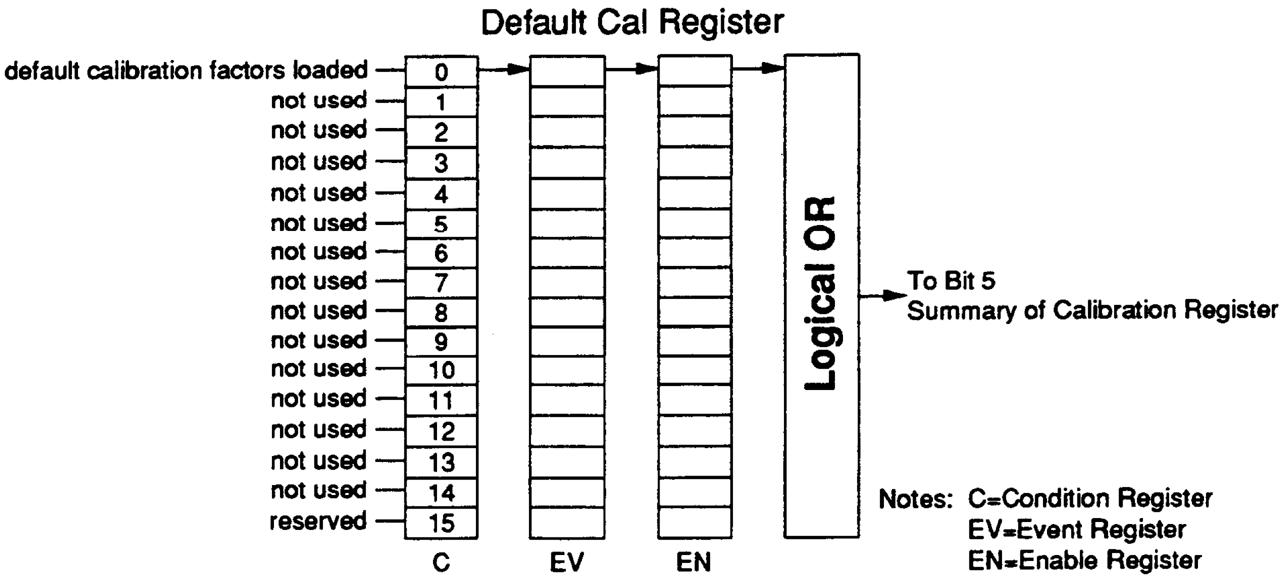
Example **Query input 2 trigger event register**

```
STAT:QUES:CAL:INP2: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.



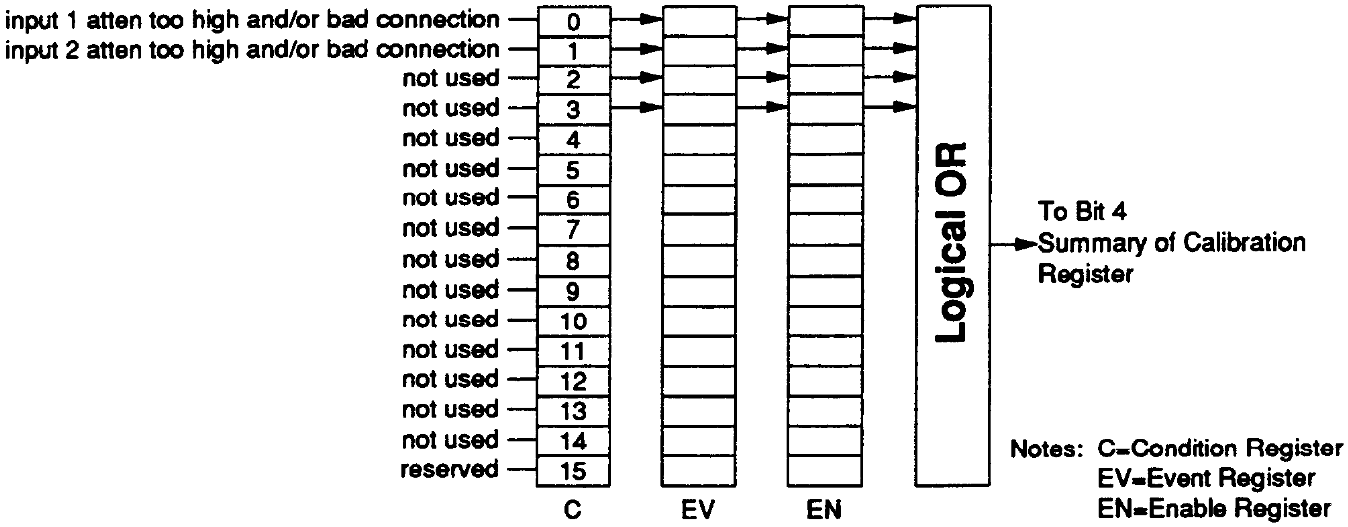
Example **Query default calibration event register**

`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



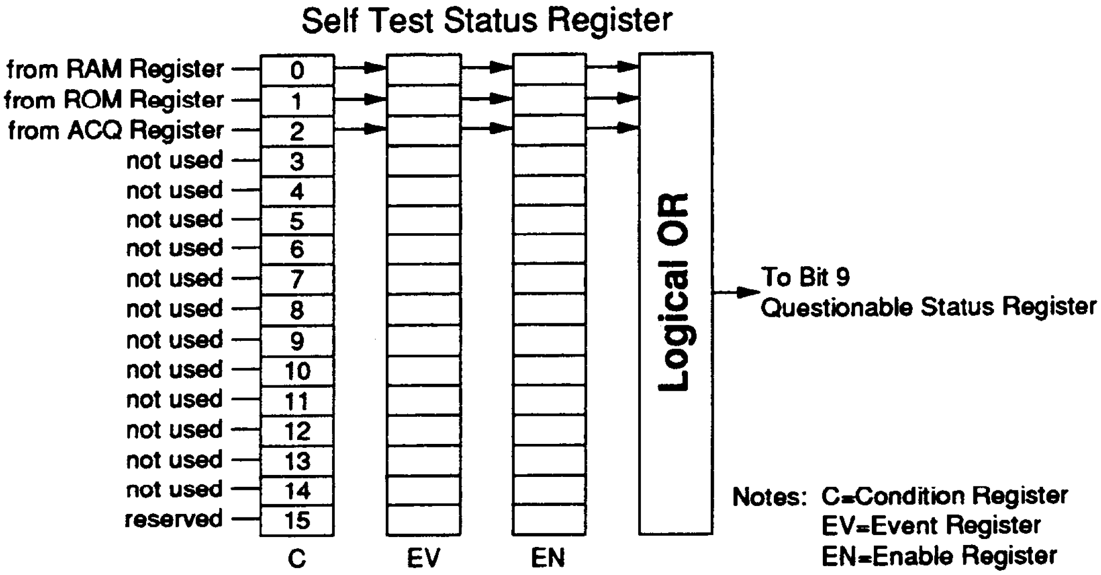
Example **Query probe calibration attenuation event register**

```
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.

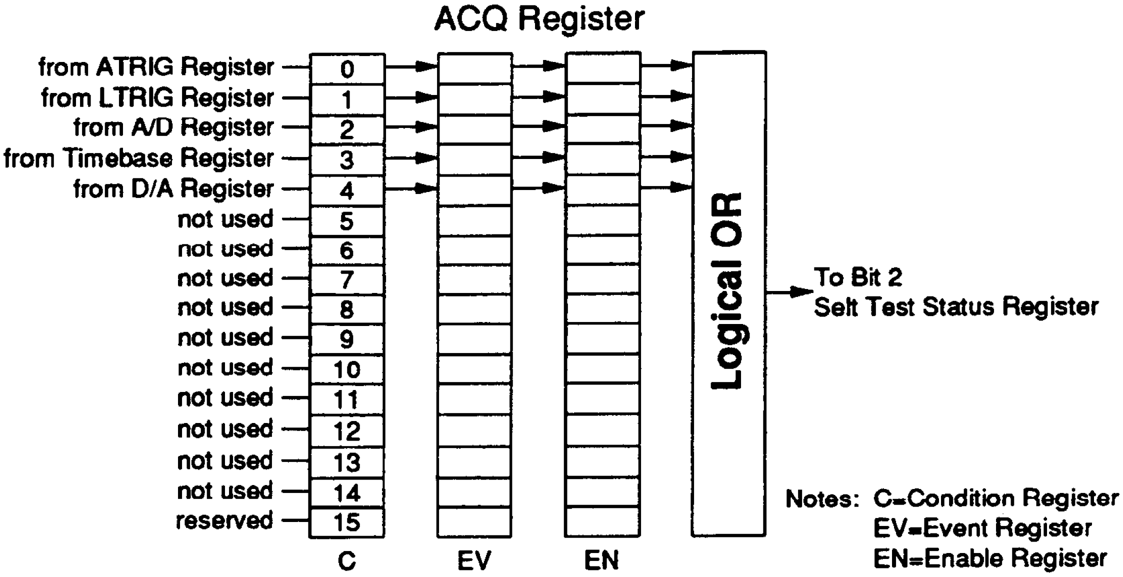


Example **Query test event register**

`STAT:QUES:TEST?`	*Query instrument to return register contents*

:QUEStionable:TEST:ACQuisition

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.

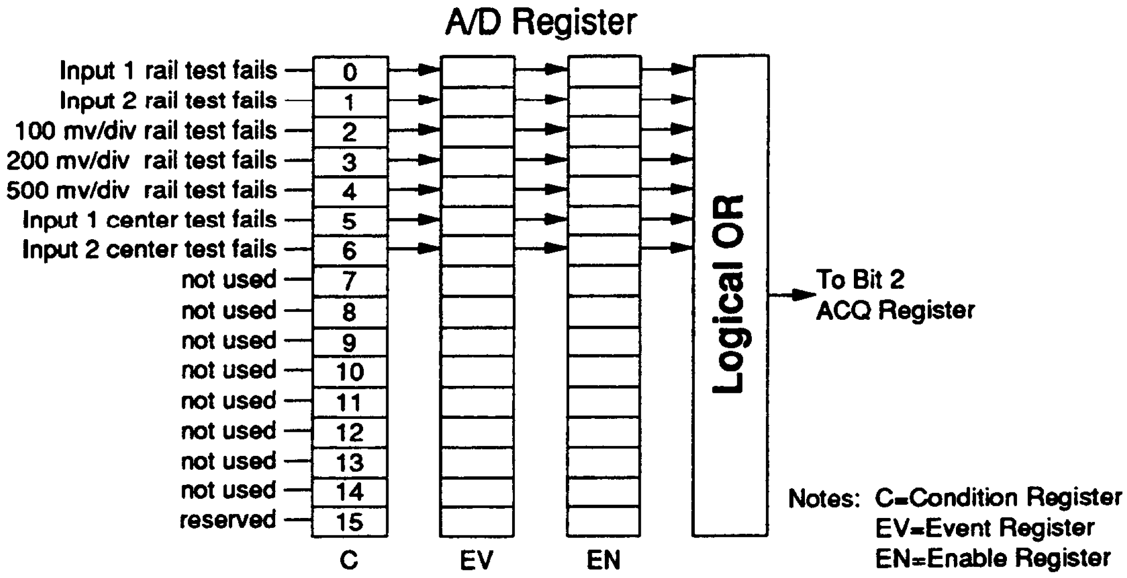


Example **Query acquisition event register**

`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.

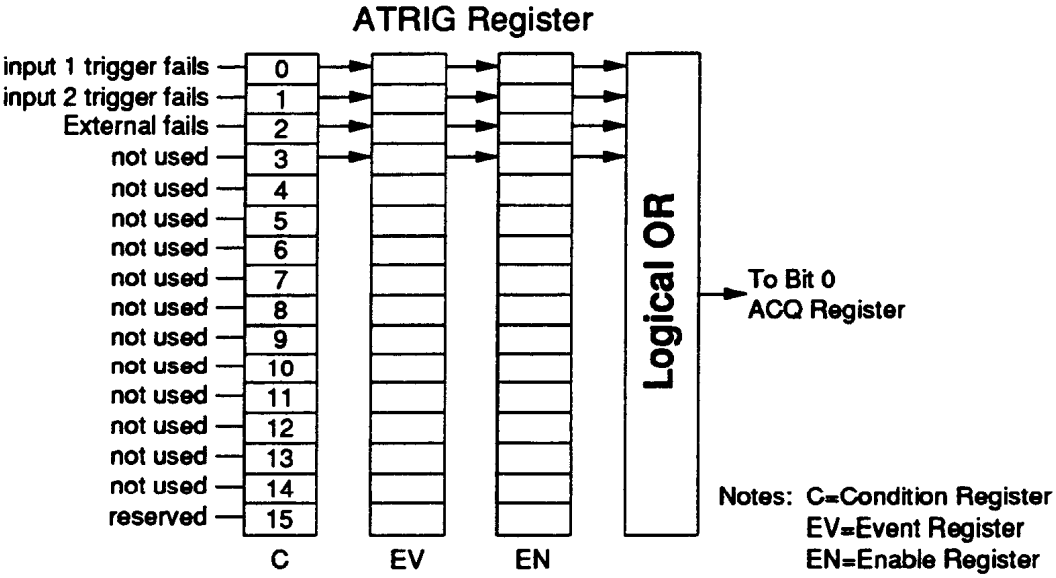


Example **Query acquisition A/D event register**

`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.

