Tektronix®

4 Series MSO (MSO44, MSO46)
5 Series MSO (MSO54, MSO56, MSO58, MSO58LP)
6 Series MSO (MSO64, MSO64B, MSO66B, MSO68B)
6 Series Low Profile Digitizer (LPD64)

Programmer Manual

Supports FW version 1.30.x
Released November 2020
www.tek.com

077-1305-12

Copyright © Tektronix. All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

FlexChannel, FastFrame, OpenChoice, iView, Pinpoint, RT-Eye, TekVPI, FastAcq, and e*Scope are registered trademarks of Tektronix, Inc.

TekSecure is a trademark of Tektronix, Inc.

4/5/6 Series MSO Programmer Manual. 077-1305-10. Released March 2020.

Contacting Tektronix

Tektronix, Inc.
14150 SW Karl Braun Drive
P.O. Box 500
Beaverton, OR 97077
USA

For product information, sales, service, and technical support:
- In North America, call 1-800-833-9200.
- Worldwide, visit www.tek.com to find contacts in your area.

Table of Contents

Preface iii
Getting Started 1-1
Setting Up Remote Communications Hardware 1-1
Ethernet 1-1
USB 1-2
Setting Up Remote Communications Software 1-3
Using VISA 1-3
Using the LXI Web Page and e*Scope 1-4
Using a Socket Server 1-5
Documentation 1-7
Dynamic programmatic interface 1-7
Implicit activation 1-7
Creating, deleting, and listing dynamic instances 1-8
Command Syntax 2-1
Command and Query Structure 2-1
Clearing the instrument output queue 2-3
Command Entry 2-3
Constructed Mnemonics 2-5
Argument Types 2-7
Command groups 2-11
Acquisition command group 2-11
Act On Event command group 2-12
AFG Command Group 2-13
Alias command group 2-15
Bus command group 2-16
Calibration command group 2-24
Callouts command group 2-25
Cursor command group 2-26
Digital command group 2-33
Digital Power Management (DPM) Group 2-34
Display control command group 2-36
DVM Command Group 2-39
Ethernet Command Group 2-40
File system command group 2-41
Horizontal command group 2-43
Inverter Motors and Drive Analysis (IMDA) Group 2-44
Mask command group 2-47
Math command group 2-48
Measurement command group 2-50

Miscellaneous command group 2-66
Plot command group 2-69
Power command group 2-69
Save and Recall command group 2-84
Save On command Group 2-85
Search and Mark command group 2-86
Self Test command group 2-112
Spectrum View command group 2-113
Status and Error command group 2-115
Trigger command group 2-116
Vertical command group 2-127
Waveform Transfer command group 2-130
Zoom command group 2-134
Commands listed in alphabetical order 2-135
Status and Events 3-1
Registers 3-1
Queues 3-4
Event Handling Sequence 3-5
Synchronization Methods 3-7
Appendix A: Character Set A-1
Appendix B: Reserved Words B-1
Appendix C: Factory Defaults C-1
Default Setup C-1
Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples D-1
Example 1: Analog Waveform (Channels 1–4) D-1
Example 2: Digital Waveform D-3
Example 3: Digital with 4 Bytes Per Point and Zoom Off D-5
Example 4: Digital with 8 Bytes Per Point and Zoom Off D-7
Example 5: Digital with 4 Bytes Per Point and Zoom On D-10
Example 6: RF Frequency Domain Waveform D-12
Example 7: Baseband IQ Waveform D-14
Appendix E: Search and Trigger Command Sequence Examples E-1
Example 1: Single Threshold Edge Search E-1
Example 2: Single Threshold Edge Trigger E-1
Example 3: Dual Threshold Runt Search E-2
Example 4: Single Threshold Logic Search on Three Waveforms E-3
Glossary
Index

Preface

This programmer guide provides you with the information required to use Programmable Interface commands to remotely control your instrument.

The programmer manual is divided into the following major topics:

- **Getting started.** This topic introduces you to the programming information and provides basic information about setting up your instrument for remote control.
- **Command syntax.** This topic provides an overview of the command syntax that you use to communicate with the instrument and other general information about commands, such as how commands and queries are constructed, how to enter commands, constructed mnemonics, and argument types.
- **Command groups.** This topic contains all the commands listed in functional groups. Each group consists of an overview of the commands in that group and a table that lists all the commands and queries for that group. You can click a command in the listing to display a detailed description of the command.
- **Commands listed in alphabetical order.** This topic contains all commands listed in alphabetical order. Command details, syntax, and examples are provided.
- **Status and events.** This topic discusses the status and event reporting system for the programming interfaces. This system informs you of certain significant events that occur within the instrument. Topics that are discussed include registers, queues, event handling sequences, synchronization methods, and messages that the instrument might return, including error messages.
- **Appendices.** These topics contain miscellaneous information, such as a list of reserved words, a table of the factory initialization (default) settings, and interface specifications that can be helpful when using commands to remotely control the instrument.

Getting Started

This manual explains the use of commands to remotely control your instrument. With this information, you can write computer programs to perform functions, such as setting the front-panel controls, taking measurements, performing statistical calculations, and exporting data for use in other programs.

Familiarity with the User Manual for your instrument is assumed. You can download the User Manual from the Tektronix website at www.tek.com.

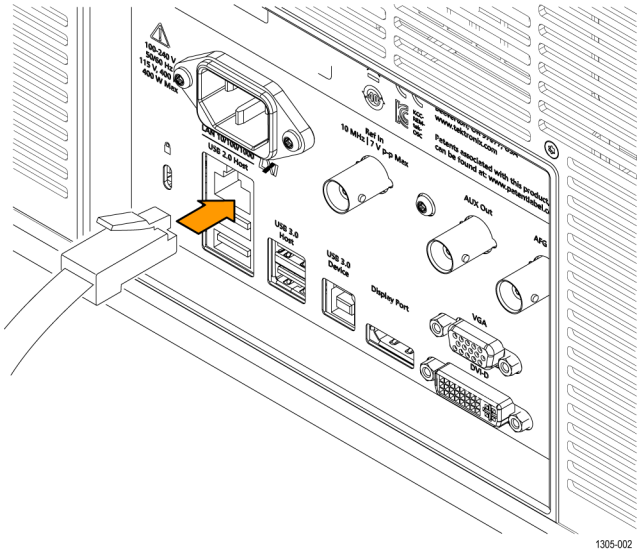
NOTE. *Most examples in this document assume that both HEADer and VERBose are set to ON.*

Setting Up Remote Communications Hardware

You can remotely control communications between your instrument and a PC via Ethernet or USB cables.

Ethernet

If you are using Ethernet, start by connecting an appropriate Ethernet cable to the Ethernet port (RJ-45 connector) on the rear panel of your instrument. This connects the instrument to a 10BASE-T/100BASE-T/1000BASE-T local area network.

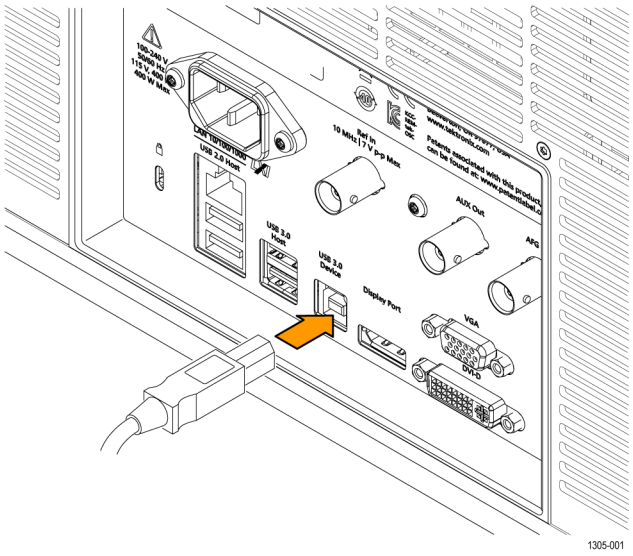


To change the Ethernet settings on your instrument, do the following:

1. Select the **Utility** drop-down menu.
2. Select the **I/O** menu.
3. Select the **LAN** panel.
4. In the menu, if you are on a DHCP Ethernet network that supplies the IP address automatically by a DHCP, tap **Auto**.
5. In the menu, if you want to supply your own network settings, tap **Manual** to set a hard coded TCP/IP address.

USB

If you are using USB, start by connecting the appropriate USB cable to the USB 3.0 super-speed (SS) Device port on the rear panel of your instrument. This port requires that the cable connected from the port to the host computer meets the USB 3.0 specification for super speed connections. Typically, such cables should be 3 feet or shorter in length, but this is determined by the quality of the cable and, with higher quality cables, this length can be extended. (It is also dependent upon the drive capability of the host USB port to which the instrument is connected.) The use of high quality short cables is recommended to avoid USB connection problems.



With USB, the system automatically configures itself. To verify that the USB is enabled:

1. Select the **Utility** drop-down menu.
2. Select the **I/O** menu.
3. Touch **USB Device Port** to open the USB Device Port configuration menu.
4. If USB is disabled, tap **USB Device Port** to enable the USB Device port.

After connection, the host, with appropriate software, can list the instrument as a USB device with the following parameters: (See Table 1-1.)

Table 1-1: USB Device Parameters

Parameter	Value
Manufacturer ID	0x0699 (decimal 1689)
Product ID	0x0522 (You can send the USBTMC:PRODUCTID:HEXadecimal? query to read the value)
Serial number	Serial number
Manufacturer description	"Tektronix"
Interface description	"USBTMC-USB488"

Setting Up Remote Communications Software

Connect your instrument directly to a computer to let the PC analyze your data, collect screen images, or to control the instrument using a program of your own creation. Three ways to connect your instrument to a computer are through the VISA drivers, the e*Scope Web-enabled tools, or via a socket server.

Using VISA

VISA lets you use your MS-Windows computer to acquire data from your instrument for use in an analysis package that runs on your PC, such as Microsoft Excel, National Instruments LabVIEW, Tektronix OpenChoice Desktop software, or your own custom software. You can use a common communications connection, such as USB or Ethernet, to connect the computer to the instrument.

To set up VISA communications between your instrument and a computer:

1. Load the VISA drivers on your computer. Also, load your application, such as OpenChoice Desktop. You will find the drivers and OpenChoice Desktop software on the appropriate CD that comes with your instrument or at the Tektronix software finder Web page (www.tektronix.com\downloads).
2. Connect the instrument to your computer with the appropriate USB or Ethernet cable. Cycle the power on the instrument.
3. Select the **Utility** drop-down menu
4. Select **I/O** menu.
5. If you are using USB, the system sets itself up automatically for you, if USB is enabled. Check **USB DEVICE PORT** panel to be sure that USB is enabled. If it is not enabled, toggle the On/Off button to On.
6. To use Ethernet, select the **LAN** panel. Use the controls to adjust your network settings, as needed. For more information, see the e*Scope setup information below.
7. If you want to change socket server parameters, select the **Socket Server** panel and enter new values through the resulting panel controls.
8. Run your application software on your computer.

Quick Tips

- The USB 3.0 super-speed (SS) device port is the correct USB port for computer connectivity. Use the rear- and front-panel USB 2.0 or 3.0 host ports to connect your instrument to USB flash drives, hard drives, keyboards or mice. Use the USB Device port to connect your instrument to a PC.

  There are both USB 2.0 and 3.0 host ports on the instrument. The device port is USB 3.0. Printers are not supported on host ports. PictBridge printers are not supported on the device port.

Using the LXI Web Page and e*Scope

With e*Scope, you can access your Internet-connected instrument from a web browser.

NOTE. *This procedure presumes that you have connected the instrument to a network and configures the instrument to work on your network. See the instrument Installation and Safety instructions, or the embedded Help on the instrument, for instructions on how to configure the instrument for network use.*

To set up e*Scope communications between your instrument and a web browser running on a remote computer:

1. Connect the instrument to your computer network with an appropriate Ethernet cable.
2. Select the **Utility** drop-down menu.
3. Select the **I/O** menu.
4. Select the **LAN** panel.
5. At the top left of the panel, there is an indicator light which turns red if the device detects a fault.
6. Tap **Test Connection** to check if your instrument can find an attached network.
7. Record the IP address shown on your instrument.
8. Start a Web browser on a PC that has access to the same network to which the instrument is attached.
9. Enter the IP address of the instrument in the browser address line. You should then see the LXI Welcome page on your Web browser on your computer screen.
10. For e*Scope, click the Instrument Control (e*Scope) link on the left side of the LXI Welcome page. You should then see a new tab (or window) open in your browser, showing the instrument screen. You can use the PC mouse to interact with the e*Scope screen and controls.

Using a Socket Server

A socket server provides two-way communication over an Internet Protocol-based computer network. You can use your instrument's socket server feature to let your instrument talk to a remote-terminal device or computer.

To set up and use a socket server between your instrument and a remote terminal or computer:

1. Connect the instrument to your computer network with an appropriate Ethernet cable.
2. Select the **Utility** drop-down menu.
3. Select the **I/O** menu.
4. Tap **Socket Server**.
5. On the resulting Socket Server panel, tap the top entry to toggle the Socket Server On.
6. Choose whether the protocol should be **None** or **Terminal**. A communication session run by a human at a keyboard typically uses a terminal protocol. An automated session might handle its own communications without using such a protocol.
7. If required, change the port number by rotating multipurpose knob **a.**
8. If required, press **OK** to set the new port number.

9. After setting up the socket server parameters, you can now have the computer "talk" to the instrument. If you are running an MS Windows PC, you could run its default client with its command-like interface. One way to do this is by typing " `Telnet` " in the Run window. The Telnet window will open on the PC.

NOTE. *On MS Windows 7, you must first enable Telnet in order for it to work.*

10. Start a terminal session between your computer and your instrument by typing in an open command with the instrument LAN address and port #. You can obtain the LAN address by pushing the **LAN** panel to view the resulting LAN setting panel. You can obtain the port # by tapping the **Socket Server** panel and viewing the **Port** item.

    For example, if the instrument IP address was `123.45.67.89` and the port # was the default of `4000`, you could open a session by writing into the MS Windows Telnet screen:

    ```
    open 123.45.67.89 4000
    ```

    The instrument will send a help screen to the computer when it is done connecting.

11. You can now type in a standard query, as found in the programmer manual, such as `*idn?`

    The Telnet session window will respond by displaying a character string describing your instrument. You can type in more queries and view more results on this Telnet session window. You can find the syntax for relevant queries and related status codes in other sections of this manual.

NOTE. *Do not use the computer's backspace key during an MS Windows' Telnet session with the instrument.*

Socket Server Terminal Protocol Mode Commands. Following are Tektronix Instrument Control Terminal Session Control commands:

!t <timeout> : set the response timeout in milliseconds.

!d : send device clear to the instrument.

!r : read response from instrument.

!h : print this usage info.

NOTE. *Commands containing a ? are treated as queries, and the responses are read automatically.*

Documentation Documentation for your instrument is available for download at *www.tek.com*.

Dynamic programmatic interface

This programmatic interface is dynamic. This means the instrument will not recognize certain commands until the objects referenced by those commands actually exist. For example, commands related to measurements are not recognized until measurements are added. Therefore, the response to a *LRN? query will not normally include the instrument's complete command set.

The following command groups are not available when the instrument is in its default state:

- Measurement
- Math
- Bus
- Search and Mark
- Plot

Adding an instance of one of those components will cause all commands related to that component to be recognized. For example, sending the MEASUrement:ADDNew command adds a measurement at which point the measurement commands will be recognized. Conversely, once all instances of a component have been deleted, the commands related to that component will no longer be recognized.

Implicit activation

When you send a command or query related to a dynamic object (such as Math1, Bus3, or Meas2) to the instrument and that instance does not yet exist, the instrument:

- creates a default instance with the name you specified
- adds all relevant commands and queries to the set of recognized commands
- responds to the command or query

Example 1 *RST followed by a *LRN? will not return any MATH<x> commands because in the default state, the instrument does not have any math waveforms. However querying :MATH:MATH1:DEFinition? will add MATH1 with the default math expression Ch1 - Ch2. Then the query will return the expected result. Note that if Ch1 or Ch2 is not active, they will be activated as part of this action. A *LRN issued after this will return MATH commands in addition to other available commands.

Example 2 *RST followed by :MEASUrement:MEAS3:TYPE? creates a measurement named MEAS3 and returns its type. Since the default type is Period, you will get Period as the response. A *LRN issued after this will return all MEASUrement:MEAS3 commands in addition to other available commands.

Not all commands in these groups start implicit activation. ADDNew, DELete, and LIST commands do not result in implicit activation.

Creating, deleting, and listing dynamic instances

You can create a new default instance of a dynamic feature by using the ADDNew command. For example, :MEASUrement:ADDNew "Meas1" will create a new measurement named Meas1. Meas1 will be a Period measurement since the default type for measurements is Period. Note that you can change Meas1 type to any other supported type using :MEASUrement:MEAS1:TYPE command. The DELete command deletes the named dynamic instance. For example :BUS:DELete “B2” will delete a bus named B2 if it exists. You can delete only one instance at a time. The LIST command returns a list of all dynamic instances currently in existence. For example, if you have added three measurements named Meas1, Meas2 and Meas3, :MEASUrement:LIST? returns MEAS1, MEAS2, MEAS3.

Command Syntax

You can control the operations and functions of the instrument through the Ethernet port or the USB 3.0 super speed (SS) device port using commands and queries. The related topics listed below describe the syntax of these commands and queries. The topics also describe the conventions that the instrument uses to process them. See the *Command Groups* topic in the table of contents for a listing of the commands by command group, or use the index to locate a specific command.