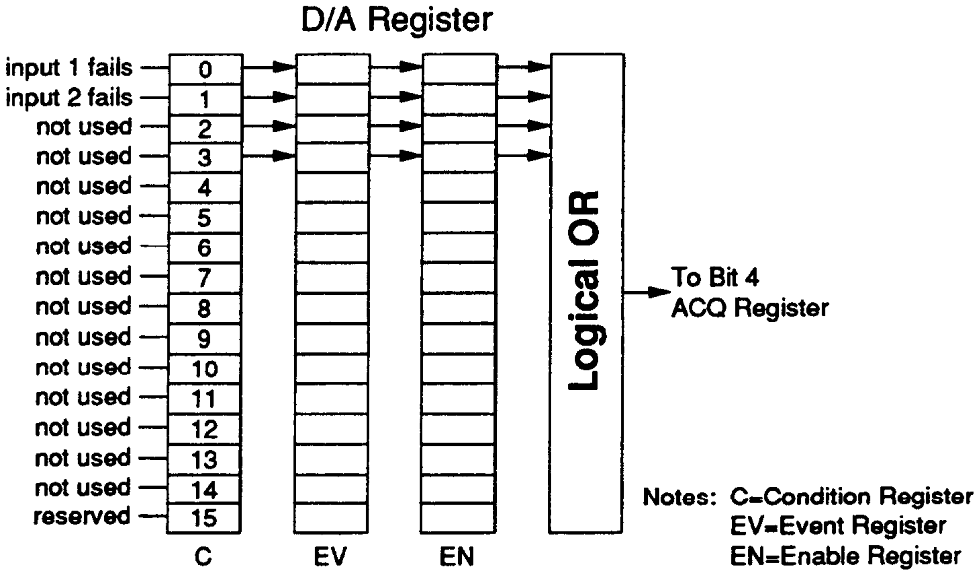


Example **Query acquisition analog trigger event register**

`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.



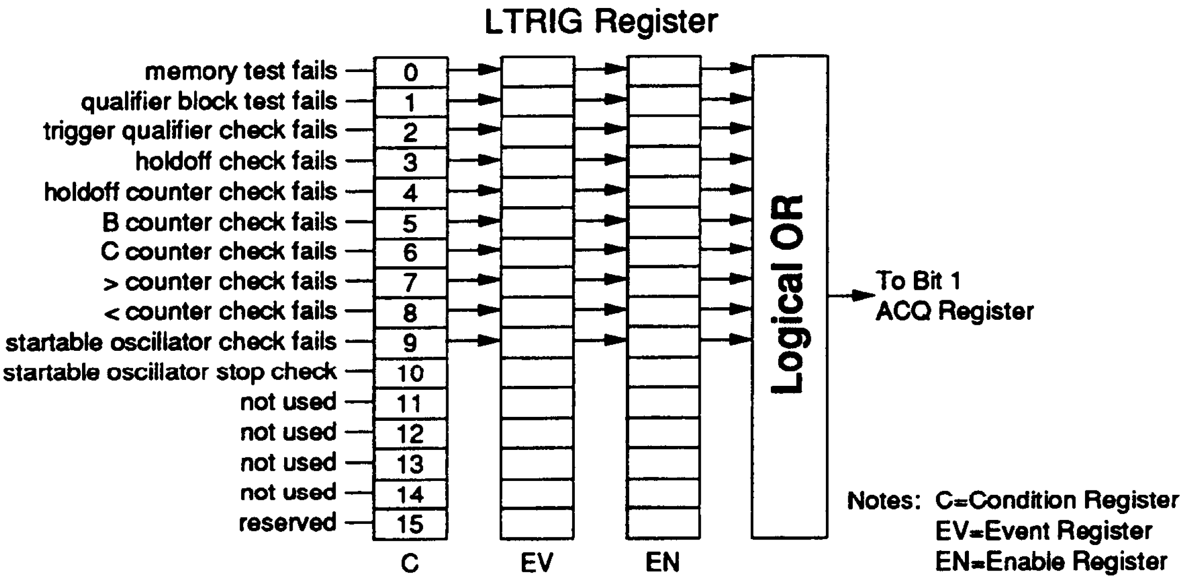
Example **Query acquisition D/A event register**

```
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.

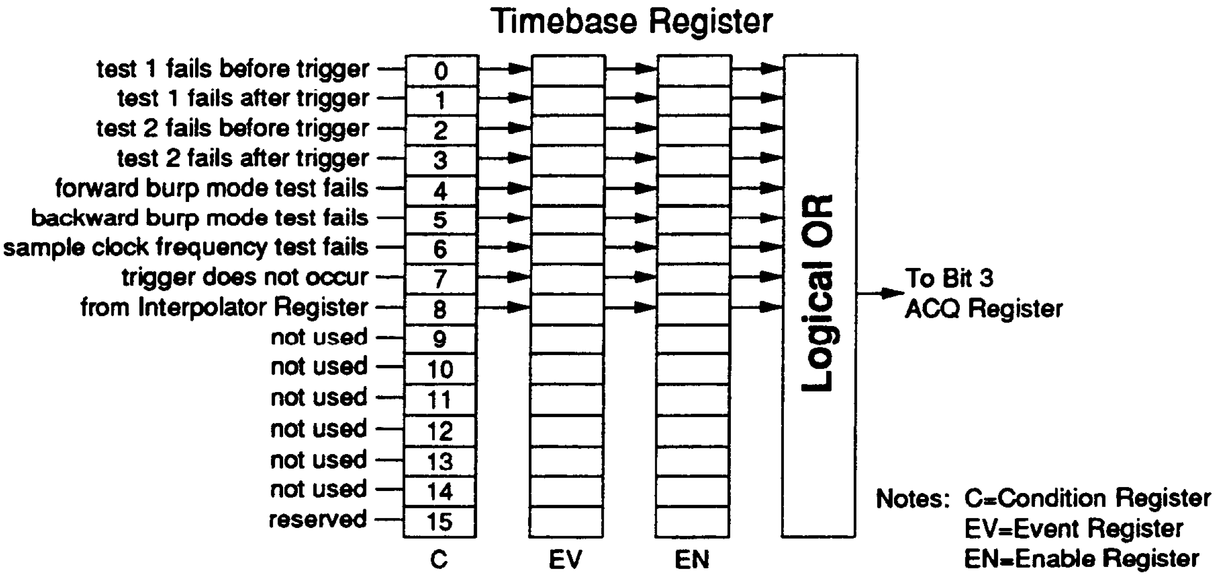


Example **Query acquisition logic trigger event register**

`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.

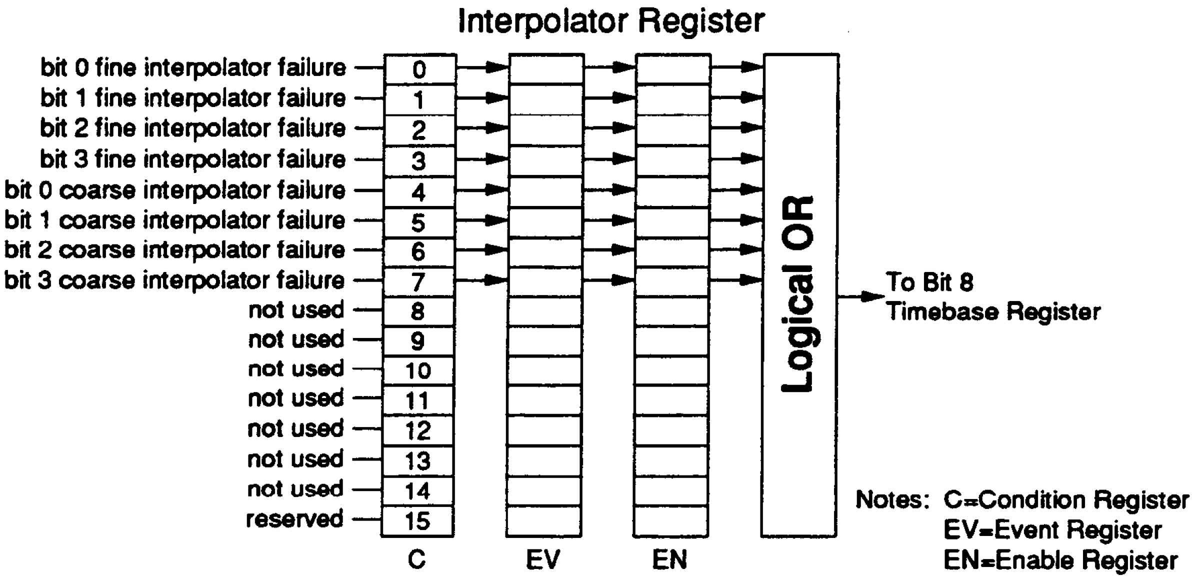

Timebase Register
test 1 fails before trigger
test 1 fails after trigger
test 2 fails before trigger
test 2 fails after trigger
forward burp mode test fails
backward burp mode test fails
sample clock frequency test fails
trigger does not occur
from Interpolator Register
not used
not used
not used
not 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.

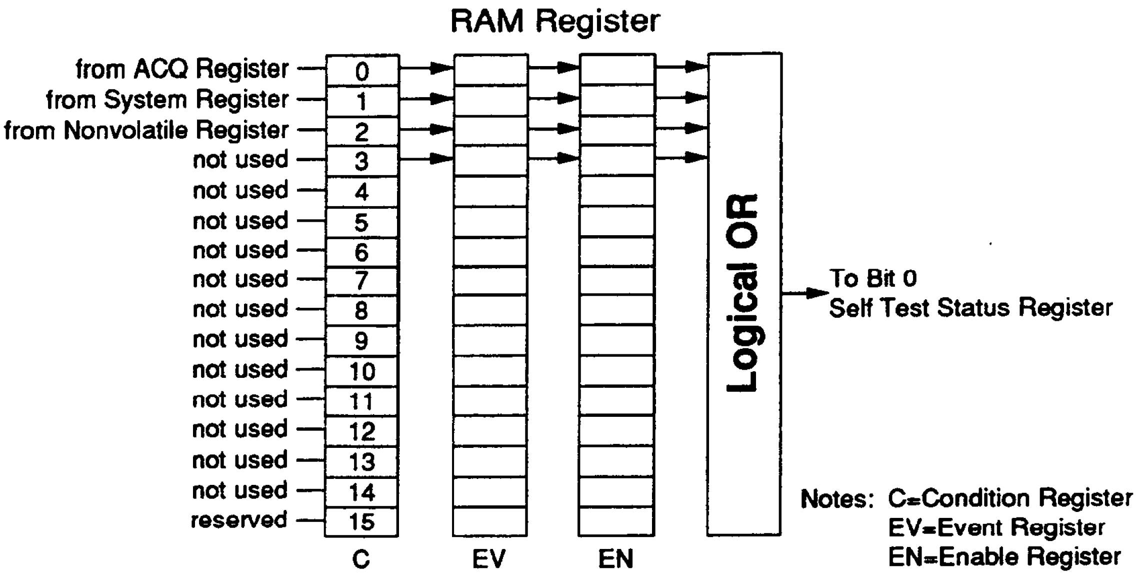

Interpolator Register
bit 0 fine interpolator failure
bit 1 fine interpolator failure
bit 2 fine interpolator failure
bit 3 fine interpolator failure
bit 0 coarse interpolator failure
bit 1 coarse interpolator failure
bit 2 coarse interpolator failure
bit 3 coarse interpolator failure
not used
not used
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.

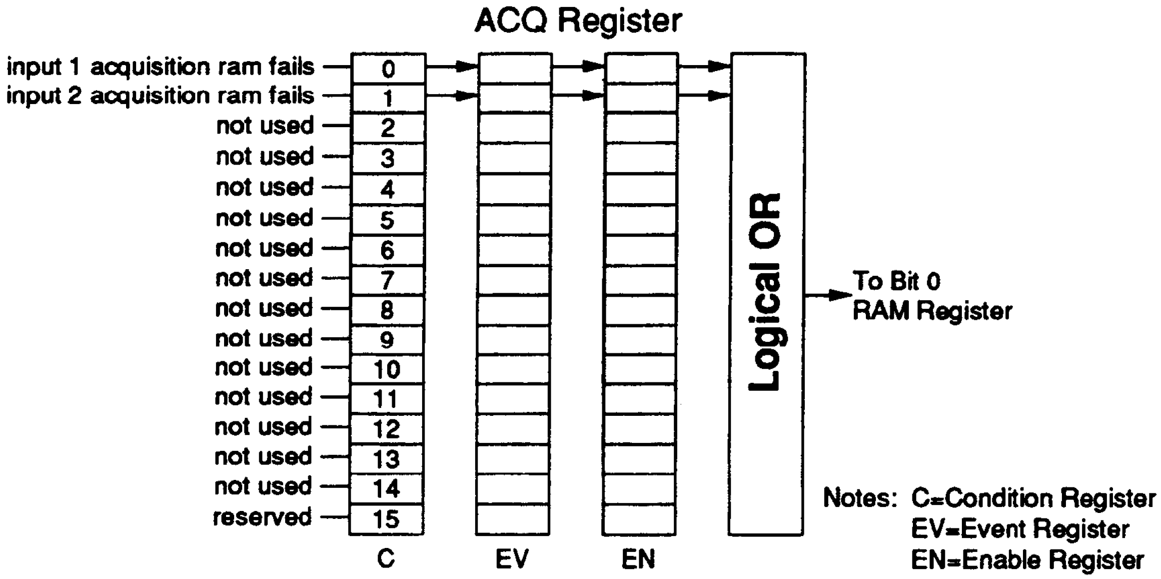


Example **Query RAM event register**

`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.

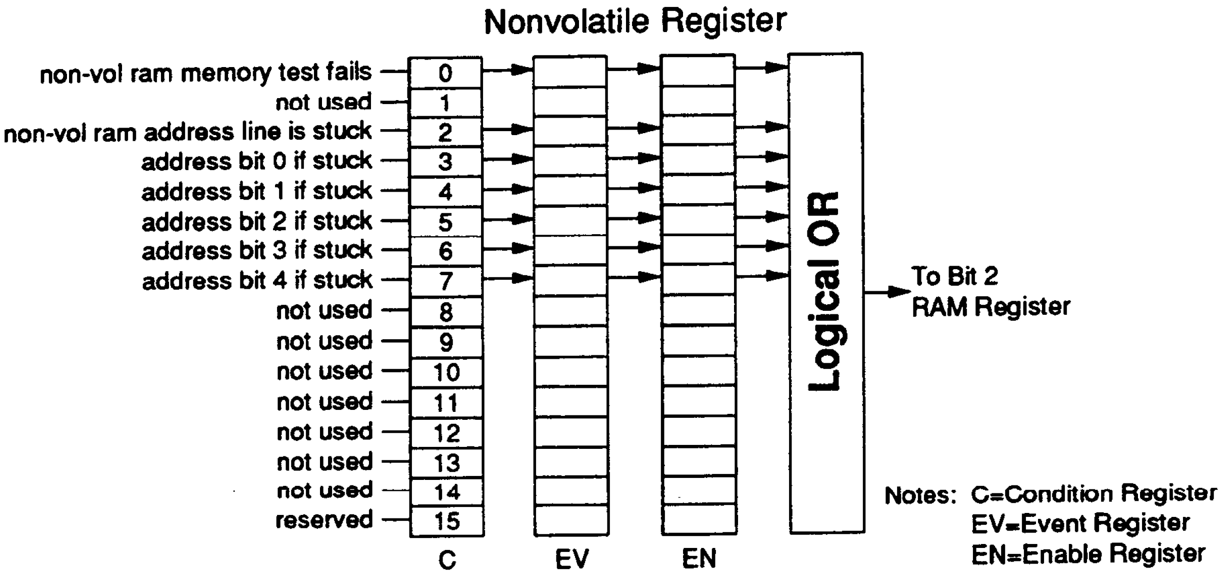


Example **Query acquisition RAM event register**

`STAT:QUES:TEST:RAM:ACQ?` *Query instrument to return register contents*

:QUEStionable:TEST:RAM: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.

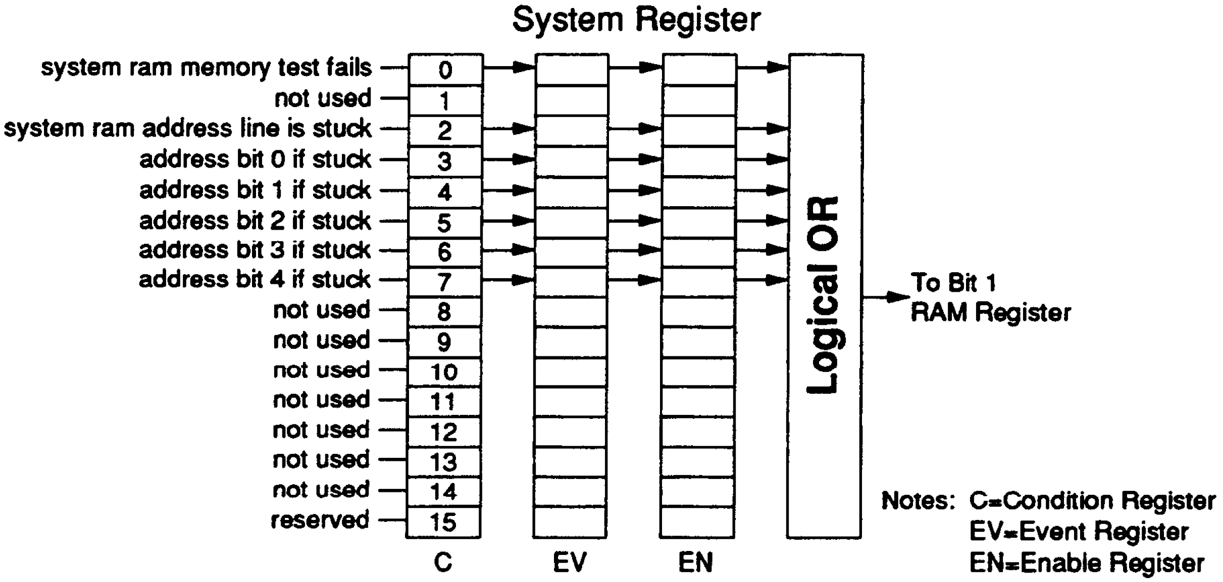


Example **Query nonvolatile RAM event register**

`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.

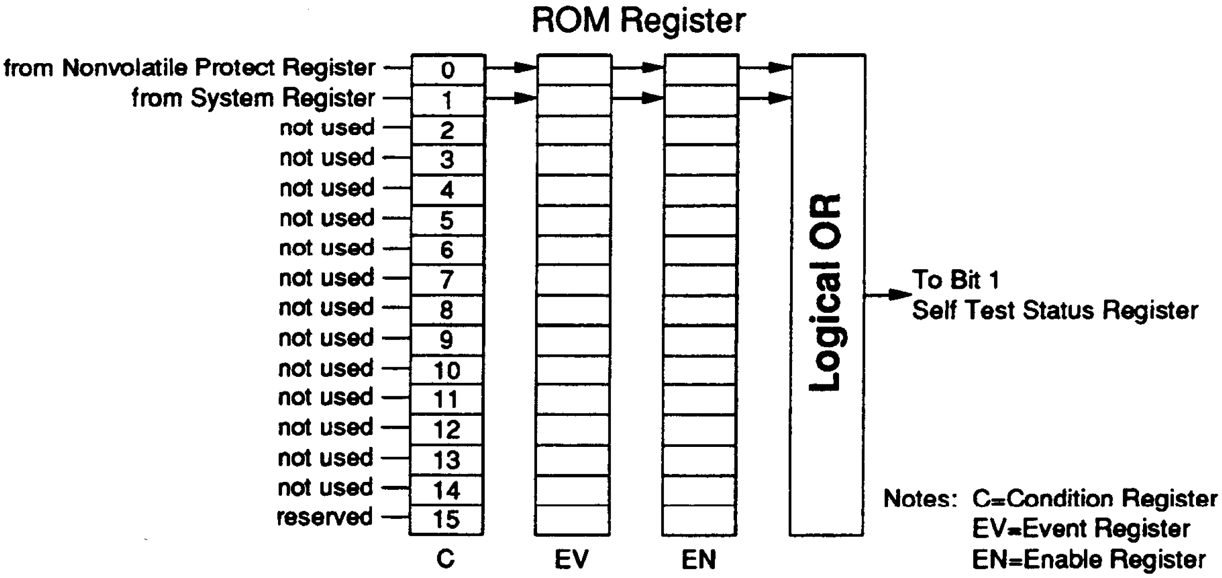


Example **Query system RAM event register**

`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.

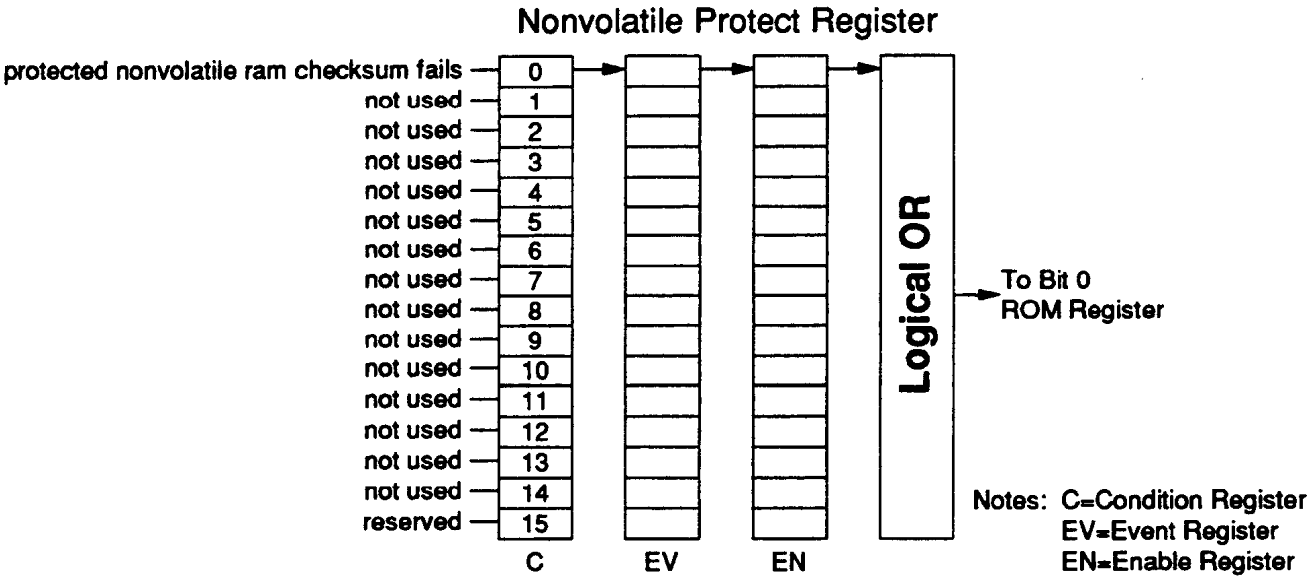


Example **Query ROM event register**

`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.

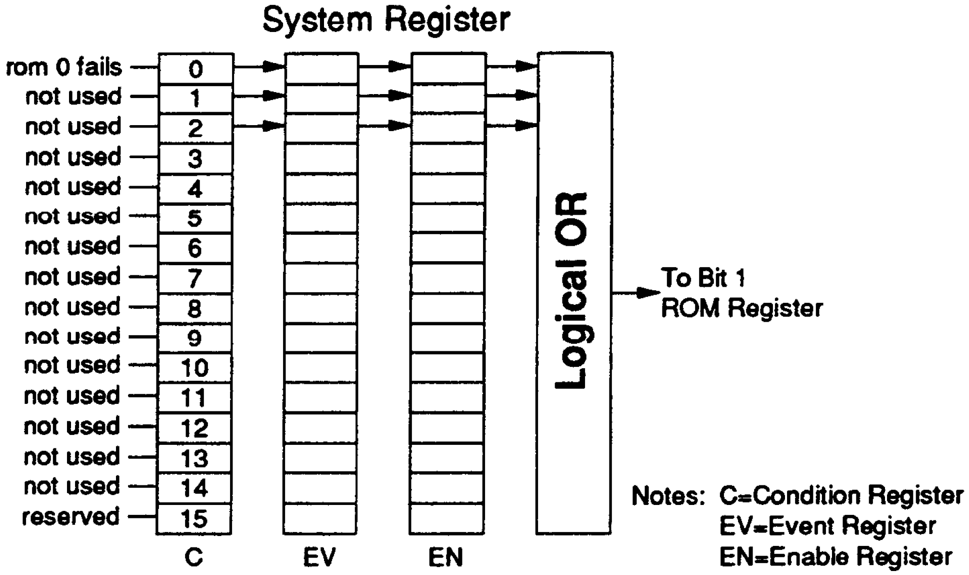


Example **Query non-volatile protected ROM event register**

`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, if calibration data is protected, 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?
    :NVPRotect
        :PASSword <old>, <new>
        [:STATe] <mode>, <pass>
        [:STATe?]
    :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 1, 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 54510A 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 54510A 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 54510A 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
```

:NVPRotect:PASSword

SYSTem:NVPRotect:PASSword ***<old>,<new>*** is used to set a new password.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
old	discrete	Six Characters 0-9, a-z, A-Z	none
new	discrete	Six Characters 0-9, a-z, A-Z	none

Example

Change password from "SYSTEM" to "System"

```
SYST:NVPR:PASS   SYSTEM,System
```

Password changed

Comments

- **Password:** The password can be any string of from one to six case-sensitive alpha-numeric characters. Spaces are allowed, but special characters are not. The old password has to be specified to change to a new password.
- **Factory Default:** The oscilloscope is shipped with resistor R208 (CAL PROTECT) installed. This is the non-protected mode and defaults the password to "SYSTEM".
- **Changing the default password:** Remove resistor R208 (CAL PROTECT), then execute the :NVPRotect:PASSword command. With resistor R208 removed, the last entered password and/or protect state are maintained after a power down.
- **Unknown Password:** If resistor R208 has been removed and you do not know the password, place a 0 ohm resistor in R208 (CAL PROTECT) space on the printed circuit board. Now powering up the oscilloscope will set the factory default password to "SYSTEM" and the protect state to non-protected. Change the default password using the procedure in the above paragraph.
- **Related Commands:** SYSTem:NVPRotect[:STATe].
- **Factory Default:** Defaults to "SYSTEM".

:NVPRotect[:STATe]

SYSTem:NVPRotect[:STATe] ***<mode>,<password>*** enables (ON | 1) and/or disables (OFF | 0) the non volatile RAM protect mode. The password has to be specified for both cases.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
mode *password*	boolean discrete	ON|OFF|1|0 Six Characters 0-9, a-z, A-Z	none none

Example **Disabling Non-Volatile RAM protect**

```
SYST:NVPR  0,"SYSTEM"   Non-volatile RAM protect to off
                        (unprotected)
```

Comments

- **Calibration:** When performing any of the firmware calibrations, the protect mode has to be OFF (see SYSTem:NVPRotect).
- **Password:** If the password is unknown, placing a 0Ω resistor in R208 (CAL PROTECT) space on the printed wiring board, then powering up the module will set the factory default password to "SYSTEM" and protect state to OFF. With R208 removed, the last entered password and/or protect state are maintained after a power down.
- **Related Commands:** SYSTem:NVPRotect:PASSword.
- **Factory Default:** Defaults to OFF.

:NVPRotect[:STATe]?