Backus-Naur Form Notation

This documentation describes the commands and queries using Backus-Naur Form (BNF) notation. Refer to the following table for the symbols that are used.

Table 2-1: Symbols for Backus-Naur Form

Symbol	Meaning
< >	Defined element
=	Is defined as
|	Exclusive OR
{ }	Group; one element is required
[ ]	Optional; can be omitted
. . .	Previous element(s) may be repeated

Command and Query Structure

Commands consist of set commands and query commands (usually called commands and queries). Commands modify instrument settings or tell the instrument to perform a specific action. Queries cause the instrument to return data and status information.

Most commands have both a set form and a query form. The query form of the command differs from the set form by its question mark at the end. For example, the set command `ACQuire:MODe` has a query form `ACQuire:MODe?`. Not all commands have both a set and a query form. Some commands have set only and some have query only.

Messages

A command message is a command or query name followed by any information the instrument needs to execute the command or query. Command messages may contain five element types, defined in the following table.

Table 2-2: Command Message Elements

Symbol	Meaning
<Header>	This is the basic command name. If the header ends with a question mark, the command is a query. The header may begin with a colon (:) character. If the command is concatenated with other commands, the beginning colon is required. Never use the beginning colon with command headers beginning with a star (*).
<Mnemonic>	This is a header subfunction. Some command headers have only one mnemonic. If a command header has multiple mnemonics, a colon (:) character always separates them from each other.
<Argument>	This is a quantity, quality, restriction, or limit associated with the header. Some commands have no arguments while others have multiple arguments. A <space> separates arguments from the header. A <comma> separates arguments from each other.
<Comma>	A single comma is used between arguments of multiple-argument commands. Optionally, there may be white space characters before and after the comma.
<Space>	A white space character is used between a command header and the related argument. Optionally, a white space may consist of multiple white space characters.

Commands

Commands cause the instrument to perform a specific function or change one of the settings. Commands have the structure:

```
[:]<Header>[<Space><Argument>[<Comma> <Argument>]...]
```

A command header consists of one or more mnemonics arranged in a hierarchical or tree structure. The first mnemonic is the base or root of the tree and each subsequent mnemonic is a level or branch off the previous one. Commands at a higher level in the tree may affect those at a lower level. The leading colon (:) always returns you to the base of the command tree.

Queries

Queries cause the instrument to return status or setting information. Queries have the structure:

- `[:]<Header>`
- `[:]<Header>[<Space><Argument> [<Comma><Argument>]...]`

You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level.

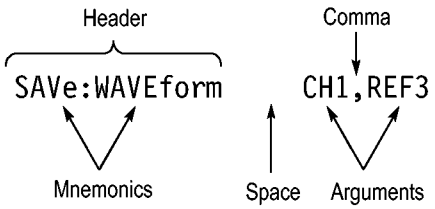
Headers

Use the `HEADer` command to control whether the instrument returns headers as part of the query response. If header is on, the query response returns command headers, then formats itself as a valid set command. When header is off, the response includes only the values. This may make it easier to parse and extract the information from the response. The table below shows the difference in responses.

Table 2-3: Comparison of Header Off and Header On Responses

Query	Header Off	Header On
TIME?	"14:30:00"	:TIME “14:30:00”
ACQuire:NUMAVg?	100	:ACQUIRE:NUMAVG 100

Command Parts



Clearing the instrument output queue

You can clear the Output Queue and reset the instrument a new command or query by using the selected Device Clear (DCL) function.

Command Entry

The following rules apply when entering commands:

- You can enter commands in upper or lower case.
- You can precede any command with white space characters. White space characters include any combination of the ASCII control characters 00 through 09 and 0B through 20 hexadecimal (0 through 9 and 11 through 32 decimal).
- The instrument ignores commands consisting of any combination of white space characters and line feeds.

Abbreviating

You can abbreviate many instrument commands. Each command in this documentation shows the minimum acceptable abbreviations in capitals. For example, you can enter the command `ACQuire:NUMAvg` simply as ACQ:NUMA or acq:numa.

Abbreviation rules may change over time as new instrument models are introduced. Thus, for the most robust code, use the full spelling.

If you use the `HEADer` command to have command headers included as part of query responses, you can further control whether the returned headers are abbreviated or are full-length with the `VERBose` command.

Concatenating

You can concatenate any combination of set commands and queries using a semicolon (;). The instrument executes concatenated commands in the order received.

When concatenating commands and queries, you must follow these rules:

1. Separate completely different headers by a semicolon and by the beginning colon on all commands except the first one. For example, the commands `TRIGger:MODe NORMal` and `ACQuire:NUMAVg 8`, can be concatenated into the following single command:

   ```
   TRIGger:MODe NORMal;:ACQuire:NUMAVg 8
   ```

2. If concatenated commands have headers that differ by only the last mnemonic, you can abbreviate the second command and eliminate the beginning colon. For example, you can concatenate the commands `ACQuire:MODe ENVelope` and `ACQuire:NUMAVg 8` into a single command:

   ```
   ACQuire:MODe ENVelope; NUMAVg 8
   ```

   The longer version works equally well:

   ```
   ACQuire:MODe ENVelope;:ACQuire:NUMAVg 8
   ```

3. Never precede a star (*) command with a colon:

   ```
   ACQuire:STATE 1;*OPC
   ```

   Any commands that follow will be processed as if the star command was not there so the commands, `ACQuire:MODe ENVelope;*OPC;NUMAVg 8` will set the acquisition mode to envelope and set the number of acquisitions for averaging to 8.

4. When you concatenate queries, the responses to all the queries are concatenated into a single response message. For example, if the display graticule is set to Full and the display style is set to dotsonly, the concatenated query `DISplay:GRAticule?;STYle?` will return the following.

If the header is on:

```
DISPLAY:GRATICULE FULL;:DISPLAY:STYLE DOTSONLY
```

If the header is off:

```
FULL;DOTSONLY
```

1. Set commands and queries may be concatenated in the same message. For example,

   ```
   ACQuire:MODe SAMple;NUMAVg?;STATE?
   ```

   is a valid message that sets the acquisition mode to sample. The message then queries the number of acquisitions for averaging and the acquisition state. Concatenated commands and queries are executed in the order received.

   Here are some invalid concatenations:

   `DISPlay:STYle DOTsonly OFF;ACQuire:NUMAVg 8` (no colon before `ACQuire`)

   `DISPlay:GRAticule FULL;:STYle DOTSONLY OFF` (extra colon before `STYle`.

   `DISPlay:GRAticule FULL;:*TRG` (colon before a star (*) command)

   `ACQUIRE:FASTACQ:PALETTE TEMPerature;FASTAcq:STATE ON` (levels of the mnemonics are different; either remove the second use of `FASTACQ:` or place `:ACQUIRE` in front of `FASTAcq:STATE`)

Terminating

This documentation uses <EOM> (End of Message) to represent a message terminator.

Table 2-4: End of Message Terminator

Symbol	Meaning
<EOM>	Message terminator

The end-of-message terminator must be the END message (EOI asserted concurrently with the last data byte). The last data byte may be an ASCII line feed (LF) character.

This instrument does not support ASCII LF only message termination. The instrument always terminates outgoing messages with LF and EOI.

Constructed Mnemonics

Some header mnemonics specify one of a range of mnemonics. For example, a channel mnemonic can be CH1, CH2, CH3, CH4, CH5, CH6, CH7, or CH8 depending on the number of FlexChannels in your instrument. You use these mnemonics in the command just as you do any other mnemonic. For example, there is a `CH1:POSition` command, and there is also a `CH2:POSition` command. In the command descriptions, this list of choices is abbreviated as CH<x>.

Bus Mnemonics

Commands specify the bus to use as a mnemonic in the header.

Table 2-5: Bus Mnemonics

Symbol	Meaning
B<x>	A bus specifier; <x> is ≥1.

Channel Mnemonics

Commands specify the channel to use as a mnemonic in the header.

Table 2-6: Channel Mnemonics

Symbol	Meaning
CH<x>	A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.
CH<x>_D<x>	A digital channel specifier; <x> in CH<x> is 1 through 8 and is limited by the number of FlexChannels in your instrument. <x> in D<x> is 0 through 7. Together they define a FlexChannel digital input.

Cursor Position Mnemonics

When cursors are displayed, commands may specify which cursor of the pair to use.

Table 2-7: Cursor Mnemonics

Symbol	Meaning
CURSOR<x>	A cursor selector; <x> is must be 1 or 2.

Math Specifier Mnemonics

Commands can specify the mathematical waveform to use as a mnemonic in the header.

Table 2-8: Math Specifier Mnemonics

Symbol	Meaning
MATH<x>	A math waveform specifier; <x> is ≥1.

Measurement Specifier Mnemonics

Commands can specify which measurement to set or query as a mnemonic in the header.

Table 2-9: Measurement Specifier Mnemonics

Symbol	Meaning
MEAS<x>	A measurement specifier; <x> is ≥1.

Reference Waveform Mnemonics

Commands can specify the reference waveform to use as a mnemonic in the header.

Table 2-10: Reference Waveform Mnemonics

Symbol	Meaning
REF<x>	A reference waveform specifier; <x> is ≥1.
REF<x>_D<x>	A digital reference waveform specifier; <x> is ≥1.

View Mnemonics

Commands can specify the view to use as a mnemonic in the header.

Table 2-11: Waveview Mnemonics

Symbol	Meaning
WAVEView<x>	A waveview specifier; <x> must be equal to 1.
PLOTView<x>	A plotview specifier; <x> must be equal to 1.
MATHFFTView<x>	A mathfftview specifier; <x> must be equal to 1.

Search Mnemonics

Commands can specify a search to use as a mnemonic in the header.

Table 2-12: Search Mnemonics

Symbol	Meaning
SEARCH<x>	A Search specifier; <x> is ≥1.

Zoom Mnemonics

Commands can specify a zoom to use as a mnemonic in the header.

Table 2-13: Zoom Mnemonics

Symbol	Meaning
ZOOM<x>	A zoom specifier; <x> must be equal to 1.

Argument Types

Commands use arguments such as enumeration, numeric, quoted string and block. Each of these arguments are listed in detail below.

Enumeration

Enter these arguments as unquoted text words. Like key words, enumeration arguments follow the same convention where the portion indicated in uppercase is required and that in lowercase is optional.

For example: `ACQuire:MODe SAMple`

Numeric

Many instrument commands require numeric arguments. The syntax shows the format that the instrument returns in response to a query. This is also the preferred format when sending the command to the instrument, though any of the formats will be accepted. This documentation represents these arguments as described below.

Table 2-14: Numeric Arguments

Symbol	Meaning
<NR1>	Signed integer value
<NR2>	Floating point value without an exponent
<NR3>	Floating point value with an exponent
<bin>	Signed or unsigned integer in binary format

Most numeric arguments will be automatically forced to a valid setting, by either rounding or truncating, when an invalid number is input, unless otherwise noted in the command description.

Quoted String

Some commands accept or return data in the form of a quoted string, which is simply a group of ASCII characters enclosed by a single quote (') or double quote ("). The following is an example of a quoted string: `"This is a quoted string"`. This documentation represents these arguments as follows:

Table 2-15: Quoted String Argument

Symbol	Meaning
<QString>	Quoted string of ASCII text

A quoted string can include any character defined in the 7-bit ASCII character set. Follow these rules when you use quoted strings:

1. Use the same type of quote character to open and close the string. For example: `"this is a valid string"`.
2. You can mix quotation marks within a string as long as you follow the previous rule. For example: `"this is an 'acceptable' string"`.
3. You can include a quote character within a string by repeating the quote. For example: `"here is a "" mark"`.
4. Strings can have upper or lower case characters.
5. A carriage return or line feed embedded in a quoted string does not terminate the string. The return is treated as another character in the string.
6. The maximum length of a quoted string returned from a query is 1000 characters.

Here are some invalid strings:

- `"Invalid string argument'` (quotes are not of the same type)
- `"test<EOI>"` (termination character is embedded in the string)

Block

Some commands use a block argument form to define a range or type of value, as defined in the table below.

Table 2-16: Block Argument

Symbol	Meaning
<NZDig>	A nonzero digit character in the range of 1–9
<Dig>	A digit character, in the range of 0–9
<DChar>	A character with the hexadecimal equivalent of 00 through FF (0 through 255 decimal)
<Block>	A block of data bytes defined as: <Block> ::= {#<NZDig><Dig>[<Dig>...][<DChar>...]|#0[<DChar>...]<terminator>}

<NZDig> specifies the number of <Dig> elements that follow. Taken together, the <NZDig> and <Dig> elements form a decimal integer that specifies how many <DChar> elements follow.

Command groups

The programmable interface conforms to Tektronix standard codes and formats except where noted. The interface also conforms to IEEE Std 488.2-1987 except where noted.

Acquisition command group

Acquisition commands set up the modes and functions that control how the instrument acquires signals and processes them into waveforms. Using these commands for acquiring waveforms, you can do the following:

- Start and stop acquisitions.
- Control whether each waveform is simply acquired, averaged, or enveloped over successive acquisitions of that waveform.
- Set the controls or conditions that start and stop acquisitions.
- Control acquisition of acquired channel waveforms.
- Set acquisition parameters.

Table 2-17: Acquisition commands

Command	Description
ACQuire?	Queries the current acquisition state.
ACQuire:FASTAcq:PALEtte	Sets or queries the waveform grading for fast acquisition mode.
ACQuire:FASTAcq:STATE	Sets or queries the state of fast acquisition mode.
ACQuire:MAXSamplerate?	Returns the maximum real-time sample rate.
ACQuire:MODe	Sets or queries the acquisition mode.
ACQuire:NUMACq?	Returns the number of waveform acquisitions that have occurred since starting acquisition with the `ACQuire:STATE RUN` command.
ACQuire:NUMAVg	Sets or queries number of acquisitions for an averaged waveform.
ACQuire:SEQuence:CURrent?	In single sequence acquisition mode, this query returns the number of acquisitions or measurements in the sequence completed so far.
ACQuire:SEQuence:MODe	In single sequence acquisition, the single sequence stop after count is based on number of acquisitions or measurements. Number of acquisitions is the only mode supported for this product.
ACQuire:SEQuence:NUMSEQuence	In single sequence acquisition mode, specify the number of acquisitions or measurements that comprise the sequence.
ACQuire:STATE	Starts, stops, or returns acquisition state.
ACQuire:STOPAfter	Sets or queries whether the acquisition is continuous or single sequence.

Act On Event command group

Use this group of commands to program the instrument to perform an action on trigger, search, measurement limit, and mask test events.

Table 2-18: Act On Event commands

Command	Description
ACTONEVent:ENable	Enables or disables actions on event (AOE).
ACTONEVent:LIMit	Sets whether the act on event should limit the number of saves, so as not to fill the hard drive.
ACTONEVent:LIMITCount	If the number of act on event saves is limited, this command sets the limit.
ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE	Save a screen capture when a mask test fails.
ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) when a mask test fails.
ACTONEVent:MASKFail:ACTION:SRQ:STATE	Generates an SRQ event when a mask fails.
ACTONEVent:MASKFail:ACTION:STOPACQ:STATE	Stops acquisitions when a mask test fails.
ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE	Saves a screen capture when a mask hit occurs.
ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) when a mask hit occurs.
ACTONEVent:MASKHit:ACTION:SRQ:STATE	Generates an SRQ event when a mask hit occurs.
ACTONEVent:MASKHit:ACTION:STOPACQ:STATE	Stops acquisitions when a mask hit occurs.
ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE	Saves a screen capture when a mask test passes.
ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) when a mask test passes.
ACTONEVent:MASKPass:ACTION:SRQ:STATE	Generates an SRQ event when a mask passes
ACTONEVent:MASKPass:ACTION:STOPACQ:STATE	Stops acquisitions when a mask test passes.
ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE	Saves a screen capture when the user-set measurement limit is exceeded.
ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) when the user-set measurement limit is exceeded.
ACTONEVent:MEASUrement:ACTION:SRQ:STATE	Generates an SRQ event when any measurement triggers the user-defined measurement limits.
ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE	Stops acquisitions when the user-set measurement limit is exceeded.
ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE	Saves a screen capture when a search event is found.

Table 2-18: Act On Event commands (cont.)

Command	Description
ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) when a search event is found.
ACTONEVent:SEARCH:ACTION:SRQ:STATE	Generates an SRQ event when any search event is found.
ACTONEVent:SEARCH:ACTION:STOPACQ:STATE	Stops acquisitions when a search event is found.
ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE	Saves a screen capture on a trigger event from a single sequence or sequence of N acquisition.
ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE	Saves the user set source waveform(s) on a trigger event from a single sequence or sequence of N acquisition.
ACTONEVent:TRIGger:ACTION:SRQ:STATE	Generates an SRQ event when a trigger event occurs.
ACTONEVent:TRIGger:ACTION:STOPACQ:STATE	Stops acquisitions on a trigger event from a single sequence or sequence of N acquisition.
SAVEONEVent:FILEDest	Sets or queries the file path.
SAVEONEVent:FILEName	Sets or queries the file name without the extension.
SAVEONEVent:IMAGe:FILEFormat	Sets or returns the file extension (png, jpg, bmp).
SAVEONEVent:WAVEform:FILEFormat	Sets or returns the file extension (csv, wfm, mat).
SAVEONEVent:WAVEform:SOUrce	Sets or returns the sources for saving waveforms when an event occurs.

AFG Command Group

Use the AFG commands for Arbitrary Function Generator functionality. Requires option AFG.