SYSTem:NVPRotect[:STATe]? queries the present state of the non volatile RAM protect mode. The query returns ON if the protection mode is enabled (protected) or OFF if the protection mode is disabled (unprotected). The value is sent to the output buffer. See SYSTem:NVPRotect[:STATe] command for more information.

Example **Query Non-Volatile RAM protect state**

```
dimension  statement    String for data
SYST:NVPR  0,"SYSTEM"   Non-volatile RAM protect to off
                        (unprotected)
SYST:NVPR?              Query instrument to return Non-
                        volatile RAM protect enable state
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 (").
- **Non-volatile RAM Protect Mode:** The Non-volatile RAM Protect Mode must be set to OFF prior to writing a new serial number to the protected non-volatile ram within the instrument (see SYSTem:NVPRotect).
- **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 three random access memory tests. When selected, the Oscilloscope performs a System RAM test, a Non-volatile RAM test, and/or an Acquisition RAM test. If the *test* parameter is not sent, all three tests are performed.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Value
test	discrete	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 three 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 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 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:

- 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	WMEMory*n*(*n*=1 to 4) See comments before selecting	none
source	discrete	0|<*data*>|INPut*n*(*n*=1 to 2)| 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 waveform memory 1

For the example, waveform data is "#41000...."

```
TRAC  WMEM1,#41000....
```

Send waveform data to Oscilloscope and place in WMEM1

Comments

- **Selecting Source and Destination:** Depending on the desired action, make selections of source and destination as follows:

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, INPut1-2, MATH1-2).

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).

- **RAWData Mode:** When in raw data acquisition mode, only entries of 500 or 8000 point records are accepted. If variable length record entries (RAWDATA acquisition type) are sent, an error message will be generated.
- **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, 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 2)| 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?  INP1             Enter value from input 1 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), MATH functions (MATH1-2), or input buffers (INP1-2) may have waveform data sent from them. Select the desired location using the *source* parameter.
- **Waveform Data Format:** The format of the waveform data being sent is specified using the FORMat[:DATA] command.

- **Selecting Type:** The type of waveform acquisition is selected by the [SENSe:]AVERage:TYPE command.

  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.

  Rawdata: Rawdata can only be acquired with a digitize operation, and cannot be stored in memories or measured. Data is stored as uncalibrated 8-bit Gray code data in a large buffer. When all acquisitions are complete, the buffer is translated into unfiltered, calibrated 1 bit binary data and sent over the bus in the 16 bit format. The FORMat[:DATA] command has no effect in Rawdata mode.

The command has two parameters: length and acquisitions. Length specifies the number of points of each acquisition. Acquisitions specify the number of acquisitions to be taken in a single digitize operation. The data is transferred from the oscilloscope in a single IEEE 488.2 data block consisting of two arrays.

- The first array consists of double precision 64 bit floating point numbers. This array contains the x-origin values of the waveform records that follow. It can read directly into a double precision real array in a controller allocated by the BASIC command ALLOCATE REAL Xorigins (l:Acquisitions).
- The second array consists of 16-bit integer numbers. This data is transmitted in a linear fashion, and starts with sample zero of the first acquisition, and continues through sample length-l of the acquisition. Then it continues in a similar fashion with sample zero through sample length-l of the each following acquisition through the last acquisition. This array can be read directly into a two dimensional integer array allocated by the BASIC command ALLOCATE INTEGER Waveforms (l:Acquisitions, l:Points).

- **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 (500 or 8000) 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 2) 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 all cases the number of time buckets actually acquired will be the number of points set with the [SENSe:]AVERage:POINts command (500, 8000, or 4 to 8000, dependent on the DETector:MODE mode selected).
- **Related Commands:** [SENSe:]SWEep:TIME, [SENSe:]AVERage:POINts, DETector:MODE.

: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	discrete	WMEMory*n*(*n*=1 to 4)	none
data	numeric	See comments below	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 *destination*.
- **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 4 for RAWData 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.

- **RAW Data Mode:** In Raw Data mode, the format is always 2 for 16 bit, and the number of acquisitions is returned by the count parameter. In all other modes, the count parameter always returns 1.
- **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 2) 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 (INP1-2) 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 54510A 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
    :COUPling <type>
    :COUPling ?
    :ECOunt <events>
    :ECOunt?
    :HYSTeresis <mode>
    :HYSTeresis?
    :LEVel <level>
    :LEVel?
    :SLOPe <polarity>
    :SLOPe?
    :SOURce <source>
    :SOURce?
```

:TRIGger:COUPling

TRIGger:COUPling *<type>* is used to select the input impedance for the EXTernal TRIGger connector. The coupling can be set to DC or DCFifty.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
type	discrete	DC|DCFifty	None

Example

Set EXT TRIG input impedance to 50Ω

`TRIG:COUP DCF`	*Input impedance to 50Ω*

Comments

- **Impedance:** DC is 1MΩ and DCFifty is 50Ω.
- ***RST Condition:** Defaults to DC.
- **Related Commands:** TRIGger:COUPling?

:TRIGger:COUPling?

TRIGger:COUPling? returns the currently selected EXTernal TRIGger connector impedance. The data is sent to the output buffer. Returns DC for 1MΩ and DCF for 50Ω.

Example

Querying EXT TRIG input impedance

`Dimension statement`	*String for data*
`TRIG:COUP DCF`	*Input impedance to 50Ω*
`TRIG:COUP?`	*Query instrument to return impedance selected*
`enter statement`	*Enter value into computer*

: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 (ON or 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:** Internal trigger *level* can be entered to a value that is ±1.5 of the current CHANnel<n>:RANGe setting **from** the current CHANnel<n>:OFFSet setting. External trigger level can be entered over a range of ±2.0 V (trigger sensitivity is then 100 mVp-p into 50Ω).
- **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. Source can be specified as channel 1 or 2, ECL Trigger lines 0 or 1, or the external trigger input.

Parameters

Parameter Name	Parameter Type	Range of Values	Default Units
source	discrete	INPut*n* (*n*=1 to 2)| EXTernal|ECLTrg*n* (*n*=0 to 1)	none

Example

Set trigger source to ECL trigger bus line 1

```
TRIG:SOUR  ECLT1
```
Source to ECL trigger bus line 1

Comments

- **Selecting Source:** *source* is specified depending on the desired source as follows:

  INPut1-2: Source is input connectors 1-2 on the instrument panel.

  EXTernal: Source is signal connected to the EXTernal TRIGger connector on the instrument panel.

  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 1-2 only. ECLTrg lines cannot be used for an autoscale.
- ***RST Condition:** Defaults to INPut1.

:SOURce?

TRIGger:SOURce? returns the selected trigger source (INPut1-2 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  ECLT1        Source to ECL trigger bus line 1
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 54510A 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	Add, subtract, multiply, diff command operates the same in both languages.
	FUNCtion<n>:MULTiply FUNCtion<n>:SUBTract FUNCtion<n>:DIFF FUNCtion<n>:INTegrate	
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.
BORDer	FORMAT:BYT MSBF|LSBF	Command operates the same in both languages.
BORDer?	FORMAT:BYTeorder?	Command operates the same in both languages.
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.
DETector		
:MODE	TIMebase:SAMPle:MODE & ACQuire:TYPE	In COMP, RAWData is selected using ACQuire:TYPE and sample mode using TIMebase:SAMPle.
:MODE?	TIMebase:SAMPle:MODE? & ACQuire:TYPE?	See DETector:MODE for an explanation.

Command Cross Reference to COMP Commands — Continued

SCPI Command	COMP Command	Description
[SENSe:]		
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.
: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.
:NVPRotect	SYSTem:NVPRotect	Command operates the same in both languages.
:NVPRotect?	SYSTem:NVPRotect?	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		
:BORDer	WAVeform:FORMat:BYT	Command operates the same in both languages.
:BORDer?	WAVeform:FORMat:BYT?	Command operates the same in both languages.
[:DATA]	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.
: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		
:COUPling	TRIGger:COUPling	Command operates the same in both languages.
:COUPling?	TRIGger:COUPling?	Command operates the same in both languages.
: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 48) 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 48) 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 E1428A 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, multiplies, differentiates, or integrates 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 (1-2) for the attenuation calibration.
	:TNULl*<time>*	Enters a time null value for input pair 1-2.
	: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 (1-2).
	:DOUTput *<level>*	Sets the DC Calibrator output to 0 volts or 5 volts.
	:LTCalibration	Configures for a logic trigger calibration on the input specified (1-2).
	:TNULl *<input_skew>*	Configures for a time null calibration on the input pair 1-2.
	:VERTical	Configures for a vertical calibration on all inputs.
	:TNULl *<value>*	Enters time null values for input pair 1-2.
	:TNULl?	Returns the current time null values for input pair 1-2.
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	:BORDer<*type*>	Selects word data output sequence.
	:BORDer?	Returns word data output sequence currently selected.
	[: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 (1-2), 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.
	:MAP? <*source*>	Returns the address assignment and size for measurement
	: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 Trigger Output on the front panel connector.
	[:STATe] <*mode*>	Enables or disables the Trigger Output from the connector.
	[:STATe]?	Returns the current 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.
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.
[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 acquisition type (scalar or envelope) currently selected.

SCPI Commands Quick Reference — Continued

Subsystem	Command/Parameter	Description
[SENSe:]	CORRection*<number>*	Correction subsystem.
	:AFACtor *<atten>*	Enters the specified input's (1-2) probe attenuation factor.
	:AFACtor?	Returns the current probe attenuation factor for the specified input.
	:DETector	Detector subsystem
	:SAMPle	Selects realtime or repetitive as the sample timebase mode.
	:SAMPle?	Returns the current sample timebase mode.
	[:FUNCtion] *<type>*	Not used.
	[:FUNCtion]?	Always returns "SAMPle"
	:MODE *<mode>* [,*<length>*] [,*<acquisition>*]]	Selects realtime or repetitive as the sample timebase mode, and RAWData acquisition type is length and acquisition are specified.
	:MODE?	Returns the current sample timebase mode, and if previously specified, length and acquisition.
	INPut<number>	Input subsystem.
	:COUPling *<type>*	Selects the specified input's (1-2) coupling (AC or DC).
	:COUPling?	Returns the specified input's (1-2) current coupling selection.
	:FILTer	Filter selections.
	[:LPASs][:STATe] *<mode>*	Enables or disables the specified input's (1-2) low pass filter.
	[:LPASs][:STATe]?	Returns the specified input's (1-2) current low pass filter state.
	:HPASs[:STATe] *<mode>*	Enables or disables the specified input's (1-2) high pass filter.
	:HPASs[:STATe]?	Returns the specified input's (1-2) current high pass filter state.
	:IMPedance *<value>*	Selects the specified input's (1-2) impedance (1MΩ or 50Ω).
	:IMPedance?	Returns the specified input's (1-2) current impedance selection.
	[:STATe] *<mode>*	Enables or disables the specified input (1-2).
	[:STATe]?	Returns the specified input's (1-2) 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 (1-2) offset.
	:OFFSet?	Returns the specified input's (1-2) current offset value.
	[:PTPeak] *<range>*	Enters the specified input's (1-2) full scale vertical range.
	[:PTPeak]?	Returns the specified input's (1-2) 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 1-2, 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>*|EXTernal	Reports the specified input's (1-2) or external trigger as 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 (1-2) 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 (1-2) 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*>|EXTernal	
	:GAIN	Reports the specified input's (1-2) 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 (1-2) or external trigger's 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 (1-2) 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 input 2's or external trigger's 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 (1-2) or external trigger's 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.
	: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.
	:NVPRotect	Non-volatile RAM state and password commands.
	:PASSword *<old>*, *<new>*	Changes the current password.
	[:STATe] *<mode>*, *<pass>*	Selects the state of non-volatile RAM.
	[:STATe?]	Queries the current state of non-volatile RAM.
	: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	:COUPling	Defines input trigger coupling.
	:COUPling?	Return trigger coupling currently selected.
	: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 1-2, ECLTrg 0-1, or EXTernal).
	: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 48) 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 48) 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 E1428A Digitizing Oscilloscope.

Vertical

Bandwidth (–3 dB, dc coupled):[1] dc to 250 MHz

Rise Time:[2] 1.4ns

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] ±(1% of ch. offset + 2% of full scale)

Voltage Measurement Accuracy (dc)[4,5]
Dual Cursor: ±(1.25% full scale + 0.032 x V/Div)
Single Cursor: ±(1.25% full scale + offset accuracy + 2% of full scale)

Horizontal

Delta-t Accuracy[6]

Repetitive (≥8 averages): ± (0.005% x delta-t + 2E-6 x delay setting + 110 ps)

Real-time: (single acquisition): ± (0.005% x delta-t + 2E-6 x delay setting + 150 ps)

Trigger

Trigger Sensitivity[4]

Internal (dc to 50 MHz): 0.5 division

Internal (50 MHz to 250 MHz): 1.0 division

External (dc to 250 MHz): 100 mVp-p into 50 Ω

NOTES:

1. Upper bandwidth reduces by 2.5 MHz for each °C above 35°C. Bandwidth in Repetitive mode is typically > 300 MHz.
2. Rise time figure are calculated from: tr = 0.35/Bandwidth.
3. On ranges ≤50 mV the maximum overdrive of the input must not exceed 100 V.
4. Magnification is used below 7 mV/div range so vertical resolution and accuracies are correspondingly reduced. Below 7 mV/div, full scale is defined as 56 mV.
5. Voltage measurement accuracy decreases 0.08% per °C from firmware calibration temperature. This specification is valid for a temperature range ±10° from software calibration temperature. Specification applies to both modes; repetitive and realtime (single acquisition).
6. Specification applies at the maximum sampling rate. At lower sampling rates the specification is ±(0.005% X delta-t + (2 X 10–6) X delay setting +0.15 X sample interval) for bandwidth limited signals (tr = 1.4 X sample interval). Sample interval is defined as 1/(sample rate). Specification also applies to those automatic measurements computing time intervals on identical slope edges (i.e. pos-pos, neg-neg).

Performance Characteristics

The following are performance characteristics of the HP E1428A Digitizing Oscilloscope.

Vertical

Switchable Bandwidth Limits
ac-coupled (lower -3 dB frequency): 7 Hz
LF reject (lower -3 dB frequency): 450 Hz
bandwidth limit (upper -3 dB frequency): dc to 30 MHz

Number of channels: 2 (simultaneous acquisition)

Vertical Sensitivity Voltage Range (all channels): 8 mV to 40 V

Vertical Gain Accuracy (dc):[1,2] ±1.25% of full scale

Vertical Resolution:[2] ±0.4% (8-bit A/D)
±0.1% (10 bits via digitize with averaging)

Maximum Sample Rate: 1 GSa/s

Waveform Record Length[3]**:** 8000 points real time, 500 points repetitive

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 50 MHz
30 dB: 50 to 250 MHz

Horizontal

Time Base Range: 10 ns to 50 s

Time Base Resolution: 20 ps

	Time Base Range	Available Delay
Delay Range (post-trigger)	10 ns—50 s	1,000 x range
Delay Range	1 µs—50 s	−16 x range
(pre-trigger)	10 ns—500 ns	−8 µsec

Trigger

Trigger Pulse Width (minimum):
Internal: 1.75 ns
External: 2.8 ns

Trigger Level Range
Internal: ±1.5 x voltage range from offset
External: ±2 V

NOTES:

1. Gain accuracy decreases 0.08% of full scale per °C from firmware calibration temperature. This characteristic is valid for a temperature range ±10°C from software calibration temperature. Characteristic applies to both modes; repetitive (eight or more averages) and real time (single acquisition).
2. Expansion is used below 7 mV/div range so vertical resolution and accuracies are correspondingly reduced. Below 7 mV/div full scale is defined as 56 mV.
3. Available over HP-IB, waveform record length is:

Real Time-	8,000 points
Repetitive-	500 points

Operating Characteristics

The following are operating characteristics of the HP E1428A Digitizing Oscilloscope.

Vertical

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 7 Hz 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.

Effective Resolution: The maximum sample rate and the number of bits in an oscilloscope's digitizer are too often used for comparing oscilloscopes. These specifications, however, do not describe performance under dynamic signal conditions. Effective Resolution is a figure of merit that describes the digitizing oscilloscope's performance under dynamic conditions, and is measured using the sinewave curve fit test. This method considers:

- Quantization error
- Non-linearities (including preamp and A/D)
- System noise
- Frequency of input signal

All of these affect the effective resolution of the instrument. Some manufacturers specify effective bits using half-scale sinewaves. While the effective bits performance using half-scale testing is overstated when compared to full-scale testing, Hewlett-Packard publishes both sets of numbers for the HP E1428A so that, when comparing effective bits performance between digitizing oscilloscopes, a fair comparison can be made.

The HP E1428A's typical performance for a single acquisition is shown below:

Frequency	50 kHz	1 MHz	20 MHz	50 MHz	100 MHz	250 MHz
Full scale	7.2	6.7	6.3	5.6	5.0	4.9
Half scale	7.4	7.1	7.0	6.4	6.0	5.2

Note

For more information about effective resolution, please contact your Hewlett-Packard sales office, and ask for Product Note 5180A-2, *Dynamic Performance Testing of A to D Converters*, (pub # 02-5952-7629).

Horizontal

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 channels 1 and 2, ECL trigger 0 or 1, or on the external trigger input.

Note

The remaining trigger modes are only available when using the COMPatible programming language.

Pattern Trigger: A pattern can be specified using channels 1, 2 and the external trigger input. 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 (user-specified time is in brackets):

- 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 > 1.75 ns (2.8 ns for the external trigger) 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 1.75 ns, (2.8 ns for the external trigger).

State Trigger: A pattern is specified on any two of the three inputs with the third 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 three inputs. After the delay, an occurrence of a rising or falling edge of any one of the three 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 three 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 three 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 three 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 E1428A'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 (Single) 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.

Oversampling Filter: On time/division settings when less than 500 points are acquired across the range (≤200 ns) a built-in digital filter automatically reconstructs the data. This filter is a combination between a (Sin X)/X and a Gaussian filter.

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.

Sequential Single-shot Data Acquisition and Transfer Rate: Using the "Raw Data" mode, the HP E1428A can automatically capture, store, and label a waveform; and re-arm the trigger; and then repeat this process until the specified RAM is filled (on-board internal RAM is 200k per channel). Once the specified number of waveforms have been captured and stored, the entire block of waveforms can be transferred to the external computer. Users can specify the number of points to be stored and the number of waveforms to be captured. Repetition rates vary depending on record length and time base setting (slower sampling rates).

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, differentiate, integrate, 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 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.

Automatic Pulse Parameter Measurements

The E1428A 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 E1428A

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 E1428A 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: Forty-eight setups (1-48) 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

Probe Compensation, Calibrator/Trigger Output: BNC female connector provides the following signals:

Calibrator: provides a signal (≈500 Hz) 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.

Trigger: provides a 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: provides the output used for vertical calibration of the E1428A.

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 (32°F to +131°F)
Non-operating: –40°C to + 70°C (–40°F to +158°F)

Humidity

Operating: up to 95% relative humidity (non-condensing) at +40°C
Non-operating: up to 90% relative humidity at + 65°C (+149°F)

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 1.8 kg (4 lb.)

Shipping: approximately 5.9 kg (13 lb.)

B

Oscilloscope Error Messages

Table B-1 lists the error messages associated with the Oscilloscope module programmed using Hewlett-Packard 54510A 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
70	Ram write protected
-100	Command error (unknown command)
-101	Invalid character received
70	Ram write protected-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	Non-volatile RAM in protect state.
–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 E1428A 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 system architecture which allows instrument modules from different manufacturers to be integrated into the same mainframe. In addition to standardization, major advantages of VXI instruments have been:

- Modularity
- Size Reduction
- High Speed Measurement Throughput

It is fairly easy to see the size reduction provided by an VXI instrument, but quantifying throughput is somewhat more complicated. Most test engineers are trying to obtain the maximum throughput from each instrument to reduce test time. This in turn reduces the number of required systems, providing cost savings for ATE systems projects.

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 E1428A. A good understanding of the instrument and proper setups can increase throughput by nearly 10 times.

This appendix describes the basic operation of the HP E1428A, its measuring capabilities and provides some typically measurement and acquisition times using optimum setups and test software. With a little education up front, the user can maximize measurement and waveform throughput and use the oscilloscope to its full potential.

Sampling Techniques

All digitizing oscilloscopes digitize waveforms, however the method with 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.

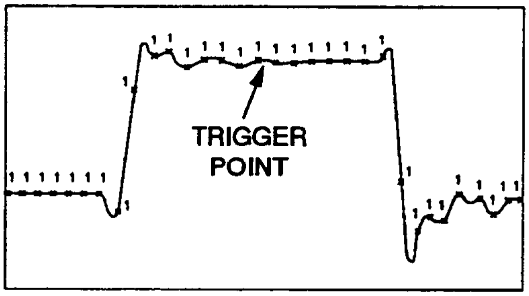
Real-time

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 (usually fairly expensive). Also, digital reconstruction to fill the points between samples is required. The HP E1428A uses 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

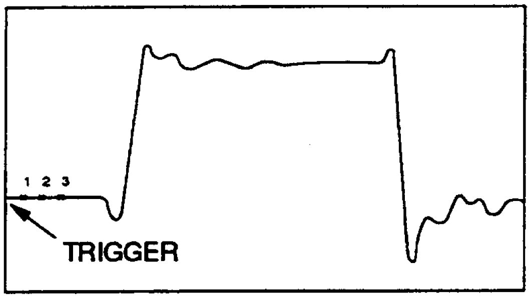
Sequential

Sequential sampling acquires one sample of the signal, on each occurrence of the trigger. 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 for 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 **on** the trigger, acquisition of pre- trigger data is limited. 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)

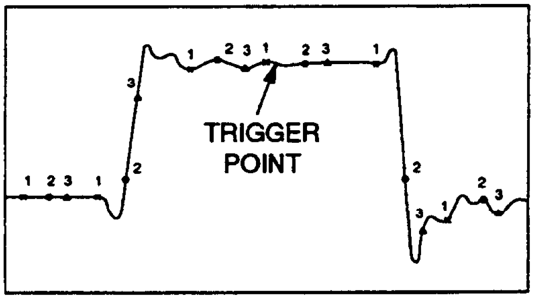
Random Repetitive

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. A high-speed track and hold circuit captures the signal voltage and holds it until the lower speed A/D converter can digitize it. After the sample is taken another circuit (fine interpolator) determines the time relationship between the sampled point and the trigger point. Knowing the time and voltage relationship of this point, the oscilloscope can place that point directly on screen. Depending upon the sweep speed setting, many points will be acquired on each trigger. Each sample will be separated by the sample period. The frequency of the sample clock is dithered to ensure that the trigger and the sample clock always have a random relationship to each other. Since the trigger and the sample clock are "random" to each other, the next trigger will cause points to be acquired which are 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.

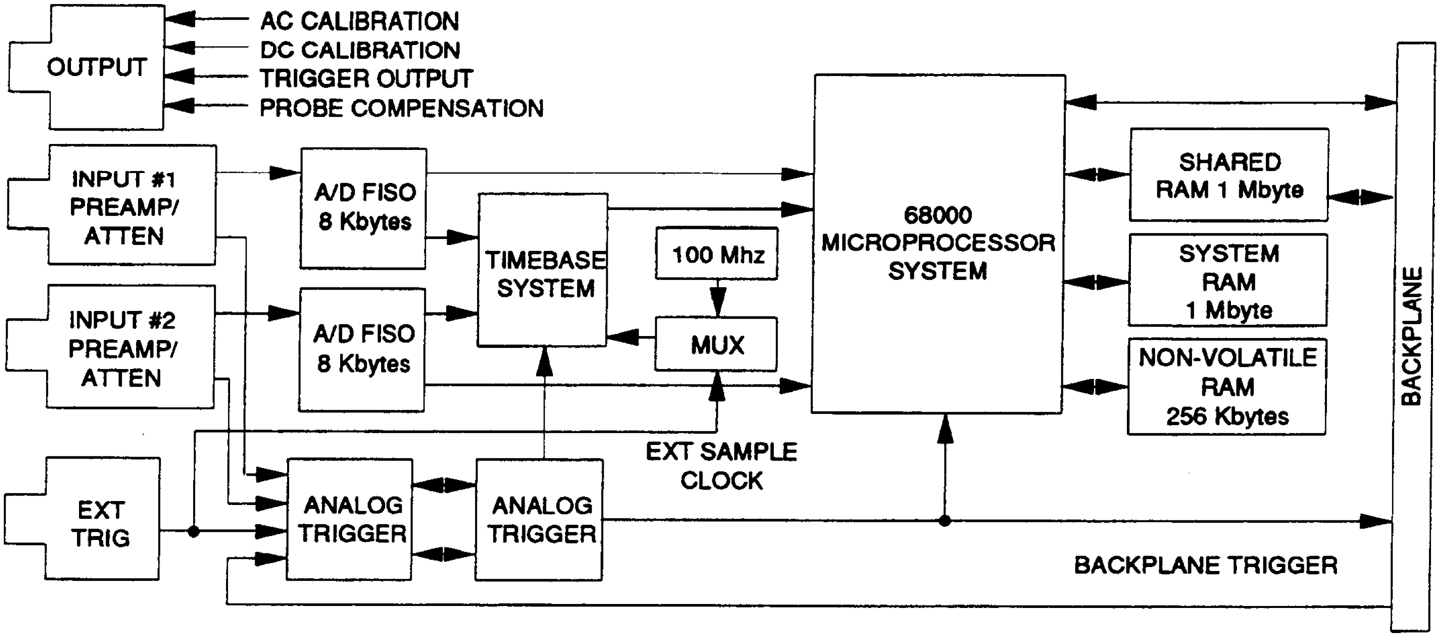
The figure shown below illustrates random repetitive sampling. Three acquisitions have been made, with six points acquired each time. Also, note that more than one point is acquired per trigger.



AFTER THREE ACQUISITIONS
(SIX POINTS PER ACQUISITION)

Theory of Operation

A block diagram for the HP E1428A is provided below.



The HP E1428A is capable of simultaneous data capture into two channels. Memory depth and data processing is identical for both inputs. The inputs are first conditioned by the preamplifier/attenuator (preamp/atten). Depending upon the voltage range setting, this module either amplifies or attenuates the input signal to provide the A/D converter and the analog trigger with the required signal amplitude.