Table 2-19: AFG commands

Command	Description
AFG:AMPLitude	Sets (or queries) the AFG amplitude in volts, peak to peak.
AFG:ARBitrary:SOUrce	Sets or queries the source name for the Arbitrary Waveform.
AFG:FREQuency	Sets (or queries) the AFG frequency, in Hz.
AFG:FUNCtion	Sets (or queries) which AFG function to execute.
AFG:HIGHLevel	Sets (or queries) the high level value of the output waveform, in volts.
AFG:LOWLevel	Sets (or queries) the low level value of the output waveform, in volts.
AFG:NOISEAdd:PERCent	Sets (or queries) the AFG additive noise level as a percentage.
AFG:NOISEAdd:STATE	Sets (or queries) the AFG additive noise state.
AFG:OFFSet	Sets (or queries) the AFG offset value, in volts.
AFG:OUTPut:LOAd:IMPEDance	Sets (or queries) the AFG output load impedance.
AFG:OUTPut:STATE	Sets (or queries) the AFG output state.
AFG:PERIod	Sets (or queries) the period of the AFG waveform, in seconds.

Table 2-19: AFG commands (cont.)

Command	Description
AFG:PULse:WIDth	Sets (or queries) the AFG pulse width, in seconds.
AFG:RAMP:SYMmetry	Sets (or queries) the AFG ramp symmetry as a percentage.
AFG:SQUare:DUty	Sets (or queries) the AFG duty cycle, as a percentage.

Alias command group

Alias commands allow you to define new commands as a sequence of standard commands. You might find this useful when repeatedly using the same commands to perform certain tasks like setting up measurements.

Aliases are similar to macros but do not include the capability to substitute parameters into alias bodies. The alias mechanism obeys the following rules:

- The alias name must consist of a valid IEEE 488.2 message unit, which may not appear in a message preceded by a colon, comma, or a command or query program header.
- The alias name may not appear in a message followed by program date, a colon, comma, or question mark.
- An alias name must be distinct from any keyword or keyword short form.
- An alias name cannot be redefined without first being deleted using one of the alias deletion functions.
- Alias names do not appear in response messages.
- The Alias commands are defined in Tektronix Standard Codes and Formats. Deviations between that standard and what is specified here will be considered errors unless specifically noted in the command description in this document.

Table 2-20: Alias commands

Command	Description
ALIas	Sets or queries the alias state.
ALIas:CATalog?	Returns a list of the currently defined alias labels.
ALIas:DEFine	Assigns a sequence of program messages to an alias label.
ALIas:DELEte	Removes a specified alias.
ALIas:DELEte:ALL	Deletes all existing aliases.
ALIas:DELEte:NAMe	Removes a specified alias.
ALIas:STATE	Sets or queries the alias state.

Bus command group

Use the commands in the Bus Command Group to configure a bus. These commands let you:

- Specify the bus type.
- Specify the signals to be used in the bus.
- Specify its display style.

NOTE. *Bus commands are present once a bus has been added.*

Bus Mnemonics

Commands specify the bus to use as a mnemonic in the header.

Table 2-21: Bus mnemonics

Symbol	Meaning
B<x>	A bus specifier; <x> is ≥1.

Table 2-22: Bus commands

Command	Description
BUS:ADDNew	Adds the specified bus.
BUS:B<x>:ARINC429A:SOUrce	Sets or queries the source for the specified ARINC429 bus.
BUS:B<x>:ARINC429A:THRESHold	Sets or queries the ARINC429 upper threshold for the specified bus.
BUS:B<x>:ARINC429A:POLARITY	Sets or queries the source polarity for the specified ARINC429 bus.
BUS:B<x>:ARINC429A:DATAFORmat	Sets or queries the format of the DATA field for the specified ARINC429 bus.
BUS:B<x>:ARINC429A:BITRate	Sets or queries the ARINC429 bit rate for the specified bus.
BUS:B<x>:ARINC429A:BITRate:CUSTom	Sets or queries the ARINC429 custom bit rate for the specified bus.
BUS:B<x>:AUDio:BITDelay	Sets or queries the number of delay bits for the specified AUDIO bus.
BUS:B<x>:AUDio:BITOrder	Specifies the bit order for the specified AUDIO bus.
BUS:B<x>:AUDio:CLOCk:POLarity	Sets or queries the clock polarity for the specified AUDIO bus.
BUS:B<x>:AUDio:CLOCk:SOUrce	Sets or queries the clock source waveform for the specified AUDIO bus.
BUS:B<x>:AUDio:CLOCk:THReshold	Sets or queries the Audio Clock source threshold for the specified bus.
BUS:B<x>:AUDio:DATa:POLarity	Specifies the data polarity for the specified AUDIO bus.
BUS:B<x>:AUDio:DATa:SIZe	Specifies the number of bits per word for the specified AUDIO bus.
BUS:B<x>:AUDio:DATa:SOUrce	Specifies the audio data source waveform for the specified AUDIO bus.
BUS:B<x>:AUDio:DATa:THReshold	Sets or queries the Audio Data source threshold for the specified bus.
BUS:B<x>:AUDio:DATa:WORDSize	Sets or queries the Audio bits per word for the specified bus.
BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL	Sets or queries the Audio bits of sync width for the specified bus.
BUS:B<x>:AUDio:FRAME:SIZe	Specifies the number of channels in each frame for the specified AUDIO bus.
BUS:B<x>:AUDio:TYPe	Specifies the audio format (type) for the specified AUDIO bus.

Table 2-22: Bus commands (cont.)

Command	Description
BUS:B<x>:AUDio:WORDSel:POLarity	Sets or queries the word select polarity for the specified AUDIO bus.
BUS:B<x>:AUDio:WORDSel:SOUrce	Specifies the word select source waveform for the AUDIO bus.
BUS:B<x>:AUDio:WORDSel:THReshold	Sets or queries the Audio Word Select source threshold for the specified bus.
BUS:B<x>:AUTOETHERnet:DATAMINUSTHRESHOLD	Sets or queries the AutoEthernet D– source threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:DATAPLUSTHRESHold	Sets or queries the AutoEthernet D+ source threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:LOWDATAMINus	Sets or queries the AutoEthernet D– source low threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:LOWDATAPLUS	Sets or queries the AutoEthernet D+ source low threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:LOWTHRESHold	Sets or queries the AutoEthernet Data source Low threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:SIGNALTYpe	Sets or queries the AutoEthernet signal type for the specified bus.
BUS:B<x>:AUTOETHERnet:SOUrce	Sets or queries the AutoEthernet data (SDATA) source for the specified bus.
BUS:B<x>:AUTOETHERnet:SOUrce:DMINus	Sets or queries the AutoEthernet D– source for the specified bus.
BUS:B<x>:AUTOETHERnet:SOUrce:DPLUs	Sets or queries the AutoEthernet D+ source for the specified bus.
BUS:B<x>:AUTOETHERnet:THRESHold	Sets or queries the AutoEthernet DATA source High threshold level for the specified bus.
BUS:B<x>:AUTOETHERnet:TYPe	Sets or queries the AutoEthernet standard speed.
BUS:B<x>:CAN:BITRate	Sets or queries the CAN bit rate.
BUS:B<x>:CAN:BITRate:VALue	Sets or queries CAN bit rate.
BUS:B<x>:CAN:FD:BITRate	Sets or queries the increased data phase bit rate used by CAN FD packets on the specified CAN bus.
BUS:B<x>:CAN:FD:BITRate:CUSTom	Sets or queries the custom bit rate for the increased data phase of CAN FD packets on the specified CAN bus.
BUS:B<x>:CAN:SAMPLEpoint	Sets or queries the sample point for the specified CAN bus.
BUS:B<x>:CAN:SIGNal	Sets or queries the signal type for the specified CAN bus.
BUS:B<x>:CAN:SOUrce	Sets or queries the CAN source channel.
BUS:B<x>:CAN:STANDard	Sets or queries which CAN standard specification to analyze the specified CAN bus with.
BUS:B<x>:CAN:THReshold	Sets or queries the source channel threshold for the specified CAN bus.
BUS:B<x>:DISplay:FORMat	Sets or queries how the data is represented in the busform for the specified bus.
BUS:B<x>:DISplay:LAYout	This command sets or queries the format a bus layer should use.
BUS:B<x>:DPHY:CLOCk:SOUrce	Sets or queries the DPHY Clock source for the specified bus line.
BUS:B<x>:DPHY:CLOCk:THRESHold	Sets or queries the DPHY Clock source threshold for the specified bus.
BUS:B<x>:DPHY:DMINus:DATATHRESHold	Sets or queries the DPHY D- source data threshold for the specified bus.
BUS:B<x>:DPHY:DMINus:LPTHRESHold	Sets or queries the DPHY D- source low power threshold for the specified bus.
BUS:B<x>:DPHY:DMINus:SOUrce	Sets or queries the DPHY D- source for the specified bus line.

Table 2-22: Bus commands (cont.)

Command	Description
BUS:B<x>:DPHY:DPlus:DATATHRESHold	Sets or queries the D+ source data threshold for the specified bus.
BUS:B<x>:DPHY:DPlus:LPTHRESHold	Sets or queries the DPHY D+ source low power threshold for the specified bus.
BUS:B<x>:DPHY:DPlus:SOUrce	Sets or queries the DPHY D+ source for the specified bus line.
BUS:B<x>:DPHY:LP:DIRection	Sets or queries the DPHY bus direction in low power.
BUS:B<x>:DPHY:PROTocol:TYPe	Sets or queries the protocol type for DPHY bus decode.
BUS:B<x>:DPHY:SIGNal:ENCoding	Sets or queries the 8b9b encoding for DPHY bus decode.
BUS:B<x>:ETHERnet:DATAMINUSTHRESHold	Sets or queries the Ethernet DATA Minus source threshold for the specified bus.
BUS:B<x>:ETHERnet:DATAPLUSTHRESHold	Sets or queries the Ethernet DATA Plus source threshold for the specified bus.
BUS:B<x>:ETHERnet:IPVFOUR	Sets or queries whether IPV4 packets are available for triggering on Ethernet.
BUS:B<x>:ETHERnet:LOWTHRESHold	Sets or queries the Ethernet DATA source Low threshold for the specified bus.
BUS:B<x>:ETHERnet:QTAGGING	Sets or queries whether Q-Tagging packets are available for triggering on Ethernet.
BUS:B<x>:ETHERnet:SIGNALTYpe	Sets or queries the Ethernet signal type for the specified bus.
BUS:B<x>:ETHERnet:SOUrce	Specifies the Ethernet data source for differential input.
BUS:B<x>:ETHERNET:SOUrce:DMINus	Sets or queries the Ethernet DMINus source.
BUS:B<x>:ETHERNET:SOUrce:DPLUs	Sets or queries the Ethernet DPLUs source.
BUS:B<x>:ETHERnet:THRESHold	Sets or queries the Ethernet DATA source High threshold for the specified bus.
BUS:B<x>:ETHERnet:TYPe	Specifies the Ethernet standard type: 10Base-T or 100Base-T.
BUS:B<x>:EUSB:BITRate	Sets or queries the eUSB data rate for the specified bus.
BUS:B<x>:EUSB:DATAMINUS:DATA:THRESHold	Sets or queries the eUSB D- Input Source Data Threshold for Data line decode for specified bus.
BUS:B<x>:EUSB:DATAMINUSTHRESHold	Sets or queries the eUSB DATA Minus source threshold for the specified bus.
BUS:B<x>:EUSB:DATAPLUS:DATA:THRESHold	Sets or queries the eUSB D+ Input Source Data Threshold for Data line decode for specified bus.
BUS:B<x>:EUSB:DATAPLUSTHRESHold	Sets or queries the eUSB DATA Plus source threshold for the specified bus.
BUS:B<x>:EUSB:LOWTHRESHold	Sets or queries the eUSB source Low threshold for the specified bus when signal type is Differential.
BUS:B<x>:EUSB:OPERating:MODe	Sets or queries the eUSB mode for the specified bus.
BUS:B<x>:EUSB:SIGNALTYpe	Sets or queries the eUSB signal type for the specified bus.
BUS:B<x>:EUSB:SOUrce:DIFF	Sets or queries the eUSB Diff source for the specified bus when signal type is Diff.
BUS:B<x>:EUSB:SOUrce:DMINus	Sets or queries the eUSB DataMinus (SDATAMINUS) source for the specified bus.
BUS:B<x>:EUSB:SOUrce:DPLUs	Sets or queries the eUSB dataPlus (SDATAPLUS) source for the Specified bus.
BUS:B<x>:EUSB:THRESHold	Sets or queries the eUSB source High threshold for the specified bus when signal type is Diff.
BUS:B<x>:FLEXray:BITRate	Sets or queries the FlexRay bus bit rate.
BUS:B<x>:FLEXray:BITRate:CUSTom	Sets or queries the FlexRay custom bit rate for the specified bus.

Table 2-22: Bus commands (cont.)

Command	Description
BUS:B<x>:FLEXray:CHannel	Sets or queries the FlexRay bus channel.
BUS:B<x>:FLEXray:LOWTHRESHold	Sets or queries the FlexRay data source low threshold for the specified bus.
BUS:B<x>:FLEXray:SIGnal	Sets or queries the FlexRay probe.
BUS:B<x>:FLEXray:SOUrce	Sets or queries the FlexRay bus source.
BUS:B<x>:FLEXray:SOUrce:TXRX	Sets or queries the FlexRay TxRx data source for the specified bus.
BUS:B<x>:FLEXray:THRESHold	Sets or queries the FlexRay data source threshold for the specified bus.
BUS:B<x>:FLEXray:TXRXTHRESHold	Sets or queries the FlexRay data source TxRx threshold for the specified bus.
BUS:B<x>:I2C:CLOCk:SOUrce	Sets or queries the I2C clock (SCLK) source for the specified bus.
BUS:B<x>:I2C:CLOCk:THReshold	Sets or queries the I2C Clock (SCLK) source threshold for the specified bus.
BUS:B<x>:I2C:DATa:SOUrce	Sets or queries the I2C data (SDA) source for the specified bus.
BUS:B<x>:I2C:DATa:THReshold	Sets or queries the I2C Data (SDA) source threshold for the specified bus.
BUS:B<x>:I2C:RWINADDR	Determines whether decoded I2C slave addresses are pure seven-bit values, or have the R/W* combined with them.
BUS:B<x>:I3C:CLOCk:SOUrce	Sets or queries the I3C clock (SCLK) source for the specified bus.
BUS:B<x>:I3C:CLOCk:THReshold	Sets or queries the I3C clock (SCLK) source threshold level for the specified bus.
BUS:B<x>:I3C:DATa:SOUrce	Sets or queries the I3C data clock (SDA) source for the specified bus.
BUS:B<x>:I3C:DATa:THReshold	Sets or queries the I3C clock (SDA) data threshold level for the specified bus.
BUS:B<x>:LABel:COLor	Sets or queries the color of the specified bus label.
BUS:B<x>:LABel:FONT:BOLD	Sets or queries the bold state of the specified bus label.
BUS:B<x>:LABel:FONT:ITALic	Sets or queries the italic state of the specified bus label.
BUS:B<x>:LABel:FONT:SIZE	Sets or queries the font size of the specified bus label.
BUS:B<x>:LABel:FONT:TYPE	Sets or queries the font type of the specified bus label, such as Arial or Times New Roman.
BUS:B<x>:LABel:FONT:UNDERline	Sets or queries the underline state of the specified bus label.
BUS:B<x>:LABel:name	Sets or queries the waveform label for the specified bus.
BUS:B<x>:LABel:XPOS	Sets or queries the x-position of the specified bus' label.
BUS:B<x>:LABel:YPOS	Sets or queries the y-position of the specified bus' label.
BUS:B<x>:LIN:BITRate	Sets or queries the LIN bus bit rate.
BUS:B<x>:LIN:IDFORmat	Sets or queries LIN bus id format.
BUS:B<x>:LIN:POLarity	Sets or queries the LIN bus polarity.
BUS:B<x>:LIN:SAMPLEpoint	Specifies the point to sample during each bit period, as a percent, for the specified LIN bus.
BUS:B<x>:LIN:SOUrce	Sets or queries sets the LIN bus source.
BUS:B<x>:LIN:SOUrce:THReshold	Sets or queries the LIN source threshold for the specified bus.
BUS:B<x>:LIN:STANDard	Sets or queries the LIN bus standard.
BUS:B<x>:LIN:BITRate:CUSTom	Sets or queries LIN custom bit rate for the specified bus.
BUS:B<x>:MANChester:BITORDer	Sets or queries the Manchester bus Bit Order.

Table 2-22: Bus commands (cont.)