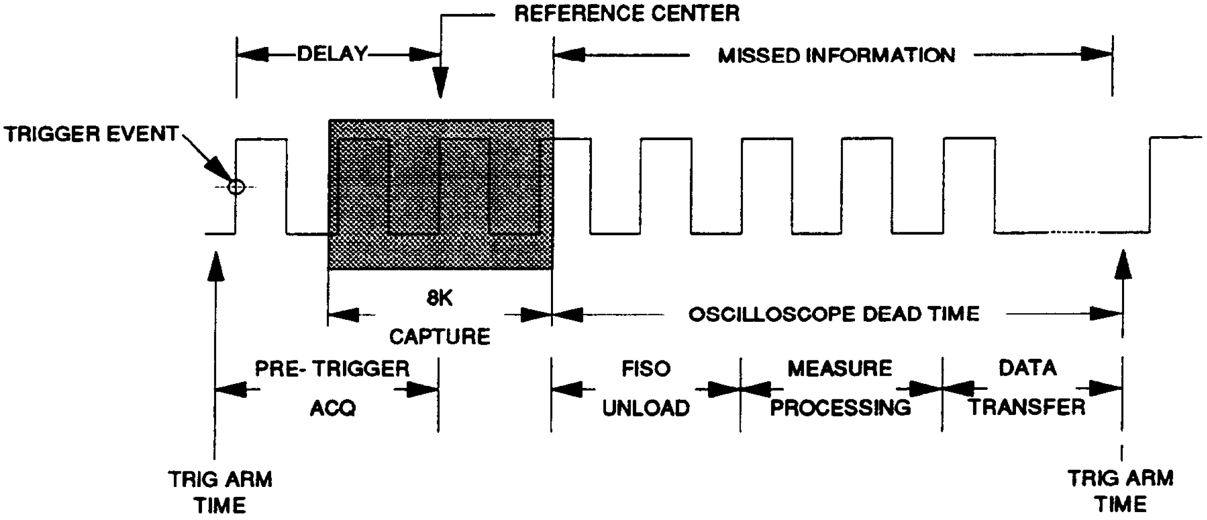
The A/D converter receives the signal and digitizes it at a maximum rate of 1 Gigasample per second (GSa/s). The sample rate varies from 50 Sa/s to 1GSa/s depending upon the time range setting. A high speed first-in-first-out (FISO) memory stores the numerical values as the A/D converter digitizes the signal. The FISO operates at the sample rate of the A/D converter, and has a memory depth of 8000 points. As the data is transferred to system memory each of the quantization levels of the 8-bit A/D converter is calibrated to produce 16-bit data. The combination of the A/D converter and the FISO is located on one hybrid module, and is referred to as the A/D FISO.

The timebase system determines the required sample rate from the time range setting as shown in the table below.

Range	Sample Rate	Sample Interval	8000 point Timespan	Real-time Reconstruction
10ns	1 GSa/s	1ns	8µs	yes
20ns	1 GSa/s	1ns	8µs	yes
50ns	1 GSa/s	1ns	8µs	yes
100ns	1 GSa/s	1ns	8µs	yes
200ns	1 GSa/s	1ns	8µs	yes
500ns	1 GSa/s	1ns	8µs	no
1µs	500 MSa/s	2ns	16µs	no
2µs	250 MSa/s	4ns	32µs	no
5µs	100 MSa/s	10ns	80µs	no
10µs	50 MSa/s	20 ns	160µs	no
20µs	25 MSa/s	40ns	320µs	no
50µs	10 MSa/s	100ns	800µs	no
100µs	5 MSa/s	200ns	1.6ms	no
200µs	2.5 MSa/s	400ns	3.2ms	no
500µs	1 MSa/s	1µs	8ms	no
1ms	500 kSa/s	2µs	16ms	no
2ms	250 kSa/s	4µs	32ms	no
5ms	100 kSa/s	10µs	80ms	no
10ms	50 kSa/s	20µs	160ms	no
20ms	25 kSa/s	40µs	320ms	no
50ms	10 kSa/s	100µs	800ms	no
100ms	5 kSa/s	200µs	1.6s	no
200ms	2.5 kSa/s	400µs	3.2s	no
500ms	1 kSa/s	1ms	8s	no
1s	500 Sa/s	2ms	16s	no
2s	250 Sa/s	4ms	32s	no
5s	100 Sa/s	10ms	80s	no
10s	50 Sa/s	20ms	160s	no
20s	25 Sa/s	40ms	320s	no
50s	10 Sa/s	100ms	800s	no

As the signal is being digitized, the fine interpolator determines the precise time of the data samples relative to the trigger event and records that time in a variable called Xorg. Xorg is unique for each acquisition and is recorded in memory along with each data record. After the data is stored in system memory, digital reconstruction is performed on the data if required by the acquisition setup. At the same time that the signal is being digitized, the trigger system accepts signals from either the input, the external trigger, or the backplane trigger to determine the proper signal on which to trigger. The logic trigger performs similar functions as a logic analyzer providing pattern detection, glitch capture, and qualifiers for various signal inputs. The powerful triggering functions of the HP E1428A are similar to that of a four channel logic analyzer.

The timing of the acquisition is shown below. When the trigger is armed, data begins to rotate through the 8k FISO memory waiting for the occurrence of a trigger event. This process is referred to as pre-trigger acquisition.



When a trigger occurs, the time base counts down to terminate the capture. Capture termination is based on the delay setting and reference (left, center, right). In the case shown previously, data is captured during the delay interval. The reference is set to center, so when the trigger occurs, the data capture continues until the delay counts down to zero. Capturing continues until the last 4 kbytes of memory is filled.

After the capture is terminated, the microprocessor must read the trigger interpolator, calculate the exact time of the first point in the record (called Xorg), and then move the data from FISO memory to the slower microprocessor memory. In the case shown previously, all this activity is lumped into FISO unload. After the data is in the microprocessor memory, post-processing continues. Functions are executed and measurements are made if they are turned on. All the activity following a capture is combined into a term called the oscilloscope's "dead time".

Once waveforms are in the scope's buffer, automatic pulse analysis and measurements are possible.

CPU

The CPU section consists of a 68000 microprocessor and 1 Mbyte of system RAM. System RAM is used both for waveform storage and miscellaneous data storage.

The non-volatile ram consists of 256 kbytes, and is used for calibration factor storage, waveform storage, and setup storage.

There is also 1 Mbyte of shared ram resident on the HP E1428A. This memory can be used to store waveforms or measurements allowing controllers direct access to the HP E1428A's memory and thereby bypassing slow waveform transfer routines. The lower 256 kbytes of shared RAM are also non-volatile.

Waveform Storage

There are three types of waveform "buffers" used in the HP E1428A.

The "display" buffer, buf_500, is 500 points in depth. This buffer corresponds to the display on the HP 54510 oscilloscope. Even though the oscilloscope acquires 8000 points per acquisition, only 500 are displayed on the screen, and thus are placed in the "display" buffer. Measurements are performed only on these 500 points and not the 8000 points. The "display" buffer also corresponds to the timebase range and contains reconstructed data if required at the timebase setting.

The 8000 point buffer, buf_8000, holds the entire acquisition record. All of this data is raw data (no reconstruction performed on it).

The sequential single-shot buffer, buf_SEQU, holds the data from sequential acquisitions when the oscilloscope is in the sequential single-shot mode.

There are also four non-volatile waveform memories in the HP E1428A. These memories are 8,000 points in depth and hold the timing and voltage data for stored waveforms.

Real-time Mode

In the real-time mode each acquisition **ALWAYS** acquires 8000 points, and stores this data into the waveform buffer. The CPU determines which of these 8000 points will fit within the timebase range, and then take those points and place them in the corresponding time buckets in the "display" buffer. At small time ranges (e.g. 10 ns), only 10 of the 8000 points will go directly into the "display" buffer. The remaining points will be determined with the reconstruction algorithms.

Only the points in the "display" buffer are used for measurements, even though all 8000 points are available to the user. The sample rate of the A/D converter is dependent on the time range setting as shown in the previous table.

At slow sweep speeds the sample rate is set at 1/500 of the time range setting. This will provide 500 points across the "display" buffer, plus 7500 more points which will not lie within the buffer. At slow sweep speeds, no digital reconstruction is needed and the acquisitions are fast. At a maximum sample rate of 1 GSa/s, the "display" buffer will be filled up for a time range of 500ns. Since the sample period is 1ns at this sample rate, the A/D converter will provide 500 points for the whole time range and no digital reconstruction is required. As the time range is decreased (sweep speed increased), fewer points will be acquired to fit within the display buffer, and the remaining points must be computed using a digital reconstruction filter.

As an example, at a time range of 100ns the A/D converter will acquire 100 points which fit within the display buffer. The remainder of the 500 points (400) must be computed using digital reconstruction. At the minimum time range of 10ns, only 10 points are acquired which fit within the display buffer. The other 490 points are computed.

Note Reconstruction is NOT performed on the remaining 7500 points acquired, since they are not used when performing measurements.

Advantages to real-time sampling are that a complete waveform is captured on a single acquisition. The disadvantages are a lower capture rate, and therefore, slower measurement throughput.

Random Repetitive Mode

The "best-case" timing resolution in the repetitive mode (one acquisition) is limited to 1-ns as a result of the HP E1428A's 1-GSa/s, Analog-to-Digital converters. Digital filter reconstruction is not implemented in the repetitive mode, because random repetitive acquisitions normally fill in the waveform to a resolution of up to 20 ps.

In random repetitive mode the full record length (8000 points) is not used and only the points which fit within the 'display buffer' are acquired. This means that in this mode, only a maximum of 500 points are saved. Digital reconstruction is not used in this mode, however, it may take more than one acquisition to fill up the entire 500 point display buffer. At time ranges of ≥500ns, the sample rate is fast enough to fill up the entire display buffer on 1 acquisition. As the time range is decreased more acquisitions will be required.

As an example, at a time range of 10ns, only 10 points are acquired per acquisition(10 points x 1ns/point = 10ns). Therefore, many acquisitions will be needed to get the full 500 points.

The advantages of this mode are better time resolution (math computation is not required for digital reconstruction). Random repetitive mode is the fastest capture mode and allows averaging. The disadvantage is that at fast sweep speeds more than one acquisition is needed to completely fill the 500 point buffer.

Sequential Single-Shot

Another mode for capturing individual waveforms one at a time indefinitely, without missing information between captures, is the rawdata mode. In the rawdata mode, the record length and the number of waveforms to capture can be specified. If you specify a single waveform capture, then the time to capture and process it is kept short just as it is for the real-time and repetitive modes. The rawdata mode, with a capture of one waveform, is similar to the 8k real-time mode, with the exception that the record can be set to variable lengths. The number of points in a record can be set as low as 4 or as high as 8000.

The entire purpose of the sequential single-shot mode is to provide fast sequential acquisitions. The oscilloscope performs an acquisition, unloads the FISO into system memory (or shared memory) and rearms itself for the next acquisition. The number and length of each acquisition determines the required amount of memory. With each acquisition, the xorg value is also stored to provide the user with the precise time relationship of the sampled data relative to the trigger event. After all the acquisitions are complete, the oscilloscope calibrates the data. In this method the maximum amount of overhead is saved for last to provide the shortest dead time for the user.

Shared RAM Usage

The HP E1428A contains 1 Mbyte of shared RAM. This RAM is basically a dual port RAM which can be accessed by either the resident CPU section, or via the backplane (i.e. other instruments, or controllers). By providing this type of storage, the traditional waveform transfer bottleneck (interface cable) is now eliminated. A typical operation would consist of the oscilloscope acquiring a waveform and storing it directly into shared memory. The controller could then have direct access to the waveform data without having to transfer it.

The measurement queue can also be accessed through shared RAM. The queue can hold a maximum of 8 measurements and can be placed in a continuous mode in which the measurements are updated after each acquisition. This queue can then be accessed directly as a part of the external controller's memory, and no point to point transfer is required.

The waveform buffers and measurement queues would take up a very small portion of the 1 Mbyte shared memory. The purpose of such a deep memory becomes obvious when considering sequential single shot mode. Since the number of possible acquisitions is determined by the depth of the available memory, 1 Mbyte would provide much more acquisition storage than the oscilloscope's private system memory. Transferring 1 Mbyte of data takes a long time over the bus or word serial protocol, however, now this waveform data is directly available to the system controller.

When the oscilloscope is placed in the shared RAM mode, the active waveform buffers, the measurement queue, and the sequential single shot buffers are all placed in specified shared memory area. The shared RAM does not need to reside on the HP E1428A itself, but can be located in the controller. A minimum size buffer is required, and if there is enough shared RAM to satisfy this constraint, it may be located anywhere in an available VME address space. If the buffer is located only in the oscilloscope's private system memory, the waveform transfer from the FISO to the system RAM will be most efficient, and the sequential single shot acquisition rate will also be most efficient.

When the buffers are placed totally in the on-board shared RAM, there is a 30% penalty placed on the sequential single shot acquisition rate because of the arbitration overhead needed to access shared memory locations.

Finally, if any part of the buffers are located in external shared memory, an even larger penalty will be noticed because of both memory access times and arbitration time.

High Speed Waveform Acquisition

The HP E1428A contains 1 Mbyte of shared RAM. This RAM is basically a dual port RAM which can be accessed by either the resident CPU section, or via the backplane (i.e. other instruments, or controllers). By providing this type of storage, the traditional waveform transfer bottleneck (interface cable) is now eliminated. A typical operation would consist of the oscilloscope acquiring a waveform and storing it directly into shared memory. The controller could then have direct access to the waveform data without having to transfer it.

Basically, the HP E1428A is used for either waveform acquisition, or waveform measurement. By understanding the previous explanation of the E1428A's operation, it will now be easier to discuss the speed tradeoffs associated with waveform acquisitions.

As discussed previously, the types of acquisitions can be divided into three main categories:

- Real-time
- Sequential (single-shot)
- Random Repetitive

It is important to understand how each operates so that the benefits of each can truly be achieved.

Note For the fastest continuous throughput, the HP E1428A should be operated in the repetitive mode.

Real-time

In the real-time mode a SINGLE acquisition is required to fill the entire "display" buffer of 500 points. Time base is essentially the only factor which affects the acquisition speed of the oscilloscope in real-time mode.

Timebase Setting

The timebase setting determines the sample rate and also determines the time width of the "display" buffer. The previous table specifies the various timebase ranges with the respective sample rates and real-time waveform buffer span.

The real-time mode ALWAYS acquires 8000 points on each acquisition. These 8000 points fill up the real-time waveform buffer and determine the 8000 point time span. However, even though 8000 points are acquired, this does not guarantee that the "display" buffer will automatically be filled up (before reconstruction).