Command	Description
BUS:B<x>:MANChester:BITRate	Sets or queries the Manchester bus bit rate.
BUS:B<x>:MANChester:DISplaymode	Sets or queries the Manchester bus Packet View.
BUS:B<x>:MANChester:HEADer:LENGth	Sets or queries the Manchester header length in bits.
BUS:B<x>:MANChester:IDLE:BITS	Sets or queries the Manchester idle bit size in bits.
BUS:B<x>:MANChester:parity	Sets or queries the Manchester bus Parity.
BUS:B<x>:MANChester:SOUrce	Sets or queries the Manchester source for the specified bus.
BUS:B<x>:MANChester:START:INDex	Sets or queries the Manchester start Index in bits.
BUS:B<x>:MANChester:SYNC:SIZe	Sets or queries the Manchester sync Bit Size in bits.
BUS:B<x>:MANChester:THReshold	Sets or queries the Manchester threshold for the specified bus.
BUS:B<x>:MANChester:TOLerance	Sets or queries the Tolerance bus parameter.
BUS:B<x>:MANChester:TRAiler:LENGth	Sets or queries the Manchester trailer length in bits.
BUS:B<x>:MANChester:TRANstion:ZERo	Sets or queries the Manchester bus for zero falling or rising.
BUS:B<x>:MANChester:WORD:COUNt	Sets or queries the Manchester word count in bits.
BUS:B<x>:MANChester:WORDSIZe	Sets or queries the Manchester word size in bits.
BUS:B<x>:MDIO:CLOCk:SOUrce	Sets or queries the MDIO Clock source for the specified bus.
BUS:B<x>:MDIO:CLOCk:THReshold	Sets or queries the MDIO Clock source threshold for the specified bus.
BUS:B<x>:MDIO:DATA:SOUrce	Sets or queries the MDIO Data source for the specified bus.
BUS:B<x>:MDIO:DATA:THReshold	Sets or queries the MDIO Data source threshold for the specified bus.
BUS:B<x>:MIL1553B:SOUrce	Sets or queries the source for the specified MIL-STD-1553 bus.
BUS:B<x>:MIL1553B:THRESHold	Sets or queries the MIL-STD-1553 upper threshold for the specified bus.
BUS:B<x>:MIL1553B:LOWTHRESHold	Sets or queries the MIL-STD-1553 lower threshold for the specified bus.
BUS:B<x>:MIL1553B:POLarity	Sets or queries the source polarity for the specified MIL-STD-1553 bus.
BUS:B<x>:MIL1553B:RESPonsetime:MINimum	Sets or queries the minimum response time to a valid command issued for the specified MIL-STD-1553 bus.
BUS:B<x>:MIL1553B:RESPonsetime:MAXimum	Sets or queries the maximum response time to a valid command issued for the specified MIL-STD-1553 bus.
BUS:B<x>:NRZ:BITOrder	Sets or queries the NRZ bit order for the specified bus.
BUS:B<x>:NRZ:BITRate	Sets or queries the NRZ bus bit rate.
BUS:B<x>:NRZ:POLarity	Sets or queries the NRZ source polarity for the specified bus.
BUS:B<x>:NRZ:SOUrce	Sets or queries the NRZ source for the specified bus.
BUS:B<x>:NRZ:SPMI:VERsion	Sets or queries the Version for the specified bus.
BUS:B<x>:NRZ:THReshold	Sets or queries the NRZ threshold for the specified bus.
BUS:B<x>:PARallel:ALLTHResholds	Sets or queries the threshold for all sources for the parallel bus.
BUS:B<x>:PARallel:ALLTHResholds:APPly	Sets all of the data source thresholds to the value of the allMRefs parameter for the parallel bus.
BUS:B<x>:PARallel:BIT<n>SOUrce	Sets or queries the specified bit source for specified parallel bus.
BUS:B<x>:PARallel:BIT<n>SOUrce:THReshold	Sets or queries the specified bit source threshold for the specified parallel bus.

Table 2-22: Bus commands (cont.)

Command	Description
BUS:B<x>:PARallel:CLOCk:EDGE	Determines which edges of its clock signal cause a clocked parallel bus to sample new states.
BUS:B<x>:PARallel:CLOCk:ISCLOCKED	Determines whether the bus operates in a clocked or asynchronous fashion.
BUS:B<x>:PARallel:CLOCkSOUrce	Sets or queries the Parallel clock source for the specified bus.
BUS:B<x>:PARallel:CLOCKSOUrce:THReshold	Sets or queries the bit source threshold for the parallel bus.
BUS:B<x>PSIFIVe:BITPERiod	Sets or queries the PSI5 Bit period bus parameter.
BUS:B<x>:PSIFIVe:BITRate	Sets or queries the PSI5 bitrate.
BUS:B<x>PSIFIVe:COMM:DIRection	Sets or queries the PSI5 bus communication direction.
BUS:B<x>:PSIFIVe:DATAA	Sets or queries the PSI5 frame mandatory data region A.
BUS:B<x>:PSIFIVe:DATAB	Sets or queries the PSI5 frame optional bits of data region B.
BUS:B<x>:PSIFIVe:DATAFORMat	Sets or queries the data format in PSI5 Frame2 packet.
BUS:B<x>:PSIFIVe:ECUSOURce	Sets or queries the ECU to sensor source channel for the specified Bus.
BUS:B<x>:PSIFIVe:FRAMECONTrol	Sets or queries the PSI5 frame optional control bits.
BUS:B<x>:PSIFIVe:MESSaging	Sets or queries the PSI5 optional messaging bits.
BUS:B<x>:PSIFIVe:SOUrce	Sets or queries serial channel on or off.
BUS:B<x>:PSIFIVe:STATus	Sets or queries the optional status bits of PSI5.
BUS:B<x>:PSIFIVe:SYNCMODe	Sets or queries the PSI5 Sync Mode.
BUS:B<x>:PSIFIVe:SYNCTHRESHold	Sets or queries the PSI5 threshold for the ECU To Sensor specified bus.
BUS:B<x>:PSIFIVe:THRESHold	Sets or queries the PSI5 threshold for the Sensor To ECU specified bus.
BUS:B<x>:RS232C:BITRate	Sets or queries the RS-232 bit rate for the specified bus.
BUS:B<x>:RS232C:BITRate:CUSTom	Sets or queries the RS232 custom bit rate for the specified bus.
BUS:B<x>:RS232C:DATABits	Sets or queries the number of RS-232 data bits for the specified bus.
BUS:B<x>:RS232C:DELIMiter	Sets or queries the RS-232 delimiting value for a packet on the specified bus.
BUS:B<x>:RS232C:DISplaymode	Sets or queries the display mode for the specified bus.
BUS:B<x>:RS232C:PARity	Sets or queries the RS-232 parity for the specified bus.
BUS:B<x>:RS232C:POLarity	Sets or queries the RS-232 polarity for the specified bus.
BUS:B<x>:RS232C:SOUrce	Sets or queries the RS-232 polarity for the specified bus.
BUS:B<x>:RS232C:SOUrce:THReshold	Sets or queries the RS232 source threshold for the specified bus.
BUS:B<x>:S8B10B:BITRate	Sets or queries the 8B10B bit rate for the specified bus.
BUS:B<x>:S8B10B:SOUrce	Sets or queries the 8B10B source for the specified bus. Specifies the source channel.
BUS:B<x>:S8B10B:THReshold	Sets or queries the 8B10B threshold for the specified bus.
BUS:B<x>:SDLC:BITRate	Sets or queries the bit rate for the specified SDLC bus.
BUS:B<x>:SDLC:DATA:SOUrce	Sets or queries the source for the specified bus.
BUS:B<x>:SDLC:DATA:THReshold	Sets or queries the SDLC data source threshold for the specified bus.
BUS:B<x>:SDLC:ENCoding	Sets or queries the SDLC Bus Encoding.
BUS:B<x>:SDLC:MODulo	Sets or queries the SDLC Bus Modulo.

Table 2-22: Bus commands (cont.)

Command	Description
BUS:B<x>:SENT:CHANWidth	Sets or queries SENT fast channel bit widths for the specified bus.
BUS:B<x>:SENT:NIBBLECount	Sets or queries SENT data nibbles for the specified bus.
BUS:B<x>:SENT:NUMCHANnel	Sets or queries SENT fast data channels for the specified bus.
BUS:B<x>:SENT:PAUSEPULSe	Sets or queries SENT pause pulse for the specified bus.
BUS:B<x>:SENT:POLARITY	Sets or queries SENT Idle State signal polarity for the specified bus.
BUS:B<x>:SENT:SLOW	Sets or queries the SENT slow channel configuration for the specified bus.
BUS:B<x>:SENT:SOUrce	Sets or queries the SENT DATA source for the specified bus.
BUS:B<x>:SENT:THRESHold	Sets or queries the SENT DATA source threshold for the specified bus.
BUS:B<x>:SENT:TICKTIME	Sets or queries the SENT bus Clock Tick parameter for the specified bus.
BUS:B<x>:SENT:TICKTOLerance	Sets or queries the SENT bus Tick Tolerance percent parameter for the specified bus.
BUS:B<x>:SPACEWIRe:BITRate	Sets or queries the SpaceWire bit rate.
BUS:B<x>:SPACEWIRe:DATa:SOUrce	Sets or queries the source of the SpaceWire Data signal for the specified bus.
BUS:B<x>:SPACEWIRe:DATa:THReshold	Sets or queries the threshold of the SpaceWire Data signal for the specified bus.
BUS:B<x>:SPACEWIRe:DECode:TYPe	Sets or queries the decode type for SpaceWire bus decode.
BUS:B<x>:SPACEWIRe:STRobe:SOUrce	Sets or queries the source of the SpaceWire Strobe signal for the specified bus.
BUS:B<x>:SPACEWIRe:STRobe:THReshold	Sets or queries the threshold level of the SpaceWire Strobe signal for the specified bus.
BUS:B<x>:SPACEWIRe:SYNC	Sets or queries sync for SpaceWire decoding.
BUS:B<x>:SPACEWIRe:SYNC:COUnt	Sets or queries the length of the data string in bytes to be used for an SpaceWire sync bytes.
BUS:B<x>:SPACEWIRe:SYNC:PATTern	Sets or queries sync pattern for SpaceWire decoding.
BUS:B<x>:SPACEWIRe:SYNC:VALUe	Sets or queries sync value for sync option data.
BUS:B<x>:SPI:BITOrder	Sets or queries the shift direction used to de-serialize data for the SPI mode of the bus.
BUS:B<x>:SPI:CLOCk:POLarity	Sets or queries the SPI clock (SCLK) polarity for the specified bus.
BUS:B<x>:SPI:CLOCk:SOUrce	Sets or queries the SPI clock (SCLK) source for the specified bus.
BUS:B<x>:SPI:CLOCk:THReshold	Sets or queries the SPI Clock (SCLK) source threshold for the specified bus.
BUS:B<x>:SPI:DATa:POLarity	Sets or queries the SPI data (DATA) polarity for the specified bus.
BUS:B<x>:SPI:DATa:SIZe	Sets or queries the number of bits per word for the specified bus.
BUS:B<x>:SPI:DATa:SOUrce	Sets or queries the SPI data (DATA) source for the specified bus.
BUS:B<x>:SPI:DATa:THReshold	Sets or queries the SPI Data (DATA) source threshold for the specified bus.
BUS:B<x>:SPI:FRAMING	Sets or queries the SPI bus framing.
BUS:B<x>:SPI:IDLETime	Sets or queries the SPI bus idle time.
BUS:B<x>:SPI:MISo:DATa:POLarity	Sets or queries the SPI MISo Data source polarity for the specified bus.
BUS:B<x>:SPI:MISo:INPut	Sets or queries the SPI MISo source for the specified bus.
BUS:B<x>:SPI:MISo:THReshold	Sets or queries the SPI MISo Data source threshold for the specified bus.
BUS:B<x>:SPI:MOSi:DATa:POLarity	Sets or queries the SPI MOSI source polarity for the specified bus.

Table 2-22: Bus commands (cont.)

Command	Description
BUS:B<x>:SPI:MOSi:INPut	Sets or queries the SPI MOSI source for the specified bus.
BUS:B<x>:SPI:MOSi:THReshold	Sets or queries the SPI MOSI source threshold for the specified bus.
BUS:B<x>:SPI:NUMBer:INputs	Sets or queries the number of inputs for the specified bus.
BUS:B<x>:SPI:SELect:POLarity	Sets or queries the SPI Slave Select (SS) polarity for the specified bus.
BUS:B<x>:SPI:SELect:SOUrce	Sets or queries the SPI Slave Select (SS) source for the specified bus.
BUS:B<x>:SPI:SELect:THReshold	Sets or queries the SPI Select (SS) source threshold for the specified bus.
BUS:B<x>:SPMI:SCLk:SOUrce	This command sets or queries the SPMI Clock (SCLK) source for the specified bus.
BUS:B<x>:SPMI:SCLk:THReshold	This command sets or queries the SPMI Clock (SCLK) source threshold for the specified bus.
BUS:B<x>:SPMI:SDATa:SOUrce	This command sets or queries the SPMI Data (SDATA) source for the specified bus.
BUS:B<x>:SPMI:SDATa:THReshold	This command sets or queries the SPMI Data (SDATA) source threshold for the specified bus.
BUS:B<x>:SVID:ALERT:SOUrce	Sets or queries alert source channel for the specified SVID bus.
BUS:B<x>:SVID:ALERT:THReshold	Sets or queries the alert threshold for the specified SVID bus.
BUS:B<x>:SVID:CLOCk:SOUrce	Sets or queries clock source channel for the specified SVID bus.
BUS:B<x>:SVID:CLOCk:THReshold	Sets or queries the clock threshold for the specified SVID bus.
BUS:B<x>:SVID:DATA:SOUrce	Sets or queries data source channel for the specified SVID bus.
BUS:B<x>:SVID:DATA:THReshold	Sets or queries the data threshold for the specified SVID bus.
BUS:B<x>:TYPe	Sets or queries the bus type specified.
BUS:B<x>:USB:BITRate	Sets or queries the USB bit rate for the specified bus.
BUS:B<x>:USB:SOUrce	Sets or queries the USB Data Source for the specified bus.
BUS:B<x>:USB:SOUrce:DMINus	Sets or queries the USB Data Source for D- input for the specified bus.
BUS:B<x>:USB:SOUrce:DPLUs	Sets or queries the USB Data Source for D+ input for the specified bus.
BUS:B<x>:USB:DATAMINUSTHRESHold	Sets or queries the USB DATA Minus source threshold for the specified bus.
BUS:B<x>:USB:DATAPLUSTHRESHold	Sets or queries the USB DATA Plus source threshold for the specified bus.
BUS:B<x>:USB:LOWTHRESHold	Sets or queries the USB DATA source Low threshold for the specified bus.
BUS:B<x>:USB:SIGNALTYpe	Sets or queries the USB signal type for the specified bus.
BUS:B<x>:USB:THRESHold	Sets or queries the USB DATA source High threshold for the specified bus.
BUS:DELete	Deletes the specified bus.
BUS:LIST?	Lists all currently defined bus.
BUSTABle:ADDNew	Adds the specified bus table.
BUSTABle:DELete	Deletes the specified bus table.
BUSTABle:LIST?	Lists all currently defined bus tables.
DISplay:WAVEView<x>:BUS:B<x>:STATE	Sets or queries the state of the specified bus in the specified Waveform View.
DISplay:WAVEView<x>:BUS:B<x>:VERTical:POSition	Sets or queries the vertical position of the specified bus in the specified Waveform View.

Calibration command group

The Calibration commands provide information about the current state of instrument calibration and allow you to initiate signal path calibration (SPC).

NOTE. *When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes, which can take several minutes. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.*

Table 2-23: Calibration commands

Command	Description
*CAL?	Instructs the instrument to perform signal path calibration and returns the calibration status when complete. Takes several minutes to run.
CALibrate?	Returns the calibration status.
CALibrate:INTERNal	Starts the signal path calibration. Takes several minutes to run.
CALibrate:INTERNal:STARt	Starts the signal path calibration.
CALibrate:INTERNal:STATus?	Returns the status of the signal path calibration.
CALibrate:PWRUpstatus?	Returns the current status of the power-up calibration.
TOUCHSCReen:CALibrate	Launches the touchscreen calibration.
TOUCHSCReen:STATe	Sets or queries the enabled state of the touch screen.

Callouts command group

The Callout commands creates custom callouts to document specific details of your test results.

Table 2-24: Callout commands

Command	Description
CALLOUTS:CALLOUT<x>	Creates a new note.
CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE	Sets or queries the source of the Bookmark callout type.
CALLOUTS:CALLOUT<x>:BOOKMark:XPOS	Sets or queries the X-Position of the Bookmark callout type.
CALLOUTS:CALLOUT<x>:COLOR	Sets or queries the text color of the callout.
CALLOUTS:CALLOUT<x>:FONT:BOLD	Sets or queries the bold state of the callout text.
CALLOUTS:CALLOUT<x>:FONT:ITALIC	Sets or queries the italic state of the callout text.
CALLOUTS:CALLOUT<x>:FONT:SIZE	Sets or queries the font size of the callout text.
CALLOUTS:CALLOUT<x>:FONT:TYPE	Sets or queries the font type of the callout.
CALLOUTS:CALLOUT<x>:FONT:UNDERLine	Sets or queries the underline state of the callout text.
CALLOUTS:CALLOUT<x>:TEXT	Sets or queries the callout text.
CALLOUTS:CALLOUT<x>:TYPE	Sets or queries type of the callout

Cursor command group

Use the commands in the Cursor Command Group to control the cursor display and readout. You can use these commands to control the setups for each cursor, such as waveform source, and cursor position.

You can also use the commands to select one of the following cursor functions:

- **Off.** Shuts off the display of all cursors.
- **Vertical bars.** Displays vertical bar cursors, which provide traditional horizontal unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), the delta between them, and 1/delta (results in frequency when the horizontal unit is time). Vertical bars are another name for vertical screen cursors.
- **Horizontal bars.** Displays horizontal bar cursors, which provide traditional vertical unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), and the delta between them. Horizontal bars are another name for horizontal screen cursors.
- **Waveform cursors.** Consists of two cursors you can independently assign to a waveform. Waveform cursors enable you to conveniently measure waveform amplitude and time at specific points on the waveform. In XY or XYZ format, waveform cursors indicate the amplitude position of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.
- **Screen cursors.** Consist of two pairs of horizontal and vertical bar cursors. You can use these cursors to indicate an arbitrary position within the waveform display area. Screen cursors are basically just turning on horizontal bars and vertical bars at the same time. These cursors have no association with any waveform, except that they inherit the color of the waveform they are assigned to.

NOTE. *Cursor commands are available once a view has been added.*

Table 2-25: Cursor commands

Command	Description
DISplay:MATHFFTView<x>:CURSor:ASOUrce?	Queries the cursor source for cursor A in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:BSOUrce?	Queries the cursor source for cursor B in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:DDT?	Queries the delta V over delta T cursor readout value in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:FUNCtion	Sets or queries the cursor typein the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition	Sets or returns the horizontal cursor A position in the specified Math-FFT waveform.