As an example, if the sample rate of the A/D converter is 1GSa/s, each point will be separated by 1ns. At a time range of 10ns there will only be 10 points acquired directly which fill the "display" buffer of 500 points. The remainder of the 500 points (490) must be computed using digital reconstruction filters. Of course digital reconstruction takes some added time, about 20 ms. Therefore the timebase range will determine whether or not digital reconstruction is required. From the previous table, note that at a display width (timebase range) of 500ns, the sample interval will be 1ns and therefore the entire display buffer will be filled up on a single acquisition, and digital reconstruction is not required for this setting. At larger time ranges (greater than 500ns) the sample rate is reduced to always provide 500 points to the "display" buffer on a single acquisition, and there is no need for reconstruction. At time ranges smaller than 500ns (10ns-200ns), the "display" buffer will not be filled up on a single acquisition, and reconstruction will be required. This will require about 20ms to perform the computations and therefore these time ranges will be slower than those ≥ 200ns.

At very slow time ranges (less than 1ms) the acquisition time is affected by the pre and post trigger data storage. For example, even though the time range is set at 1ms, the actual real-time waveform buffer has a time span of 16ms, so at least 16ms is needed to fill the entire real-time waveform buffer. At a time range of 50s the real-time waveform buffer would have a span of 800s requiring almost 14 minutes to perform a single acquisition.

Maximum Throughput

Selecting Time Base For Maximum Acquisition Speed: If possible, use time ranges from 500ns through 1 ms to perform the fastest acquisition.

Random Repetitive

For the fastest continuous throughput, the HP E1428A should be operated in the repetitive mode. Even though this mode is primarily used for repetitive waveform acquisition, it can also be used for single-shot capture. Actually, random-repetitive sampling is made up of many single-shot captures.

The repetitive mode of acquisition is the fastest conventional acquisition mode for a couple of reasons. First, this mode limits the waveform record length to 500 points. This length is often sufficient memory depth for single pulse analysis. Less points to process will naturally result in faster throughput. Secondly, digital filter waveform reconstruction is not executed in the repetitive mode. Not running the filter minimizes waveform processing dead time.

Random repetitive sampling is a bit more complicated than real-time sampling, but it can have a major time savings for each acquisition (over real-time sampling) if used properly. An important point to remember is that in the repetitive mode a maximum of 500 points are acquired with each acquisition. This is a big difference compared to real-time mode which acquires 8000 points. This saves time in unloading the FISOs and calibrating the data. On top of this no digital reconstruction is required in this mode. There are four factors which determine acquisition time in random repetitive mode.

Timebase Setting

The timebase setting will affect the acquisition time, especially at small time ranges. The sampling rate of the A/D converter varies with time range as previously shown in the table. As previously discussed, random repetitive sampling usually requires multiple acquisitions to build an entire waveform. This is usually the case since such architectures usually use a slower, less-expensive A/D converter. However, the HP E1428A has a high speed converter which translates into fewer acquisitions required to build up the waveform. When possible, the A/D converter samples at the rate needed to fill up the 500 point buffer.

As an example, at a time range of 500µs, the sample interval would have to be 1 µs to fill up the buffer on one acquisition. This translates into a sample rate of 1MSa/s. At 1GSa/s, the oscilloscope can acquire a full buffer at a time range of 500ns. At time ranges smaller than this, multiple acquisitions are needed to fill up the entire buffer. At a range of 10ns, only 10 points will be acquired per acquisition so at least 50 acquisitions will be needed to fill up the "display" buffer.

At very long time ranges (10ms or longer) a significant amount of time is required to store pre and post trigger information which will slow down acquisition time.

Maximum Throughput

Selecting Time Base For Maximum Acquisition Speed: If possible, use a time range of between 500ns and 1ms to obtain the fastest acquisition time. At smaller time ranges more acquisitions are required to fill the "display" buffer, and at longer time ranges, more time is needed to store pre and post trigger information.

Repetition Rate

Another factor which affects digitization speed is the repetition rate of the trigger event. If more than one acquisition is needed to build up the waveform, and only one trigger comes per minute, it will take several minutes to complete a digitization. If a trigger happens every microsecond (1 MHz square wave) there will be many trigger events which will not be acquired since the acquisition hardware cannot reset itself in less than a microsecond. There will be a certain amount of "blind" time in which triggers will not be acquired. This is the time it takes the microprocessor to transfer and calibrate the data from the FISO to the waveform memory, and to reset the trigger. The shortest "blind" time on the HP E1428A is 1ms. This means that for maximum digitization speed, a trigger should occur at a rate faster than 1/1ms, or faster than 1 kHz. If the repetition rate of the signal is slower than 1kHz, the oscilloscope may not be digitizing at its fastest rate.

As an example, if a 1 Hz signal is used to generate the trigger, there will be some time when the scope is waiting to get a trigger slowing down the digitizing rate. The repetition rate is important only at the small time range settings because at these ranges, more than one acquisition is needed to complete a digitize operation. At time ranges of ≥500ns, only one acquisition is needed for a complete digitize operation.

Maximum Throughput

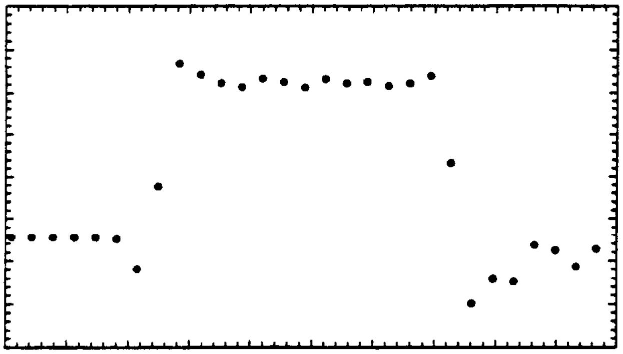
Trigger Repetition Rate: At time ranges ≤ 500ns, it is important to have a trigger that has a repetition rate of ≥ 1kHz to ensure that the oscilloscope does not wait a long time for the next acquisition.

Completion Criteria

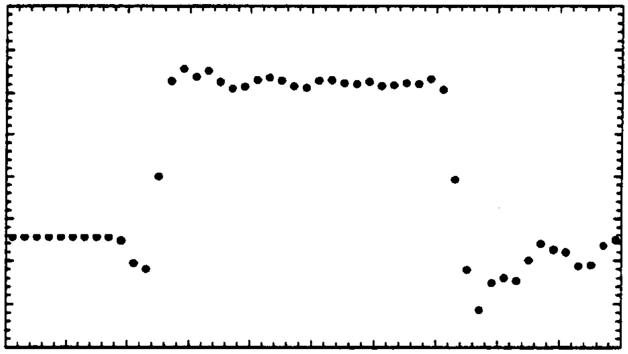
Another factor which will affect digitizing speed is completion criteria. If the scope is set to acquire 500 points at a completion criteria of 100%, it will take much longer to digitize that signal than one specified with a 50% completion criteria. Completion criteria is not a linear relationship, because 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 must be filled. The randomness of the trigger and sample clock time relationship ensures that all the points will eventually be filled. This parameter is important only at time ranges of less than 500ns, since at slower ranges, all the points are acquired on a single acquisition (basically 100% complete).

Does it matter how complete the waveform is? 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.

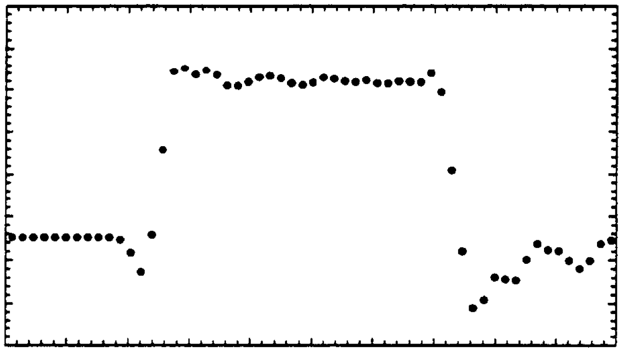


20% Complete

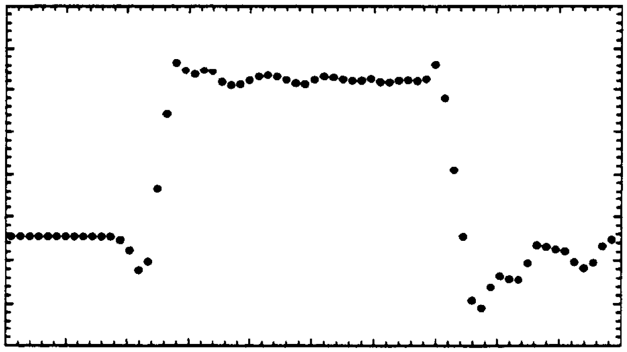


40% Complete

- As the completion percentage increases, the waveform becomes more discernible. At 60% and 80% completion criteria, the waveform is very discernible.

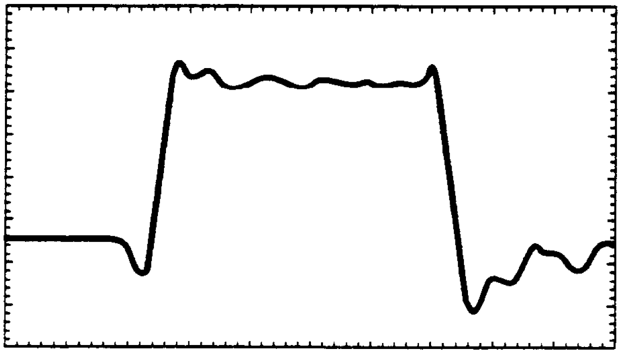


60% Complete



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: At time ranges of smaller than 500ns, use a completion criteria of 60% to provide a waveform with reasonable clarity if a full 500 points is not needed.

Averaging

Averaging is a method which the oscilloscope uses to reduce the amount of random noise present on the signal, and to improve vertical resolution. If the oscilloscope is set up to average 16 times, this means that each time point must be hit 16 times, and then use the average of all 16 values. 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.

Maximum Throughput

Selecting Acquisition Mode For Maximum Digitizing Speed: The acquisition mode should be set to normal or scalar unless the signal is noisy, or if high accuracy and repeatability are required using an averaging filter.

Sequential Single Shot

In sequential single-shot, only calibrated data is available without reconstruction. Because of this, the timebase range will not affect acquisition time, except that a larger sample period corresponds to a longer acquisition time. The amount of dead time between acquisitions is directly dependent upon the number of points acquired per acquisition. This dead time is attributed to unloading the FISOs and resetting the trigger and timebase circuitry. Short record lengths take less time to unload.

High Speed Measurements

There are two steps required in making an automatic measurement.

- The first step is digitizing the waveform, or digitization. The waveform on which the measurement will be taken must be acquired.
- The second step is measuring the parameter using the waveform data.

These are two separate steps and each require a certain amount of time to complete. In most cases the digitization process takes longer than the measuring process.

Once the waveform is digitized, many measurements can be taken on that waveform without the need for a new digitization. To determine the amount of time required for a measurement, add the "digitize" time to the "measure" time. Some typical numbers can be found in later in this appendix. The measurement times are fairly constant with changes in timebase setting. However, after each digitize and before a measurement is performed, a table is constructed with the acquired points. This table is only created once, so the first measurement may take longer than subsequent ones.

A measurement is performed on the 500 points in the "display" buffer. Even though 8000 points are acquired for a real-time or single-shot acquisition, only the 500 points in the "display" buffer are used for the measurement process.

Once the waveform is digitized, make as many measurements as required on that waveform. Do no do a digitize for each measurement.

Continuous measurement mode will increase digitize times.

Fast Throughput Summary

Note For fastest capture of single 500 point waveform see item #1 below.

For capture of one to xxx waveforms with record length selectable from 4 to 8000 points, see item #2 below. This method does not allow for built-in measurements to be performed.

1. **Fastest Measurement Throughput and fastest capture of a single 500 point waveform of time range 500ns or longer.**
   - Use repetitive sampling (TIMebase:SAMPle REPetitive)
   - Turn averages off (ACQuire:TYPe NORMal)
   - Time range ≥500ns (TIMebase:RANGe)
   - After waveform is captured, perform all needed measurements on the waveform

2. **Fastest Capture of Waveforms**
   - Realtime sampling (TIMebase:SAMPle REALtime)
   - Raw Data mode (ACQuire:TYPe RAWData)
   - Select any time range
   - Waveforms can be transferred using shared RAM

3. **For capture of time ranges from 10ns to 200ns, two choices are available.**
   - Repetitive Sampling (TIMebase:SAMPle REPetitive): The 500 point display buffer will be filled with multiple acquisitions at 1GSa/s. Select 60% completion (acquire:complete 60) since a higher percentage of completion takes more acquisitions and generally doesn't produce much higher accuracy in the measurements.
   - Realtime Sampling (TIMebase:SAMPle REALtime): An 8000 point record is acquired a 1GSa/s and then the 500 point "display" buffer is filled using digital reconstruction. (timebase:sample realtime)

Typical Acquisition and Measurement Times

This section compares the various digitize, measurement and transfer times of the HP 54510 (rack and stack), HP 54503 (rack and stack), the E1426A (VXI), and the E1428A (VXI). The times recorded are typical times.

Digitize Times

The "digitize" time (in milliseconds) is the time from when the digitize command is sent out until the Oscilloscope signals the computer that it has completed the operation.

Digitize times were as follows:

Time Range	E1428A	HP54510A	E1426A	HP54503A
≥500ns Real-Time	173	292	N/A	N/A
≥500 Repetitive	17	40	92	110
100 ns Real-Time	250	395	N/A	N/A
100ns Repetitive @ 60% complete	84	138	202	224

Transfer Times

The "transfer" time (in milliseconds) 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:

No. of Points	E1428A	HP54510A
500 points	30	30
8000 points	136	136

Measurement Times

The "measurement" time (in milliseconds) is the time from when the measurement query is sent until the number is returned.