Table 2-25: Cursor commands (cont.)

Command	Description
DISplay:MATHFFTView<x>:CURSor:HBArs:AUNIts?	Queries cursor A vertical units of the specified cursor in the specified view.
DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition	Sets or returns the vertical cursor B position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:HBArs:BUNIts?	Queries the cursor B vertical units in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:HBArs:DELTa?	Queries the delta V cursor readout value of the specified cursor in the specified view.
DISplay:MATHFFTView<x>:CURSor:MODe	Sets or returns the cursor tracking mode in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTATVALUE?	Queries the one over delta T cursor readout value in the specified Math-FFT waveform.
DISplay:MATHFFTView<n>:CURSor:ROLOCATION	Sets or queries the location to display the specified Math FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition	Sets or returns the horizontal cursor A position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOSition	Sets or returns the vertical cursor A position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOSition	Sets or returns the horizontal cursor B position of the specified cursor in the specified view.
DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOSition	Sets or returns the vertical cursor B position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:STATE	Sets or queries the visible state of cursors in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:VBArs:APOSition	Sets or queries the horizontal cursor A position in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:VBArs:AUNIts?	Queries the vertical cursor A measurement units for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:VBArs:BPOSition	Sets or queries the horizontal cursor B positionin the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:VBArs:BUNIts?	Queries the vertical cursor B measurement units for the specified Math-FFT view.
DISplay:MATHFFTView<x>:CURSor:VBArs:DELTa?	Queries the delta T cursor readout value of the specified cursor in the specified Math-FFT waveform.
DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition	Sets or queries the waveform cursor A position in the specified plot view.
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition	Sets or queries the waveform cursor B position in the specified plot view.
DISplay:PLOTView<x>:CURSor:ASOUrce?	Queries the cursor source for plot cursor A.
DISplay:PLOTView<x>:CURSor:BSOUrce?	Queries the cursor source for plot cursor B.
DISplay:PLOTView<x>:CURSor:DDT?	Queries the delta V over delta T cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:FUNCtion	Sets or queries the cursor mode of the specified cursor in the specified view.

Table 2-25: Cursor commands (cont.)

Command	Description
DISplay:PLOTView<x>:CURSor:HBArs:APOSition	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArs:AUNIts?	Sets or queries the cursor A vertical units of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArs:BPOSition	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArs:BUNIts?	Sets or queries the cursor B vertical units of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:HBArs:DELTa?	Sets or queries the delta V cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:MODe	Sets or queries the cursor tracking mode of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE?	Sets or queries the one over delta T cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<n>:CURSor:ROLOCATION	Sets or queries the location to display the specified Plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:SPLITMODE	Sets or queries the cursor source mode in the specified view.
DISplay:PLOTView<x>:CURSor:STATE	Sets or queries the visible state of the cursor of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArs:APOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArs:BPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArs:DELTa?	Queries the delta T cursor readout value of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:VBArs:UNIts?	Queries the VBArs cursor readout units of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:WAVEform:APOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:PLOTView<x>:CURSor:WAVEform:BPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:ASOUrce?	Queries the cursor source for plot cursor A

Table 2-25: Cursor commands (cont.)

Command	Description
DISplay:REFFFTView<x>:CURSor:BSOUrce?	Queries the cursor source for plot cursor B.
DISplay:REFFFTView<x>:CURSor:DDT?	Queries the delta V over delta T cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:FUNCtion	Sets or queries the cursor type of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:APOSition	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:AUNIts?	Queries cursor A vertical units of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:BPOSition	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:BUNIts?	Queries the cursor B vertical units of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:HBArs:DELTa?	Queries the delta V cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:MODe	Sets or queries the cursor tracking mode of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:ONEOVERDELTATVALUE?	Queries the one over delta T cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<n>:CURSor:ROLOCATION	Sets or queries the location to display the specified Reference FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).
DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition	Sets or queries the vertical cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SCREEN:BXPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SCREEN:BYPOSition	Sets or queries the vertical cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:SPLITMODE	Sets or queries whether both cursors have same or different source.
DISplay:REFFFTView<x>:CURSor:STATE	Sets or queries the visible state of the cursor of the specified cursor n the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:APOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:BPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:DELTa?	Queries the delta T cursor readout value of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:VBArs:UNIts?	Queries cursor A vertical units of the specified cursor in the specified view.
DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOSition?	Queries the value of the cursor A horizontal position.

Table 2-25: Cursor commands (cont.)

Command	Description
DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition	Sets or returns the waveform cursor A position in the specified plot view.
DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition?	Queries the value of the cursor A vertical position.
DISplay:REFFFTView<x>:CURSor:WAVEform:BHPOSition?	Queries the value of the cursor B horizontal position.
DISplay:REFFFTView<x>:CURSor:WAVEform:BPOSition	Sets or returns the waveform cursor B position in the specified plot view.
DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOSition?	Queries the value of the cursor B vertical position.
DISplay:SPECView<x>:CURSor:CURSOR:ANOISEDensity?	Returns the noise density of the selected RF trace at the Cursor A position
DISplay:SPECView<x>:CURSor:CURSOR:ASOUrce	Sets or queries the Cursor A source in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:BNOISE?	Returns the noise density of the selected RF trace at the Cursor B position when in Absolute mode, or the phase noise when in Delta mode.
DISplay:SPECView<x>:CURSor:CURSOR:BSOUrce	Sets or queries the Cursor B source in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:HBArs:APOSition	Queries the horizontal bar Cursor A vertical position (where it crosses the spectrum signal) in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:HBArs:AUNIts?	Queries the horizontal bar Cursor A amplitude units in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:HBArs:BPOSition	Queries the horizontal bar Cursor B vertical position (where it crosses the spectrum signal) in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:HBArs:BUNIts?	Queries the horizontal bar Cursor B amplitude units in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:READout	Sets or queries the Cursor B readout type in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:SPLITMODE	Sets or queries whether both Spectrum View cursors have the same or different sources.
DISplay:SPECView<x>:CURSor:CURSOR:STATE	Sets or queries the visibility state of the cursors in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:VBArs:APOSition	Sets or queries the Cursor A vertical position in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:VBArs:BPOSition	Sets or queries the Cursor B horizontal position in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:VBArs:UNIts?	Queries the cursor horizontal units in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:WAVEform:APOSition	Sets or queries the Cursor A vertical position in the Spectrum View.
DISplay:SPECView<x>:CURSor:CURSOR:WAVEform:BPOSition	Sets or queries the Cursor B vertical position in the Spectrum View.

Table 2-25: Cursor commands (cont.)

Command	Description
DISplay:WAVEView<x>:CURSor?	Queries the cursor parameters for the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>?	Queries the cursor parameters for the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:ASOUrce	Sets or queries the cursor A source of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:BSOUrce	Sets or queries the cursor B source of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:DDT?	Returns the delta V over delta T cursor readout value of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCtion	Sets or queries the cursor mode (SCREEN or DATA) of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition	Sets or queries the vertical cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:AUNIts?	Queries the cursor A vertical units of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition	Sets or queries the vertical cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BUNIts?	Queries the cursor B vertical units of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:DELTa?	Sets or queries the delta V cursor readout value over the history period.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:MODe	Sets or queries the cursor tracking mode of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:ONEOVERDELTATVALUE?	Sets or queries the one over delta T cursor readout value of the specified cursor in the specified Waveform View.
DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION	Sets or queries the location to display the Waveform View cursor readouts (in the Waveform View graticule or in a badge in the Results Bar).
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AXPOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition	Sets or queries the vertical cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSition	Sets or queries the vertical cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE	Sets or queries the cursor source mode in the specified view.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE	Sets or queries the visible state of the cursor of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:APOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.

Table 2-25: Cursor commands (cont.)

Command	Description
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:BPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:DELTa?	Sets or queries the delta T cursor readout value of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:UNIts?	Queries cursor A vertical units of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:APOSition	Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:AVPOSition?	Queries the vertical waveform value at the cursor A position in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition	Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.
DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:BVPOSition?	Queries the vertical waveform value at the cursor B position in the specified Waveform View.

Digital command group

Use the commands in the Digital Command Group to acquire up to 64 digital signals and analyze them. Digital channels are only available when a digital probe is attached to the super channel.

Table 2-26: Digital commands

Command	Description
CH<x>_DALL:LABel:COLor	Sets or queries the color of the specified digital group label.
CH<x>_DALL:LABel:FONT:BOLD	Sets or queries the bold state of the specified digital group.
CH<x>_DALL:LABel:FONT:ITALic	Sets or queries the italic state of the specified digital group.
CH<x>_DALL:LABel:FONT:SIZE	Sets or queries the font size of the specified digital group.
CH<x>_DALL:LABel:FONT:TYPE	Sets or queries the font type of the specified digital group, such as Arial or Times New Roman.
CH<x>_DALL:LABel:FONT:UNDERline	Sets or queries the underline state of the specified digital group.
CH<x>_DALL:LABel:NAMe	Sets or queries the label of the specified digital group.
CH<x>_D<x>:LABel:COLor	Sets or queries the color of the label of the specified digital bit.
CH<x>_D<x>:LABel:FONT:BOLD	Sets or queries the bold state of the label of the specified digital bit.
CH<x>_D<x>:LABel:FONT:ITALic	Sets or queries the italic state of the label of the specified digital bit.
CH<x>_D<x>:LABel:FONT:SIZE	Sets or queries the font size of the label of the specified digital bit
CH<x>_D<x>:LABel:FONT:TYPE	Sets or queries the font type of the label of the specified digital bit,
CH<x>_D<x>:LABel:FONT:UNDERline	Dets or queries the underline state of the label of the specified digital bit.
CH<x>_D<x>:LABel:NAMe	Sets or queries the label of the specified digital bit.
REF<x>_DALL:LABel:COLor	Sets or queries the color of the specified digital group.
REF<x>_DALL:LABel:FONT:BOLD	Sets or queries the bold state of the specified digital group.
REF<x>_DALL:LABel:FONT:ITALic	Sets or queries the italic state of the specified digital group.
REF<x>_DALL:LABel:FONT:SIZE	Sets or queries the font size of the specified digital group.
REF<x>_DALL:LABel:FONT:TYPE	Sets or queries the font type of the specified digital group.
REF<x>_DALL:LABel:FONT:UNDERline	Sets or queries the underline state of the specified digital group.
REF<x>_DALL:LABel:NAMe	Sets or queries the label of the specified digital group.
REF<x>_DALL:LABel:XPOS	Sets or queries the x-position of the label of the specified digital group.
REF<x>_DALL:LABel:YPOS	Sets or queries the y-position of the label of the specified digital group.
REF<x>_D<x>:LABel:COLor	Sets or queries the color of the label of the specified digital channel.
REF<x>_D<x>:LABel:FONT:BOLD	Sets or queries the bold state of the label of the specified digital bit.
REF<x>_D<x>:LABel:FONT:ITALic	Sets or queries the italic state of the label of the specified digital bit.
REF<x>_D<x>:LABel:FONT:SIZE	Sets or queries the font size of the label of the specified digital bit.
REF<x>_D<x>:LABel:FONT:TYPE	Sets or queries the font type of the label of the specified digital bit.
REF<x>_D<x>:LABel:FONT:UNDERline	Sets or queries the underline state of the label of the specified digital bit.
REF<x>_D<x>:LABel:NAMe	Sets or queries the label of the specified digital bit.
REF<x>_D<x>:LABel:XPOS	Sets or queries the x-position of the label of the specified digital bit.

Table 2-26: Digital commands (cont.)

Command	Description
REF<x>_D<x>:LABel:YPOS	Sets or queries the y-position of the label of the specified digital channel.
DIGGRP<x>:D<x>:THReshold	Sets or queries the threshold level in volts for specified digital channel.

Digital Power Management (DPM) Group

Use the commands in the DPM command group for Digital Power Management functionality. Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument).

Table 2-27: DPM Command Group

Command	Description
MEASU:MEAS1:SUBGROUP:RESULts:ALLAcqs:MAXimum?	Returns the maximum value of the measurement specified by the string, for all acquisitions.
MEASU:MEAS1:SUBGROUP:RESULts:ALLAcqs:MEAN?	Returns the mean value of the measurement specified by the string, for all acquisitions.
MEASU:MEAS1:SUBGROUP:RESULts:ALLAcqs:MINimum?	Returns the minimum value of the measurement specified by the string, for all acquisitions.
MEASU:MEAS1:SUBGROUP:RESULts:ALLAcqs:PK2PK?	Returns the peak-to-peak value of the measurement specified by the string, for all acquisitions.
MEASU:MEAS1:SUBGROUP:RESULts:ALLAcqs:POPUlation?	Returns the population value of the measurement specified by the string, for all acquisitions.
MEASU:MEAS1:SUBGROUP:RESULts:ALLAcqs:STDDev?	Returns the standard deviation value of the measurement specified by the string, for all acquisitions.
MEASU:MEAS1:SUBGROUP:RESULts:CURRentacq:MAXimum?	Returns the maximum value of the measurement specified by the string, for the current acquisition.
MEASU:MEAS1:SUBGROUP:RESULts:CURRentacq:MEAN?	Returns the mean value of the measurement specified by the string, for the current acquisition.
MEASU:MEAS1:SUBGROUP:RESULts:CURRentacq:MINimum?	Returns the minimum value of the measurement specified by the string, for the current acquisition.
MEASU:MEAS1:SUBGROUP:RESULts:CURRentacq:PK2PK?	Returns the peak-to-peak value of the measurement specified by the string, for the current acquisition.
MEASU:MEAS1:SUBGROUP:RESULts:CURRentacq:POPUlation?	Returns the population value of the measurement specified by the string, for the current acquisition.
MEASU:MEAS1:SUBGROUP:RESULts:CURRentacq:STDDev?	Returns the standard deviation value of the measurement specified by the string, for the current acquisition.
MEASUrement:AUTOset DPMAutoset	This command performs a power rail autoset operation.
MEASUrement:AUTOset DPMPReset	This command performs a power rail preset operation.
MEASUrement:MEAS<x>:EDGEQUALifier	Sets or queries the edge qualifier source.
MEASUrement:MEAS<x>:INPUTLEVel	Sets or queries the input trigger voltage level.
MEASUrement:MEAS<x>:MAXVoltage	Sets or queries the maximum input voltage.

Table 2-27: DPM Command Group (cont.)

Command	Description
MEASUrement:MEAS<x>:OUTPUT<n>VOLTage	Sets or queries the voltage level for the specified output voltage source n.
MEASUrement:MEAS<x>:REFVOLTAGE<n>Val	Sets or queries the reference voltage per rail of Overshoot and Undershoot measurements.
MEASUrement:MEAS<x>:RIPPLEFREQ<n>Val	Sets or queries the ripple frequency per specified rail for Ripple measurement.
MEASUrement:MEAS<x>:WAITTime	Sets or queries the wait time.
PLOT:PLOT<x>:RAILNUM	Sets the DPM histogram source.

Display control command group

Display commands can be found in this section as well as the sections of related components. These commands control general instrument settings, such as the intensity of the graticule, stacked or overlay display mode, and the fastacq color palette. Display commands also control how and where waveforms are shown, their position on screen, and zoom settings applied to the view. For example, display commands can turn on or off the display of channels or set the selected source.

Some actions can create a new view which can have its own settings. For example, adding a histogram will create a new view where the Histogram plot is displayed. Each view acts as a separate window within the instrument application and can be rearranged or annotated as desired.

A WaveView is the primary view used for viewing inputs and time-domain signals. Buses, non-FFT maths, refs, analog and digital channels, and time-trends are displayed in the WaveView.

A PlotView is used for viewing measurement results and other plotted data. Histograms, eye diagrams, XY/XYZ plots, FFTs, and other plots are shown in individual PlotViews.

Each of these views can have separate settings, zoom, cursors, and annotations. Display commands which are view specific have a view parameter in the programmable interface, such as DISPLAY:WAVEView<x>: ... or DISPLAY:PLOTView<x>: ... Selected source can also be specified on a per-view basis. The overall selected source is determined by the selected view and the selected source within that view.

PlotViews and WaveViews have some differences in command syntax due to differences in view functionality. For example, WaveViews and PlotViews have a different zoom model. The commands for specifying the zoom reflects these differences, and different command syntax is available depending on the view.

Table 2-28: Display control commands

Command	Description
DISplay?	Returns current display settings.
DISplay:{CH<x>|Math<x>|REF<x>}:INVERTColor	Sets or queries the Inverted mode color of the specified input source to the specified color.
DISplay:{CH<x>|Math<x>|REF<x>}:NORMALColor	Sets or queries the Inverted mode color of the specified input source to the specified color.
DISplay:COLors	Sets or queries the color mode for the graticule and waveform display.
DISplay:GLObal:B<x>:STATE	Sets or queries the display mode (on or off) of the specified bus.
DISplay:GLObal:CH<x>:STATE	Sets or queries the display mode (on or off) of the specified channel (both analog and digital).
DISplay:GLObal:MATH<x>:STATE	Sets or queries the display mode (on or off) of the specified math.
DISplay:GLObal:PLOT<x>:STATE	Sets or queries the display mode (on or off) of the specified time trend plot.

Table 2-28: Display control commands (cont.)

Command	Description
DISplay:GLObal:REF<x>:STATE	Sets or queries the display mode (on or off) of the specified reference.
DISplay:INTENSITy?	Returns the waveform and graticule saturation levels.
DISplay:INTENSITy:BACKLight	Sets or queries the waveform backlight intensity settings.
DISplay:INTENSITy:BACKLight:AUTODim:ENAble	Sets or queries the state of the display auto-dim feature.
DISplay:INTENSITy:BACKLight:AUTODim:TIMe	Sets or queries the amount of time, in minutes, to wait for no user interface activity before automatically dimming the display.
DISplay:MATHFFTView<x>:AUTOScale	Sets or returns the enabled state of autoscale for plots.
DISplay:MATHFFTView<x>:GRIDlines	Sets or queries the grid lines setting of the plot.
DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE	Sets or queries the state of the specified math waveform in the specified Waveform View.
DISplay:MATHFFTView<x>:XAXIS:SCALE	Sets or queries the x-axis scale setting for FFT Math waveforms.
DISplay:MATHFFTView<x>:YAXIS:SCALE	Sets or queries the vertical scale setting for FFT Maths.
DISplay:PERSistence	Sets or queries display persistence setting.
DISplay:PERSistence:RESET	Clears the persistence data.
DISplay:PLOTView<x>:AUTOScale	Sets or queries the enabled state of autoscale for plots.
DISplay:PLOTView<x>:GRIDlines	Sets or queries the Grid lines setting of the specified plot.
DISPLAY:PLOTVIEW<x>:XAXIS:SCALE	Sets or queries the horizontal scale setting for applicable plot.
DISPLAY:PLOTVIEW<x>:YAXIS:SCALE	Sets or queries the vertical scale setting for applicable plots.
DISplay:REFFFTView<x>:AUTOScale	Sets or queries the enabled state of auto-scale for plots.
DISplay:REFFFTView<x>:GRIDlines	Sets or returns the grid lines setting of the plot.
DISplay:REFFFTView<x>:REF:REF<x>:STATE	Sets or queries the state of the specified reference waveform in the specified Waveform View.
DISplay:REFFFTView<x>:XAXIS:SCALE	Sets or queries the x-axis scale setting for REFFFT.
DISplay:SELect:BUS	Sets or queries the overall selected bus.
DISplay:SELect:MATH	Sets or queries the overall selected math.
DISplay:SELect:REFerence	Sets or queries the overall selected reference waveform.
DISplay:SELect:SOUrce	Sets or queries the overall selected source.
DISplay:SELect:SPECView<x>:SOUrce	Sets or queries the spectrum trace (source channel) to select in the Spectrum View window.
DISplay:SPECView<x>:GRAticule	Sets or queries the type of graticule that is displayed.
DISplay:SPECView<x>:HORZ	Sets or queries the horizontal display scaling of the Spectrum View window.
DISplay:SPECView<x>:INTENSITy:GRATicule	Sets or queries the graticule saturation level
DISplay:SPECView<x>:INTENSITy:WAVEform	Sets or queries the waveform saturation level.
DISplay:SPECView<x>:VIEWStyle	Sets or queries the waveform layout style used by the display.
DISplay:SELect:VIEW	Sets or queries the selected view.

Table 2-28: Display control commands (cont.)

Command	Description
DISplay:SELect:WAVEView<x>:SOUrce	Sets or queries the selected source in the given waveview.
DISplay:VARpersist	Sets or queries the persistence decay time.
DISplay:WAVEform	Globally enables or disables the waveform display.
DISplay:WAVEView<x>:BUS:B<x>:STATE	Sets or queries the state of the specified bus in the specified Waveform View.
DISplay:WAVEView<x>:BUS:B<x>:VERTical:POSition	Sets or queries the vertical position of the specified bus in the specified Waveform View.
DISplay:WAVEView<x>:CH<x>:STATE	Sets or queries the state of the specified channel in the specified Waveform View.
DISplay:WAVEView<x>:CH<x>:VERTical:POSition	Sets or queries the vertical position f the specified channel in the specified Waveform View in divisions.
DISplay:WAVEView<x>:CH<x>:VERTical:SCAle	Sets or queries the vertical scale of the specified channel in volts per division within the specified Waveform View.
DISplay:WAVEView<x>:CH<x>_DALL:STATE	Sets or queries the display state of the specified digital channels in the specified Waveform View.
DISplay:WAVEView<x>:CH<x>_DALL:VERTical:POSition	Sets or queries the vertical position of the specified digital channel in the specified Waveform View in divisions.
DISplay:WAVEView<x>:CH<x>_D<x>:STATE	Sets or queries the display state of the specified digital channel in the specified Waveform View.
DISplay:WAVEView<x>:FILTer	Sets or queries the type of interpolation filter for the display.
DISplay:WAVEView<x>:GRAticule	Selects or queries the type of graticule that is displayed.
DISplay:WAVEView<x>:INTENSITy:GRATicule	Sets or queries the graticule saturation level.
DISplay:WAVEView<x>:INTENSITy:WAVEform	Sets or queries the waveform saturation level.
DISplay:WAVEView<x>:MATH:MATH<x>:AUTOScale	Sets or queries whether the specified math gets auto-scaled when the math equation changes within the specified Waveform View.
DISplay:WAVEView<x>:MATH:MATH<x>:STATE	Sets or queries the state of the specified math waveform in the specified Waveform View.
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition	Sets or queries the vertical position in divisions of the specified math waveform.
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle	Sets or queries the vertical scale of the specified math waveform.
DISplay:WAVEView<x>:PLOT:PLOT<x>:AUTOScale	Sets or queries whether the specified math gets auto-scaled when the math equation changes within the specified Waveform View.
DISplay:WAVEView<x>:PLOT:PLOT<x>:STATE	Sets or queries the state of the specified trend waveform in the specified Waveform View.
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSition	Sets or queries the vertical position of the specified time trend in the specified Waveform View in divisions.
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCAle	Sets or queries the vertical scale of the specified time trend in volts per division in the specified Waveform View.
DISplay:WAVEView<y>:REF<x>_DALL:FRAMe	Sets or returns the selected frame of the specified digital ref.

Table 2-28: Display control commands (cont.)

Command	Description
DISplay:WAVEView<y>:REF:REF<x>:FRAMe	sets or returns the selected frame of the specified analog ref.
DISplay:WAVEView<x>:REF:REF<x>:STATE	Sets or queries the state of the specified reference waveform in the specified Waveform View.
DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition	Sets or queries the vertical position in divisions of the specified reference waveform.
DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle	Sets or queries the vertical scale of the specified reference waveform.
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:POSition	Sets or queries the vertical position of the specified Frequency vs. Time trace in the Waveform View, in divisions.
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:SCAle	Sets or queries the vertical scale of the specified Frequency vs. Time trace in the Waveform View, in divisions.
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:POSition	Sets or queries the vertical position of the specified Magnitude vs. Time trace in the Waveform View, in divisions.
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:SCAle	Sets or queries the vertical scale of the specified Magnitude vs. Time trace in the specified Waveform View in Volts, Watts, or dB per division.
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:POSition	Sets or queries the vertical position of the specified Phase vs. Time trace in the Waveform View, in divisions.
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:SCAle	Sets or queries the vertical scale of the specified Phase vs. Time trace in the specified Waveform View in degrees per division.
DISplay:WAVEView<x>:STYle	Sets or queries the waveforms are displayed for analysis mode.
DISplay:WAVEView<x>:VIEWStyle	Sets or queries the waveform layout style used by the display.
DISplay:WAVEView:GRIDTYPE	Sets or queries the Waveform View Graticule mode (Fixed or Moveable).
DISPLAY:PLOTVIEW<x>:XAXIS:SCALE	Sets or queries the horizontal scale setting for applicable plots, either Linear or Log.
DISPLAY:PLOTVIEW<x>:YAXIS:SCALE	Sets or queries the vertical scale setting for applicable plots, either Linear or Log.

DVM Command Group

Use the commands in the DVM command group for Digital Voltmeter functionality. Requires DVM option (free with product registration).

Table 2-29: DVM Command Group

Command	Description
DVM	Resets the Digital Voltmeter measurements and history.
DVM:AUTORange	Sets or queries the autorange state for the Digital Voltmeter.
DVM:MEASUrement:FREQuency?	Returns the current frequency value for the Digital Voltmeter.
DVM:MEASUrement:HIStory:AVErage?	Returns the average readout value for the Digital Voltmeter function over the history period.
DVM:MEASUrement:HIStory:MAXimum?	Returns the maximum readout value for the DVM function over the history period.

Table 2-29: DVM Command Group (cont.)

Command	Description
DVM:MEASUrement:HIStory:MINImum?	Returns the minimum readout value for the DVM function over the history period.
DVM:MEASUrement:INFMAXimum?	Returns the maximum DVM readout value over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or DVM RESET.
DVM:MEASUrement:INFMINimum?	Returns the minimum readout value of the DVM function over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or DVM RESET.
DVM:MEASUrement:VALue?	Returns the DVM readout value.
DVM:MODe	Specifies or queries the mode to use for the Digital Voltmeter.
DVM:SOUrce	Sets or queries the source for the Digital Voltmeter.
DVM:TRIGger:FREQuency:COUNTer	Sets or queries the state of the trigger frequency counter.

Ethernet Command Group

Use the commands in the Ethernet Command Group to set up the 10BASE-T, 100BASE-TX, 1000BASE-TX or 100BASE-T Ethernet remote interface.

Table 2-30: Ethernet Commands

Command	Description
ETHERnet:DHCPbootp	Specifies the network initialization search for a DHCP/BOOTP server.
ETHERnet:DNS:IPADDress	Specifies the network Domain Name Server (DNS) IP address.
ETHERnet:DOMAINname	Specifies the network domain name.
ETHERnet:ENET:ADDress?	Returns the Ethernet address (MAC address) value assigned to the instrument.
ETHERnet:GATEWay:IPADDress	Specifies the network gateway IP address.
ETHERnet:IPADDress	Specifies the IP address assigned to the instrument.
ETHERnet:LXI:LAN:RESET	Resets the LXI local area network.
ETHERnet:LXI:LAN:SERVICENAMe	Specifies the mDNS service name used for the LXI interface.
ETHERnet:LXI:LAN:STATus?	Returns the LXI network status.
ETHERnet:NAME	Sets or queries the instrument Ethernet hostname assigned to the instrument.
ETHERnet:NETWORKCONFig	Specifies the Ethernet network configuration setting.
ETHERnet:PING	Causes the instrument to ping the gateway IP address.
ETHERnet:PING:STATus?	Returns the results from sending the ETHERnet:PING command to ping the gateway IP address.
ETHERnet:SUBNETMask	Specifies the network subnet mask value.

File system command group

Use the commands in the File System Command Group to help you use the built-in hard disk drive. You can use the commands to do the following:

- List the contents of the current directory
- Create and delete directories
- Create, copy, read, rename, or delete a file

When using these commands, keep the following points in mind:

- File arguments are always enclosed within double quotes: “C:\MYDIR\TEK00001.SET”
- File names follow the MS-DOS format: [DRIVE:][\PATH\]filename
- Path separators can be either forward slashes (/) or back slashes (\)

NOTE. *Using a back slash as a path separator can produce some unexpected results, depending on how your controller application treats escaped characters. Many applications recognize the sequence of a back slash followed by an alphabetic character as an escaped character, and, as such, interpret that alphabetic character as a control character. For example, the sequence “\n” might be interpreted as a newline character; “\t” might be interpreted as a tab character. To ensure that this interpretation does not occur, you can use double back slashes. For example, “C:\\testfile.txt”.*

- Some FILESystem commands can fail because a file has read-only attributes. You will not be able to delete or replace such files until this attribute is removed. Refer to the operating system help on file properties for further information.

Table 2-31: File system commands

Command	Description
FILESystem?	Returns the file system state.
FILESystem:COPy	Copies one or more files to a new file.
FILESystem:CWD	Sets or queries the current working directory for FILESystem commands.
FILESystem:DELEte	Deletes a named file or directory.
FILESystem:DIR?	Returns a list of directory contents.
FILESystem:HOMEDir?	Returns the current user's home directory.
FILESystem:LDIR?	Returns a list of directory contents.
FILESystem:MKDir	Makes a new directory.
FILESYSTEM:MOUNT:DRIVE	Mounts a network drive or queries if a network drive is currently mounted.
FILESystem:READFile	Copies the named file to the interface.

Table 2-31: File system commands (cont.)

Command	Description
FILESystem:REName	Assigns a new name to an existing file.
FILESystem:RMDir	Deletes the named directory.
FILESystem:UNMOUNT:DRIve	Unmount the USB drive.
FILESystem:WRITEFile	Copies the block data to a named file.

Horizontal command group

Horizontal commands control the time base of the instrument. You can set the time per division (or time per point) of the main time base. You can use the Horizontal commands to do the following:

- Set the scale, horizontal position and reference, and units of the time base
- Get the screen resolution, time of first point and time of last point, or get all the horizontal settings
- Enable or disable the display of the time base

Table 2-32: Horizontal commands

Command	Description
ACQuire:NUMFRAMESACQuired?	Returns the number of FastFrame frames which have been acquired.
CH<x>:SCALERATio	Sets or returns the scale ration for the specified analog channel.
HORizontal?	Queries the current horizontal settings.
HORizontal:ACQDURATION?	Returns the time base duration.
HORizontal:DELay:MODe	Sets or queries the horizontal delay mode.
HORizontal:DELay:TIMe	Sets or queries the horizontal delay time (position) that is used when delay is on.
HORizontal:DIVisions?	Returns the number of graticule divisions over which the waveform is displayed.
HORizontal:FASTframe?	Returns all information under horizontal:fastframe.
HORizontal:FASTframe:COUNt	Sets or returns the number of frames.
HORizontal:FASTframe:MAXFRames?	Returns the maximum number of frames.
HORizontal:FASTframe:MULtipleframes:MODe	Sets or returns the overlay display type.
HORizontal:FASTframe:REF:FRAme	Sets or returns the reference frame number.
HORizontal:FASTframe:REF:INCLUde	Sets or returns whether the reference frame delta information is shown in the display.
HORizontal:FASTframe:SELECTED	Sets or returns the selected frame number for acquired frames.
HORizontal:FASTframe:STATE	Sets or returns the state of FastFrame.
HORizontal:FASTframe:SUMFrame?	Sets or returns the summary frame type.
HORizontal:FASTframe:SUMFrame:STATE	Sets or returns the state of FastFrame summary frame.
HORizontal:FASTframe:TIMEStamp:REFerence?	Returns the time-stamp of the FastFrame Reference frame.
HORizontal:FASTframe:TIMEStamp:SELECTED?	Returns the time-stamp of the FastFrame Selected acquired frame.
HORizontal:FASTframe:TIMEStamp:DELTa?	Returns the time difference between the Selected and Reference time-stamps.
HORizontal:FASTframe:TIMEStamp:ALL?	Returns the time stamp of all frames.
HORizontal:FASTframe:XZEro:ALL?	Returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command for all frames.

Table 2-32: Horizontal commands (cont.)

Command	Description
HORizontal:FASTframe:XZEro:REF?	Returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command for the reference frame.
HORizontal:FASTframe:XZEro:SELECTED?	Returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command for the selected frame.
HORizontal:MAIn:INTERPRatio?	Returns the main horizontal time base interpolation ratio.
HORizontal:MODE	Sets or queries the horizontal mode.
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:VALue	Sets or queries the horizontal FastAcq maximum record length.
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:ZOOMOVERride	Sets or queries the flag which allows override of the horizontal FastAcq maximum record length.
HORizontal:MODe:MANual:CONFIGure	Sets or queries which horizontal control (scale or record length) will also change when the sample rate is adjusted.
HORizontal:MODE:RECOrdlength	Sets or queries the record length.
HORizontal:MODE:SAMPLERate	Sets or queries the sample rate.
HORizontal:MODE:SCAle	Sets or queries the horizontal scale.
HORizontal:POSition	Sets or queries the waveform horizontal position, in percent, that is used when delay is off.
HORizontal:PREViewstate?	Returns the display system preview state.
HORizontal:RECOrdlength	Sets or queries the horizontal record length.
HORizontal:ROLL?	Queries the horizontal roll mode status.
HORizontal:SAMPLERate	Sets or queries the horizontal sample rate.
HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide	Sets or queries the flag which allows override of the horizontal analyze minimum sample rate.
HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue	Sets or queries the minimum sample rate used by Analysis Automatic horizontal mode.
HORizontal:SCAle	Sets or queries the horizontal scale.

Inverter Motors and Drive Analysis (IMDA) Group

Use the commands in the IMDA group for input analysis, output analysis, ripple analysis measurements.

Table 2-33: IMDA Command Group

Command	Description
CUSTOMTABle:ADDNew	Adds a new custom results table.
CUSTOMTABle:DELete	Deletes the custom result(s) table..
CUSTOMTABle:LIST?	Queries the list of custom result tables have been added.

Table 2-33: IMDA Command Group (cont.)