Measurement times were as follows:

Measurement	E1428A	HP54510A	E1426A	HP54503A
Vamp, Vpp, Vmax, Vmin, Vtop, Vbase	37	85	34	76
Frequency, Period	132	190	65	107
Risetime, Falltime Pulsewidth	93	145	103	145

D

Programming Examples

Using This Appendix

This appendix explains the example programs provided with the HP E1428A. A few of the sample programs are included in printed form for your quick reference. This appendix contains the following sections:

Introduction

This appendix contains example programs using the command set for the E1428A digitizing oscilloscope. In general, the programs use the long form of the command with each command having a separate output statement for clarity. To optimize the program, switch to the concatenated short form.

Two 3.5 inch floppy disks are included with this manual, one in DOS format for the PC environment, and one in LIF format for the HP Series 200/300 controller environment.

- The DOS disk contains all the Microsoft QuickBASIC and Microsoft QuickC programs as well as copies of the HP-BASIC programs (HP-BASIC will not execute on DOS).
- The LIF disk contains only the HP-BASIC programs.

The sample programs are stored on the DOS disk in the following sample directories:

- Q_BASIC directory contains HP 54510 Compatible and SCPI programs written for the PC environment in QuickBASIC.
- QUICK_C directory contains HP 54510 Compatible and SCPI programs written for the PC environment in QuickC.
- HP-BASIC directory contains HP 54510 Compatible and SCPI examples for the HP Series 200/300 controller. These are stored on the DOS disk using a LIF-to-DOS utility. These are the programs on the LIF disk.

Note The majority of the programs were written in HP-BASIC, and are included on the DOS disk as a "how-to-do" reference only.

Microsoft is a U.S. registered trademark of Microsoft Corporation.

The programs are categorized by the programming language used and by the computer system. Examples in both HP 54510 Compatible and SCPI are in three different language platforms as follows:

- HP BASIC 5.0 running on an HP series 200/300 controller.
- Microsoft QuickBASIC 4.5 running on a Vectra, IBM PC, or compatible.
- Microsoft QuickC 2.5 running on a Vectra, IBM PC, or compatible.

Note To use these QuickBASIC and QuickC example programs on a PC, you must have the HP 82335A HP-IB interface. The HP-IB Command Library must be installed in your PC. See the manual for the 82335A for instructions on using the library with QuickBASIC or QuickC.

There is a README file on each disk, and also in each directory, that should be referenced for the latest information concerning the example files. Information includes instructions concerning addresses of the oscilloscope and the interface, etc.

A sample signal should be connected to Channel 1 of the scope with a BNC to BNC connector or a 1:1 probe when running these programs. If a different probe is used, the probe attenuation value needs to be set in the INITIALIZE function of the programs.

HP BASIC

INIT_COM – This sample program demonstrates some of the commands used to initialize the oscilloscope, capture data, and make parametric measurements.

The 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    !INIT_COM                 HP Basic Compatible program
20    !
30    !MAIN PROGRAM
40    !
50    CLEAR SCREEN
60    PRINT "This example program will perform the following tasks:"
70    PRINT "         a.  initialize the interface and scope"
80    PRINT "         b.  digitize signal                  "
90    PRINT "         c.  measure and print the frequency   "
100   PRINT
110   PRINT
120   PRINT "Before running this sample program, set addresses to your"
130   PRINT "system configuration. This sample program ASSIGNS"
140   PRINT "                       Isc to 7   (HP_IB interface)"
150   PRINT "                       Scope to 70901"
160   PRINT
170   PRINT "PRESS Continue WHEN READY TO START"
180   PAUSE
190   GOSUB Initialize                  !initialize interface and scope
200   GOSUB Get_waveform                !digitize signal
210   GOSUB Measure                     !measure and print frequency
220   STOP
230   !
240   !INITIALIZE INTERFACE AND SCOPE
250   !
260 Initialize:                         !
270   ASSIGN @Scope TO 70901            !scope address
280   ASSIGN @Isc TO 7                  !HP-IB address
290   CLEAR @Isc                        !clear HPIB interface
300   OUTPUT @Scope;":SYSTEM:LANG COMP" !set language to compatible
310   WAIT 1                            !WAIT 1 SEC FOR LANGUAGE SWITCH
320   OUTPUT @Scope;"*RST"              !set scope to default config
330   OUTPUT @Scope;":AUTOSCALE"        !AUTOSCALE
340   OUTPUT @Scope;":SYST:HEADER OFF"  !turn headers off
350   !
360   !The following acquire subsystem commands are used to specify
370   !the acquisition mode, percent completion, and number of points
380   !for the DIGITIZE.  These are the defaults set by the *RST, but
390   !are included here to stress the fact that you can control them.
400   !
410   OUTPUT @Scope;":ACQUIRE:TYPE NORMAL"   !normal acquisition mode
420   OUTPUT @Scope;":ACQUIRE:COMPLETE 100"  !100% completion criteria
430   OUTPUT @Scope;":ACQUIRE:POINTS 500"    !500 points acquired
440   CLEAR SCREEN                           !clear screen
450   RETURN
460     !
470     !DIGITIZE waveform to acquire data and stop scope for further
480     !analysis and measurement.
490     !
```

```
500 Get_waveform:                                 !
510   OUTPUT @Scope;":WAVEFORM:SOURCE CHAN1"  !set source to channel 1
520   OUTPUT @Scope;":DIGITIZE CHAN1"         !macro to acquire data & stop
530   RETURN
540     !
550     !have scope to a frequency measurement and read results into
560     !controller.
570     !
580 Measure:                                      !
590   OUTPUT @Scope;":MEASURE:FREQUENCY?"     !FREQUENCY query
600   ENTER @Scope;Value                      !read from scope
610   PRINT "Frequency = ";Value;"Hz"
620   OUTPUT @Scope;":MEASURE:VPP?"           !Vpp query
630   ENTER @Scope;Vpp
640   PRINT "Vpp = ";Vpp;" V"
650   RETURN
660   END
```

Microsoft QuickBASIC

INIT.BAS – This sample program demonstrates some of the commands used to initialize the oscilloscope, capture data, and make parametric measurements.

The example is written using:

- **an HP-IB select code of 7, primary address of 09, and secondary address of 01 for the Oscilloscope**
- **an Vectra, IBM PC, or compatible with Microsoft QuickBASIC 4.5**
- **An HP HP 8116A or suitable Pulse Generator**

Execute:

```
DECLARE SUB get.waveform ()
DECLARE SUB measure ()
DECLARE SUB Initialize ()
DECLARE SUB send (cmd$)
'
'Filename : \QB-HPIB\INIT_COM.bas
'
REM $INCLUDE: 'c:\hpib\qbsetup'
DIM SHARED isc&, scope&

CLS                                        'clear screen
PRINT "This example program will perform the following tasks:"
PRINT "         a.  initialize the interface and scope"
PRINT "         b.  digitize signal"
PRINT "         c.  measure and print the frequency"
PRINT
PRINT "The sample program assumes the following address settings:"
PRINT "      HP-IB interface select code = 3     value of isc& in main program"
PRINT "      scope address = 30901               value of scope& in main program"
PRINT "set these variables as needed for your system."
PRINT
PRINT "PRESS: 1 TO CONTINUE"
PRINT "       2 TO TERMINATE"
PRINT "?";
BEEP
LINE INPUT cmd$
IF cmd$ = "2" THEN SYSTEM

isc& = 3                                 'HP-IB interface select code
scope& = 30901                           'scope address

CALL Initialize                          'initialize interface and scope
CALL get.waveform                        'tell scope to acquire data
CALL measure                             'measure frequency of input signal

END

SUB get.waveform
   send (":digitize channel1")                 'tell scope to acquire data
END SUB

SUB Initialize
SHARED pcib.err, pcib.baserr, noerr

   CALL ioreset(isc&)                          'set interface to start-up state
   IF pcib.err <> noerr THEN ERROR pcib.baserr 'check for error
   timeout! = 10!
   CALL iotimeout(isc&, timeout!)          'set timeout to 10 seconds
   IF pcib.err <> noerr THEN ERROR pcib.baserr 'check for error

   CALL ioclear(isc&)                          'clear devices attached to the interfaces to known
 state
   IF pcib.err <> noerr THEN ERROR pcib.baserr 'check for error

   send (":system:language comp")              'set 54510 compatiblelanguage
   CLS
   PRINT "one second wait required for language switch.  PRESS Enter TO CONTINUE"
   BEEP
   LINE INPUT dummy$
   CLS

   send ("*cls")                               'clear status registers
   send ("*RST")                               'reset the scope
   send (":autoscale")                         'autoscale the input signal
   send (":system:header off")                 'turn headers off
   '
   'The following acquire subsystem commands are used to specify
   'the acquisition mode, percent completion, and number of points
   'for the DIGITIZE.  These are the defaults set by the *RST, but
   'are included here to stress the fact that you can control them.
   '
   send (":acquire:type normal")               'normal acquisition mode
   send (":acquire:complete 100")              '100% completion criteria
   send (":acquire:points 500")                '"500 points acquired

   CLS                                         'clear PC screen

END SUB

SUB measure
SHARED pcib.err, pcib.baserr, noerr
   send (":measure:frequency?")                'Query for result of frequency mesurement
   CALL ioenter(scope&, freq!)                 'Get the number from the scope
   IF pcib.err <> noerr THEN ERROR pcib.baserr 'Check if any error occur
   PRINT "Frequency = "; freq!; " Hz"          'Print out the result
   send (":measure:vpp?")                      'query for result of Vpp
   CALL ioenter(scope&, vpp!)                  'get result from scope
   IF pcib.err <> noerr THEN ERROR pcib.baserr
   PRINT "Vpp = "; vpp!; " V"
END SUB

SUB send (cmd$)
SHARED pcib.err, pcib.baserr, noerr
   CALL iooutputs(scope&, cmd$, LEN(cmd$))     'send command to scope
   IF pcib.err <> noerr THEN ERROR pcib.baserr 'check for any error
END SUB
```

Microsoft QuickC

INIT.C – This sample program demonstrates some of the commands used to initialize the oscilloscope, capture data, and make parametric measurements.

The example is written using:

- **an HP-IB select code of 7, primary address of 09, and secondary address of 01 for the Oscilloscope**
- **an Vectra, IBM PC, or compatible with Microsoft QuickC 2.5**
- **An HP HP 8116A or suitable Pulse Generator**

Execute:

```
#include "c:\qc25\chpib.h"
#include "c:\qc25\cfunc.h"
#include <graph.h>
#include <stdio.h>

send (char *cmd);
initialize();
get_waveform();
measure();
/*
      Filename : \QUICK_C\Init.c
*/
#define ISC     3L                     /* select code of HP-IB interface */
#define SCOPE   30901L                 /* address of scope */
#define TRUE    1

main()
{
   char ch;
   _clearscreen(_GCLEARSCREEN);              /* clear screen of PC     */

   printf ("This example program will perform the following tasks:\n");
   printf ("     1.  initialize the interface and scope\n");
   printf ("     2.  digitize the signal\n");
   printf ("     3.  measure and print the frequency and VPP\n");
   printf ("The sample program assumes the following address settings:\n");
   printf ("     HP-IB interface select code = 3   value defined by ISC\n");
   printf ("     scope address = 30901             value defined by SCOPE\n");
   printf ("Set these constants in the define statements for your system\n");
   printf ("PRESS Enter TO CONTINUE OR Ctrl C TO TERMINATE\n\n");
   ch = getch();
   initialize();                       /* initialization of interface & scope */
   get_waveform();                     /* tell scope to get waveform */
   measure();                          /* make measurement and print result */
}

error_handler(int error, char *routine)
{
char    ch;
if (error != NOERR)
   {
      printf("Error in call to %s \n", routine);
      printf("%d %s \n", error, errstr(error));
      printf("Press Enter to continue: ");
      scanf ("%c",&ch);
   }
}
```

```
initialize()
{
   char ch;
   error_handler(IORESET(ISC), "ioreset");          /* reset interface */
   error_handler(IOTIMEOUT(ISC, (double)10), "iotimeout");  /* timeout */
   error_handler(IOCLEAR(ISC), "ioclear");          /* clear interface */
   send (":system:language comp");                  /* 54510 compatible*/
   printf ("one second wait need to perform language switch\n");
   printf ("PRESS Enter TO CONTINUE OR Ctrl C TO TERMINATE\n");
   ch = getch();

   send ("*RST");                            /* reset scope to default */
   send ("*CLS");                            /* clear status registers */
   send (":autoscale");                      /* autoscale input signal */
   send (":system:header off");              /* turn headers off       */
   _clearscreen(_GCLEARSCREEN);              /* clear screen of PC     */
}

get_waveform()
{
   send (":digitize channel1");
}

measure()
{
   float        freq,vpp;

   send (":measure:frequency?");                    /* query frequency of input signal */
   error_handler(IOENTER(SCOPE, &freq), "ioenter"); /* get the freq. from scope */
                                                                                           /* noted
that freq must be passed by ref */
   printf ("Frequency = %e Hz \n", freq);           /* print it out */
   send (":measure:vpp?");                          /* query Vpp of input signal */
   error_handler(IOENTER(SCOPE,&vpp), "ioenter");   /* get vpp from scope */
   printf ("Vpp = %f V \n",vpp);                    /* print Vpp */
}

send(char *cmd)
{  /* this subroutine send the char string pointed by cmd to scope */
   error_handler(IOOUTPUTS(SCOPE, cmd, strlen(cmd)), cmd);
}
```

Index