Command	Description
MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MEAN?	Query the mean value of the measurement for current acquisition
MEASUrement:ADDMEAS	Adds a measurement.
MEASUrement:AUTOset	Sets or queries 3 phase autoset for the measurement.
MEASUrement:HIGHLEVel:CONNECTion	Sets or queries the connection type for the measurement.
MEASUrement:HIGHLEVel:L2LTOL2N	Enables the Line-to-Line to Line-to-Neutral Conversion.
MEASUrement:HIGHLEVel:WIRing	Sets or queries the wiring configuration for the measurement.
MEASUrement:MEAS<x>:CONFIGuration	Sets or queries the Input or Output Configuration of the measurement.
MEASUrement:MEAS<x>:CUSTOMLIMITSFile	Sets or queries the custom limits file path for custom harmonics standard.
MEASUrement:MEAS<x>:EDGEQUALifier	Sets or queries the Edge Qualifier source.
MEASUrement:MEAS<x>:EDGEQUALONE	Sets or queries the measurement 1 input edge qualifier one source for Efficiency measurement.
MEASUrement:MEAS<x>:EDGEQUALTWO	Sets or queries the measurement 1 output edge qualifier two source for Efficiency measurement.
MEASUrement:MEAS<x>:FILTERRANGEFROM	Sets or queries harmonics filter from order.
MEASUrement:MEAS<x>:FILTERRANGETO	Sets or queries harmonics filter to order.
MEASUrement:MEAS<x>:FREQ	Sets or queries switching frequency for Switching Ripple measurement.
MEASUrement:MEAS<x>:FUNDCURRent	Sets or queries the fundamental current value for IEC-Class C type harmonics standard.
MEASUrement:MEAS<x>:HARMONICSCLass	Sets or queries the class type for the harmonics measurement.
MEASUrement:MEAS<x>:HARMONICSSOURce	Sets or queries harmonics source.
MEASUrement:MEAS<x>:HARMONICSStd	Sets or queries harmonics standard.
MEASUrement:MEAS<x>:HARMONICSUNits	Sets or queries harmonics units.
MEASUrement:MEAS<x>:INFILTers:LOWPass:FREQ	Sets or queries the measurement input low pass filter frequency.
MEASUrement:MEAS<x>:INFILTers:LOWPass:SPEC	Sets or queries the measurement input low pass filter specification.
MEASUrement:MEAS<x>:INPUTPOwer	Sets or queries the input power value for IEC-Class C and Class D harmonics standard.
MEASUrement:MEAS<x>:LINESelected	Sets or queries selected lines for the measurement.
MEASUrement:MEAS<x>:ODDEVen	Sets or queries harmonics oddeven.
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:FREQ	Sets or queries the measurement output low pass filter frequency.
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:SPEC	Sets or queries the measurement output low pass filter specification.
MEASUrement:MEAS<x>:POWERFACtor	Sets or queries the power factor value for IEC-Class C type harmonics standard.

Table 2-33: IMDA Command Group (cont.)

Command	Description
MEASUrement:MEAS<x>:SIGNALFREQUEncy	Sets or queries line frequency for Harmonics and Line Ripple frequency.
MEASUrement:MEAS<x>:TYPe	Sets or queries the measurement type for the measurement.

Mask command group

Mask commands compare incoming waveforms to standard or user-defined masks. A mask is a set of polygonal regions on the screen. Unlike limit testing, the inside of a mask is the region where waveform data would not normally fall. Standards with eye patterns usually have three masks, but some have four.

Table 2-34: Mask commands

Command	Description
EYEMASK:MASK<x>:COUNt:HITS?	Returns the total number of hit violations for all segments in the specified eye diagram mask test.
EYEMASK:MASK<x>:COUNt:SEG<y>:HITS?	Returns the number of hit violations for the specified segment (area) in the specified eye diagram mask test.
EYEMASK:MASK<x>:CREATor?	Returns the name of the eye diagram plot that created the mask.
EYEMASK:MASK<x>:ENAbled	Enables or disables eye mask testing in the specified plot.
EYEMASK:MASK<x>:MASKfile	Sets or queries the current mask definition file name for the specified mask test.
EYEMASK:MASK<x>:TESt:SAMple:THReshold	Sets or queries the total number of hit violations that will cause a mask test failure.
EYEMASK:MASK<x>:TESt:STATUS?	Returns the mask hit test status for the specified mask test.
MASK:DELete	Deletes all mask segments of the specified mask test.
MASK:MASK<x>:COUNT?	Returns the total number of mask hits in all segments and the number of mask hits in each individual mask segment for the specified mask test.
MASK:MASK<x>:COUNT:HITS?	Returns the total number of mask hits in all mask segments for the specified mask test.
MASK:MASK<x>:DEFinedby	Sets or queries whether the specified mask is defined by segments or tolerances.
MASK:MASK<x>:DISplay	Sets or queries the display state of the mask used for the specified mask test.
MASK:MASK<x>:LIST?	Queries the list of segments in the mask used by the specified mask test.
MASK:MASK<x>:SEG<x>:POINTS	Sets or queries the X/Y coordinates of all points in the designated mask segment.
MASK:MASK<x>:SEG<x>COUNT:HITS?	Returns the total number of mask hits in the specified mask segment of the specified mask test.
MASK:MASK<x>:SOUrce	Sets or queries analog source for the specified mask test.
MASK:MASK<x>:TESt:STATE	Sets or queries state of the specified mask test.
MASK:MASK<x>:TESt:STATUS?	Queries the status of the specified mask test.
MASK:MASK<x>:TEST:THReshold	Sets or queries the number of waveform violations needed for the specified mask test to change from PASS to FAIL.
MASK:MASK<x>:TOLerance:HORizontal	Sets or queries the mask horizontal tolerance.
MASK:MASK<x>:TOLerance:UPDatenow	Causes the tolerance mask to be recalculated with the current horizontal and vertical tolerances.
MASK:MASK<x>:TOLerance:VERTical	Sets or queries the mask vertical tolerance.
MASK:TESt:WAVEforms	Sets or queries the number of waveform acquisitions to test during mask testing.
RECAll:MASK	Recalls a saved mask definition from a Mask File.

Math command group

Use the commands in the Math Command Group to create and define math waveforms. Use the available math functions to define your math waveform.

The math waveform you create depends on sources listed in the math expression. If you change these sources, the math waveforms you previously defined will be affected.

Math expressions can be simple, containing no mathematical computation, such as CH1, which specifies that a waveform shows the signal source of channel 1. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources, functions, and operands.

The acquisition of a live waveform can stop for several reasons: You can turn off the channel, stop the waveform (via Run/Stop from the Horiz/Acq menu), or stop the trigger (via Run/Stop from the Trig menu). When you turn off the channel, math continues and data is acquired but is not displayed. When you stop either the waveform or the trigger, the math calculation stops, and the last math calculation performed is displayed.

When a live waveform updates or a reference waveform is altered, math waveforms containing those waveforms as sources are also updated to reflect the changes. Also, sources must exist but do not need to be displayed to be used in and to update math waveforms.

NOTE. *Math commands are present once a math has been added.*

Table 2-35: Math commands

Command	Description
DISplay:GLObal:MATH<x>:STATE	Sets or queries the global state of the specified math.
DISplay:SELect:MATH	Sets or queries the overall selected math.
DISplay:WAVEView<x>:MATH:MATH<x>:STATE	Sets or queries the state of the specified math waveform in the specified Waveform View.
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition	Sets or queries the vertical position in divisions of the specified math waveform.
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle	Sets or queries the vertical scale of the specified math waveform.
MATH:ADDNew	Adds the specified math.
MATHArbflt<x>:FILepath	Sets the file path for a file of filter coefficients and reads the file.
MATH:DELete	Deletes the specified math.
MATH:LIST?	Lists all currently defined math waveforms.
MATH:MATH<x>:AVG:MODE	Sets or queries the math average mode flag.
MATH:MATH<x>:AVG:WEIGht	Sets or queries the number of acquisitions at which the averaging algorithm will begin exponential averaging.

Table 2-35: Math commands (cont.)

Command	Description
MATH:MATH<x>:FUNCtion	Sets or queries the basic math arithmetic function.
MATH:MATH<x>:LABel:COLor	Sets or queries color of the specified math's label.
MATH:MATH<x>:LABel:FONT:BOLD	Sets or queries bold state of the specified math label.
MATH:MATH<x>:LABel:FONT:ITALic	Sets or queries italic state of the specified math label.
MATH:MATH<x>:LABel:FONT:SIZE	Sets or queries font size of the specified math label.
MATH:MATH<x>:LABel:FONT:TYPE	Sets or queries font type of the specified math label.
MATH:MATH<x>:LABel:FONT:UNDERline	Sets or queries the underline state of the specified math label.
MATH:MATH<x>:LABel:XPOS	Sets or queries the X screen offset where the math waveform label is displayed.
MATH:MATH<x>:LABel:YPOS	Sets or queries the Y screen offset where the math waveform label is displayed.
MATH:MATH<x>:DEFine	Defines new waveforms using mathematical expressions.
MATH:MATH<x>:GATing	Specifies or returns the gating setting.
MATH:MATH<x>:LABel:NAMe	Sets or queries the label string.
MATH:MATH<x>:SOUrce<x>	Sets or queries the specified math source.
MATH:MATH<x>:SPECTral:HORZ	Sets or queries the horizontal display scale of the spectral math waveform.
MATH:MATH<x>:SPECTral:MAG	Sets or queries the units of the SpectralMag function in the specified math definition string.
MATH:MATH<x>:SPECTral:PHASE	Sets or queries the units of a SpectralPhase function in the specified math definition string.
MATH:MATH<x>:SPECTral:SOUrce	Sets or queries the specified spectral math source.
MATH:MATH<x>:SPECTral:SUPPress	Sets or queries whether suppression threshold for the specified math waveform is enabled.
MATH:MATH<x>:SPECTral:SUPPress:VALue	Sets or queries in volts the value of suppression threshold of the specified math waveform.
MATH:MATH<x>:SPECTral:TYPE	Sets or queries the FFT type selected for spectral analysis.
MATH:MATH<x>:SPECTral:UNWRap	Sets or queries whether phase unwrap of the spectral analyzer output data is enabled.
MATH:MATH<x>:SPECTral:UNWRap:DEGrees	Sets or queries in degrees the value of unwrap phase.
MATH:MATH<x>:SPECTral:WINdow	Sets or queries the window function used to multiply the spectral analyzer input data for the specified math waveform.
MATH:MATH<x>:TYPe	Sets or queries the math type.
MATH:MATH<x>:VUNIT	Sets or queries the math custom vertical units.

Measurement command group

Use the commands in the Measurement Command Group to control the automated measurement system.

Measurement commands can set and query measurement parameters. You can assign parameters, such as waveform sources and reference levels, differently for each measurement.

Clock recovery, edge, filter, gating, population and range measurement commands can be either global or per-measurement.
Global clock recovery commands are of the form `:MEASUrement:CLOCKRecovery:XXXX`
Global edge commands are of the form `:MEASUrement:XXXX`
Global filter commands are of the form `:MEASUrement:FILTers:XXXX`
Global gating commands are of the form `:MEASUrement:GATing:XXXX`
Global population commands are of the form `:MEASUrement:POPUlation:XXXX`
Global range commands are of the form `:MEASUrement:MEASRange:XXXX`

Reference levels for measurements can be global, per-measurement or per-source. The default is global. Per-measurement settings are used when local reference levels are enabled for the measurement with the command `:MEASUrement:MEAS?:GLOBalref 0`. Per-source settings are used when per-source settings are enabled with the command `:MEASUrement:REFLevels:TYPE PerSource` and per-source settings are selected for the measurement with the command `:MEASUrement:MEAS?:GLOBalref 0`.

Global reference level commands are of the form `:MEASUrement:REFLevels:XXXX`

Per-source reference level commands are of the form
`:MEASUrement:CH1:REFLevels:XXXX`
`:MEASUrement:MATH1:REFLevels:XXXX`

Table 2-36: Measurement commands

Command	Description
CUSTOMTABle:ADDNew	This command adds new custom results table.
CUSTOMTABle:DELete	This command deletes the custom result(s) table that was added.
CUSTOMTABle:LIST?	This command queries the list of custom result tables has been added.
MEASTABle:ADDNew	Adds a new measurement results view table to the scope application.
MEASTABle:DELETE	Removes the requested measurement results view table from the scope application.
MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MEAN?	This query returns the mean value of the measurement represented by the string, for the current acquisition.
MEASUrement?	This command returns all measurement parameters.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:ADDMEAS	This command adds a measurement.
MEASUrement:ADDNew	Adds the specified measurement.
MEASUrement:ANNOTate	This command sets or queries the annotation state for measurements.
MEASUrement:AUTOset	Performs an analysis jitter autoset.
MEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh	This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:FALLLow	This command sets or queries the value used as the low reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:FALLMid	This command sets or queries the value used as the mid reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:HYSTeresis	This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:RISEHigh	This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:RISELow	This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:RISEMid	This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:CH<x>:REFLevels:ABSolute:TYPE	This command sets or queries the reference level type for the source.
MEASUrement:CH<x>:REFLevels:BASETop	This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.
MEASUrement:CH<x>:REFLevels:METHod	This command sets or queries the method used to calculate reference levels for the source.
MEASUrement:CH<x>:REFLevels:PERCent:FALLHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:FALLLow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:FALLMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis	This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the source ref level method is set to percent.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:CH<x>:REFLevels:PERCent:RISELow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:RISEMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:CH<x>:REFLevels:PERCent:TYPE	This command specifies or queries the reference level percent type for the source.
MEASUrement:CLOCKRecovery:ADVanced:METHod	This command sets or queries the global advanced clock recovery method.
MEASUrement:CLOCKRecovery:CLOCKFrequency	This command sets or queries the global clock frequency used for fixed constant clock recovery.
MEASUrement:CLOCKRecovery:CLOCKMultiplier	This command sets or queries the global clock multiplier used for explicit clock recovery.
MEASUrement:CLOCKRecovery:CONSTCLOCKMODe	This command sets or queries the global constant clock mode used for constant clock recovery.
MEASUrement:CLOCKRecovery:DAMPing	This command sets or queries the global damping value used for PLL clock recovery.
MEASUrement:CLOCKRecovery:DATAPath	This command sets or queries the global file containing the data pattern used for known data pattern clock recovery.
MEASUrement:CLOCKRecovery:DATARate	This command sets or queries the global nominal data bit rate used for nominal data rate clock recovery.
MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODe	This command sets or queries the global explicit clock mode used for explicit clock recovery.
MEASUrement:CLOCKRecovery:JTFBandwidth	This command sets or queries the global JTF bandwidth used for PLL clock recovery.
MEASUrement:CLOCKRecovery:LOOPBandwidth	This command sets or queries the global loop bandwidth used for PLL clock recovery.
MEASUrement:CLOCKRecovery:MEANAUTOCalculate	This command sets or queries how often the clock is calculated for constant clock recovery.
MEASUrement:CLOCKRecovery:METHod	This command sets or queries the global clock recovery method.
MEASUrement:CLOCKRecovery:MODel	This command sets or queries the global PLL clock recovery model used for PLL clock recovery.
MEASUrement:CLOCKRecovery:NOMINALOFFset	This command sets or queries the global offset value used for explicit clock recovery.
MEASUrement:CLOCKRecovery:NOMINALOFFset:SELECTIONtype	This command sets or queries the global offset type used for explicit clock recovery.
MEASUrement:CLOCKRecovery:STAndard	This command sets or queries the global communications standard used for PLL clock recovery.
MEASUrement:DELete	The command deletes the specified measurement.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:DELETEALL	Deletes all the active instances of measurements defined in the scope application.
MEASUrement:DIRacmodel	This command sets or queries the dirac model used to separate random from deterministic jitter for jitter measurements.
MEASUrement:DISPLAYUnits	This command sets or queries the display units used for jitter summary measurements.
MEASUrement:EDGE<x>	Sets or queries the type of the edge for the measurement.
MEASUrement:EYERENDER	This command sets or queries the state of high-performance eye rendering for an eye diagram.
MEASUrement:FILTers:BLANKingtime	This command sets or queries the global filter blanking time.
MEASUrement:FILTers:HIGHPass:FREQ	This command sets or queries the global high pass filter frequency.
MEASUrement:FILTers:HIGHPass:SPEC	This command sets or queries the global high pass filter order.
MEASUrement:FILTers:LOWPass:FREQ	This command sets or queries the global low pass filter cutoff frequency.
MEASUrement:FILTers:LOWPass:SPEC	This command sets or queries the global low pass filter order.
MEASUrement:FILTers:RAMPtime	This command sets or queries the global filter ramp time.
MEASUrement:GATing	This command sets or queries the global gating type.
MEASUrement:GATing:ACTive	This command sets or queries the global gating active level used for logic gating.
MEASUrement:GATing:ENDtime	Sets or queries the end gate time for all measurements that use Global gating.
MEASUrement:GATing:HYSTeresis	This command sets or queries the global gating hysteresis value used for logic gating.
MEASUrement:GATing:LOGICSource	This command sets or queries the gating data source used for logic gating.
MEASUrement:GATing:MIDRef	This command sets or queries the global gating mid ref value used for logic gating.
MEASUrement:GATing:SEARCHSource	This command sets or queries the global gating search source used for logic gating.
MEASUrement:GATing:STARTtime	Sets or queries the start gate time for all measurements that use Global gating.
MEASUrement:HIGHLEVel:CONNECTion	Sets or queries the connection type for the measurement.
MEASUrement:HIGHLEVel:L2LTOL2N	Enables the Line-to-Line to Line-to-Neutral Conversion.
MEASUrement:HIGHLEVel:WIRing	Sets or queries the wiring configuration for the measurement.
MEASUrement:INTERp	This command sets or queries the interpolation mode used to locate edge crossings.
MEASUrement:JITTermodel	This command sets or queries the model used to separate random from deterministic jitter for jitter measurements.
MEASUrement:LIST?	Lists all currently defined measurements.
MEASUrement:LOCKRJ	Sets or queries the state of RJ locking.
MEASUrement:LOCKRJValue	Sets or queries the RJ lock value.
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh	This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow	This command sets or queries the value used as the low reference level of the falling edge when the source ref level method is set to absolute.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid	This command sets or queries the value used as the mid reference level of the falling edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis	This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh	This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow	This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid	This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute.
MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE	This command sets or queries the reference level type for the source.
MEASUrement:MATH<x>:REFLevels:BASETop	This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the source.
MEASUrement:MATH<x>:REFLevels:METHod	This command sets or queries the method used to calculate reference levels for the source.
MEASUrement:MATH<x>:REFLevels:PERCent:FALLHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:FALLLow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis	This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:RISEHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:RISELow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the source ref level method is set to percent.
MEASUrement:MATH<x>:REFLevels:PERCent:TYPE	This command specifies or queries the reference level percent type for the source.
MEASUrement:MEAS<x>:BER	This command sets or queries the BER value for the measurement.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:BER:TARGETBER	This command sets or queries the target BER value for the measurement.
MEASUrement:MEAS<x>:BIN	This command sets or queries the bin count for the measurement.
MEASUrement:MEAS<x>:BITCfgmode	This command sets or queries whether the measurement returns the mean or mode statistic result when the measurement type is bit amplitude/high/low.
MEASUrement:MEAS<x>:BITEnd	This command sets or queries the bit end as a percentage of the unit interval.
MEASUrement:MEAS<x>:BITPcnt	This command sets or queries the bit center as a percentage of the unit interval.
MEASUrement:MEAS<x>:BITSTart	This command sets or queries the bit start as a percentage of the unit interval.
MEASUrement:MEAS<x>:BITType	This command sets or queries the bit type for the measurement.
MEASUrement:MEAS<x>:BURSTEDGTYPe	This command sets or queries the burst edge type for the measurement.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MAXimum?	This query-only command returns the maximum cycle-cycle value for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MEAN?	This query-only command returns the mean cycle-cycle value for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MINimum?	This query-only command returns the minimum cycle-cycle value for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:PK2PK?	This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:POPUlation?	This query-only command returns the population of all cycle-cycle statistics for the specified measurement for all acquisitions accumulated since statistics were last reset.
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:STDDev?	This query-only command returns the standard deviation cycle-cycle for the specified measurement for all acquisitions.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MAXimum?	This query-only command returns the maximum cycle-cycle value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MEAN?	This query-only command returns the mean cycle-cycle value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MINimum?	This query-only command returns the minimum cycle-cycle value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:PK2PK?	This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:POPUlation?	This query-only command returns the population of the cycle-cycle statistics for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:STDDev?	This query-only command returns the standard deviation cycle-cycle for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:CLOCKRecovery:ADVanced:METHod	This command sets or queries the advanced clock recovery method when advanced clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKFrequency	This command sets or queries the clock frequency used when fixed constant clock recovery is used for the measurement.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier	This command sets or queries the clock multiplier used when explicit clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:CONSTCLOCKMODe	This command sets or queries the constant clock mode used when constant clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:DAMPing	This command sets or queries the damping value used when PLL clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath	This command sets or queries the file containing the data pattern used when known data pattern clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:DATARate	This command sets or queries the nominal data bit rate when nominal data rate clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODe	This command sets or queries the explicit clock mode used when explicit clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal	This command sets or queries the clock recovery settings global flag for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth	This command sets or queries the JTF bandwidth used when PLL clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth	This command sets or queries the loop bandwidth used when PLL clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate	This command sets or queries how often the clock is calculated when constant clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:METHod	This command sets or queries the clock recovery method for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:MODel	This command sets or queries the PLL clock recovery model used when PLL clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset	This command sets or queries the offset value used when explicit clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset:SELECTIONtype	This command sets or queries the offset type used when explicit clock recovery is used for the measurement.
MEASUrement:MEAS<x>:CLOCKRecovery:STAndard	This command sets or queries the communications standard when PLL clock recovery is used for the measurement.
MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE	This command sets or queries whether a filter is used for the measurement when the measurement type is AC common mode.
MEASUrement:MEAS<x>:COMMONMode:SOURCEs	This command sets or queries the number of sources for the measurement when the measurement type is AC common mode.
MEASUrement:MEAS<x>:CONFIGuration	Sets or queries the Input or Output Configuration of the measurement.
MEASUrement:MEAS<x>:CUSTOMLIMITSFile	Sets and queries the custom limits file path for custom harmonics standard.
MEASUrement:MEAS<x>:CYCLemode	This command sets or queries the cycle mode for the measurement.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:DELay:EDGE<x>	This command sets or queries the 'to edge' type when EDGE? is EDGE1 and the 'from edge' type when EDGE? is EDG2, for the measurement when the measurement type is DELAY.
MEASUrement:MEAS<x>:DISPlaystat:ENABle	Turns on and off the display of statistics in measurement badges.
MEASUrement:MEAS<x>:EDGE<x>	This command sets or queries the type of the specified edge, rise or fall, for the measurement.
MEASUrement:MEAS<x>:EDGEIncre	This command sets or queries the edge increment value for the measurement.
MEASUrement:MEAS<x>:EDGEQUALifier	Sets and queries the Edge Qualifier source.
MEASUrement:MEAS<x>:EDGEQUALONE	Set and queries the measurement 1 input edge qualifier one source for Efficiency measurement.
MEASUrement:MEAS<x>:EDGEQUALTWO	Sets and queries the measurement 1 output edge qualifier two source for Efficiency measurement.
MEASUrement:MEAS<x>:EDGES:FROMLevel	This command sets or queries the 'from level' edge for the measurement.
MEASUrement:MEAS<x>:EDGES:LEVel	This sets or queries the level type for the 'time outside level' measurement.
MEASUrement:MEAS<x>:EDGES:LOWERFREQuency	This command sets or queries the lower frequency for the measurement when the measurement type is phase noise. Lower frequencies are ignored.
MEASUrement:MEAS<x>:EDGES:N	The command sets or queries the number of accumulation cycles for the measurement when the measurement type is nperiod.
MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod	This command sets or queries the slew rate method for the measurement.
MEASUrement:MEAS<x>:EDGES:TOLevel	This command sets or queries the 'to level' edge for the measurement.
MEASUrement:MEAS<x>:EDGES:UPPERFREQuency	This command sets or queries the upper frequency for the measurement when the measurement type is phase noise. Higher frequencies are ignored.
MEASUrement:MEAS<x>:FAILCount?	Returns the number of measurement failures, if applicable, for the selected measurement.
MEASUrement:MEAS<x>:FILTERRANGEFROM	This command sets or queries harmonics filter from order.
MEASUrement:MEAS<x>:FILTERRANGETO	This command sets or queries harmonics filter to order.
MEASUrement:MEAS<x>:FILTers:BLANKingtime	This command sets or queries the filter blanking time for the measurement.
MEASUrement:MEAS<x>:FILTers:GLOBal	This command sets or queries the global flag for filter settings for the measurement.
MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ	This command sets or queries the high pass filter frequency for the measurement.
MEASUrement:MEAS<x>:FILTers:HIGHPass:SPEC	This command sets or queries the high pass filter order for the measurement.
MEASUrement:MEAS<x>:FILTers:LOWPass:FREQ	This command sets or queries the low pass filter cutoff frequency for the measurement.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC	This command sets or queries the low pass filter order for the measurement.
MEASUrement:MEAS<x>:FILTers:RAMPtime	This command sets or queries the filter ramp time for the measurement.
MEASUrement:MEAS<x>:FREQ	Sets or queries switching frequency for Switching Ripple measurement.
MEASUrement:MEAS<x>:FROMedge	This command sets or queries the 'from edge' type for the measurement.
MEASUrement:MEAS<x>:FROMEDGESEARCHDIRect	This command sets or queries the 'from edge' search direction for the measurement.
MEASUrement:MEAS<x>:FROMSymbol:LOGIC2SOUrce	This command sets or queries the DDR digital measurement logic 1 source From symbol.
MEASUrement:MEAS<x>:FROMSymbol:LOGIC3SOUrce	This command sets or queries the DDR digital measurement logic 2 source From symbol.
MEASUrement:MEAS<x>:FROMSymbol:LOGIC4SOUrce	This command sets or queries the DDR digital measurement logic 3 source From symbol.
MEASUrement:MEAS<x>:FROMSymbol:LOGIC5SOUrce	This command sets or queries the DDR digital measurement logic 4 source From symbol.
MEASUrement:MEAS<x>:FROMSymbol:MEASUREAT	This command sets or queries the DDR digital measurement From Symbol MeasureAT value.
MEASUrement:MEAS<x>:FUNDCURRent	Sets and queries the fundamental current value for IEC-Class C type harmonics standard.
MEASUrement:MEAS<x>:GATing	This command sets or queries the gating type for the measurement.
MEASUrement:MEAS<x>:GATing:ACTive	This command sets or queries the gating active level when the gating type is logic.
MEASUrement:MEAS<x>:GATing:ENDtime	Sets or queries the end gate time for the measurement when using Local gating.
MEASUrement:MEAS<x>:GATing:GLOBal	This command sets or queries the gating settings global flag.
MEASUrement:MEAS<x>:GATing:HYSTeresis	This command sets or queries the gating hysteresis value when the gating type is logic.
MEASUrement:MEAS<x>:GATing:LOGICSource	This command sets or queries the gating data source when the gating type is logic.
MEASUrement:MEAS<x>:GATing:MIDRef	This command sets or queries the gating mid ref value when the gating type is logic.
MEASUrement:MEAS<x>:GATing:SEARCHSource	This command sets or queries the gating search source when the gating type is search.
MEASUrement:MEAS<x>:GATing:STARTtime	Sets or queries the start gate time for the measurement when using Local gating.
MEASUrement:MEAS<x>:GLOBalref	This command sets or queries the reference levels global flag for the measurement.
MEASUrement:MEAS<x>:HARMONICSCLass	Sets and queries the class type for the harmonics measurement.
MEASUrement:MEAS<x>:HARMONICSSOURce	Sets or queries harmonics source.
MEASUrement:MEAS<x>:HARMONICSStd	Sets or queries harmonics standard.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:HARMONICSUNits	Sets or queries harmonics units.
MEASUrement:MEAS<x>:HIGHREFVoltage	This command sets or queries the high reference voltage value for the 'time outside level' measurement.
MEASUrement:MEAS<x>:IDLETime	This command sets or queries the idle time for the measurement when the measurement type is burst width.
MEASUrement:MEAS<x>:INFILTers:LOWPass:FREQ	Sets and queries the measurement input low pass filter frequency cutoff frequency on the scope.
MEASUrement:MEAS<x>:INFILTers:LOWPass:SPEC	Sets and queries the measurement input low pass filter specification filter order on the scope
MEASUrement:MEAS<x>:INPUTPOwer	Sets and queries the input power value for IEC-Class C and Class D harmonics standard.
MEASUrement:MEAS<x>:JITTERSummary:DCD	This command sets or queries whether DCD is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:DDJ	This command sets or queries whether DDJ is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:DJDD	This command sets or queries whether DJ-dd is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER	This command sets or queries whether EyeWidth@BER is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:NPJ	This command sets or queries whether NPJ is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:PJ	This command sets or queries whether PJ is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:RJDD	This command sets or queries whether RJ-dd is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:TIE	This command sets or queries whether TIE is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:JITTERSummary:TJBER	This command sets or queries whether TJ@BER is included in the jitter summary for the measurement.
MEASUrement:MEAS<x>:LABel	This command sets or queries the label for the measurement.
MEASUrement:MEAS<x>:LINESelected	Sets or queries selected lines for the measurement.
MEASUrement:MEAS<x>:LOWREFVoltage	This command sets or queries the low reference voltage value for the 'time outside level' measurement.
MEASurement:MEAS<x>:MAXCycle	Sets or queries the maximum cycle value for the DDRTERRN and DDRTERRMN measurements.
MEASUrement:MEAS<x>:MEASRange:GLOBal	This command sets or queries the range settings global flag for the measurement.
MEASUrement:MEAS<x>:MEASRange:MAX	This command sets or queries the range maximum value for the measurement.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:MEASRange:MIN	This command sets or queries the range minimum value for the measurement.
MEASUrement:MEAS<x>:MEASRange:STATE	This command sets or queries the range state for the measurement.
MEASurement:MEAS<x>:MINCycle	Sets or queries the minimum cycle value for the DDRTERRN and DDRTERRMN measurements.
MEASUrement:MEAS<x>:ODDEVen	Sets or queries harmonics oddeven.
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:FREQ	Sets and queries the measurement output low pass filter frequency cutoff frequency on the scope.
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:SPEC	Sets and queries the measurement output low pass filter specification filter order on the scope.
MEASUrement:MEAS<x>:PASSFAILENabled	Sets or returns the pass/fail test enable status.
MEASUrement:MEAS<x>:PASSFAILHIGHlimit	Sets or returns the high limit for a measurement test.
MEASUrement:MEAS<x>:PASSFAILLIMit	Sets or returns the limit for a measurement test.
MEASUrement:MEAS<x>:PASSFAILLOWlimit	Sets or returns the low limit for a measurement test.
MEASUrement:MEAS<x>:PASSFAILMARgin	Sets or returns the allowed margin for limit comparisons for all pass/fail checks
MEASUrement:MEAS<x>:PASSFAILWHEN	Sets or returns the condition on which a measurement test fails.
MEASUrement:MEAS<x>:PATTERNDETECTION	This command sets or queries the pattern detection type for the measurement.
MEASUrement:MEAS<x>:PATTERNLENgth	This command sets or queries the pattern length for the measurement.
MEASUrement:MEAS<x>:PATTERNTYPe	This command sets or queries the pattern type for the measurement.
MEASUrement:MEAS<x>:PERFREQ:EDGE	This command sets or queries the edge type of a Period/Frequency measurement.
MEASUrement:MEAS<x>:POLarity	This command sets or queries the polarity for the measurement when the measurement type is burst width.
MEASUrement:MEAS<x>:POPUlation:GLOBal	This command sets or queries the population settings global flag.
MEASUrement:MEAS<x>:POPUlation:LIMIT:STATE	This command sets or queries the population limit state for the measurement.
MEASUrement:MEAS<x>:POPUlation:LIMIT:VALue	This command sets or queries the population limit value for the measurement.
MEASUrement:MEAS<x>:POWERFACtor	Set and queries the power factor value for IEC-Class C type harmonics standard.
MEASUrement:MEAS<x>:REFLevels:ABSolute:FALLHigh	This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow	This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid	This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSTeresis	This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh	This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISELow	This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMid	This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:TYPE	This command sets or queries the reference level type for the measurement.
MEASUrement:MEAS<x>:REFLevels<x>:BASETop	This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement.
MEASUrement:MEAS<x>:REFLevels<x>:METHod	This command sets or queries the method used to calculate reference levels for the measurement.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLLow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis	This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid	This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE	This command specifies or queries the reference level percent type for the measurement.
MEASUrement:MEAS<x>:REFMode	This command sets or queries the reference level mode for the measurement.
MEASUrement:MEAS<x>:REFVoltage	This command sets or queries the reference voltage value for the measurement.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MAXimum?	This command is identical to that described in the DPOJet programmer manual.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MEAN?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MINimum?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:PK2PK?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:POPUlation?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:STDDev?	This command is identical to that described in the DPOJet programmer manual.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:MAXimum?	This query-only command returns the maximum value found for the specified measurement since the last statistical reset.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:MEAN?	This query-only command returns the mean value for the measurement for the current acquisition.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:MINimum?	This query-only command returns the minimum value found for the specified measurement since the last statistical reset.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:PK2PK?	This query-only command returns the peak-to-peak value for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:POPUlation?	This query-only command returns the population for the specified measurement for the current acquisition.
MEASUrement:MEAS<x>:RESUlts:CURRentacq:STDDev?	This query-only command returns the standard deviation for the specified measurement for all acquisitions accumulated since statistics were last reset.
MEASUrement:MEAS<x>:SIGNALFREQUEncy	Sets or queries line frequency for Harmonics and Line Ripple frequency.
MEASUrement:MEAS<x>:SIGNALType	This command sets or queries the signal type of source 1 for the measurement.
MEASUrement:MEAS<x>:SOUrce<x>	This command sets or queries the measurement source.
MEASUrement:MEAS<x>:SSC:NOMinalfreq	This command sets or queries the user-defined frequency for the measurement when the measurement type is SSC.
MEASUrement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype	This command sets or queries the frequency detection type for the measurement when the measurement type is SSC.
MEASUrement:MEAS<x>:STATUS?	Returns the pass fail status, if applicable, for the selected measurement
MEASUrement:MEAS<x>:TCKAVG	This command sets or queries the average clock period value used in DDR measurements.
MEASurement:MEAS<x>:TIMINGMode	This command sets or queries the Timing mode for the specified DDR measurement.
MEASUrement:MEAS<x>:TOEdge	This command sets or queries the 'to edge' type for the measurement.
MEASUrement:MEAS<x>:TOEDGESEARCHDIRect	This command sets or queries the 'to edge' search direction for the measurement.
MEASUrement:MEAS<x>:TOSYmbol:LOGIC2SOUrce	This command sets or queries the DDR digital measurement logic 1 source To Symbol.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:MEAS<x>:TOSYmbol:LOGIC3SOUrce	This command sets or queries the DDR digital measurement logic 2 source To Symbol.
MEASUrement:MEAS<x>:TOSYmbol:LOGIC4SOUrce	This command sets or queries the DDR digital measurement logic 3 source To Symbol.
MEASUrement:MEAS<x>:TOSYmbol:LOGIC5SOUrce	This command sets or queries the DDR digital measurement logic 4 source To Symbol.
MEASUrement:MEAS<x>:TOSYmbol:MEASUREAT	This command sets or queries the DDR digital measurement ToSymbol MeasureAT value.
MEASUrement:MEAS<x>:TRANSition	This command sets or queries the transition edges flag for the measurement.
MEASUrement:MEAS<x>:TYPe	This command sets or queries the measurement type.
MEASUrement:MEAS<x>:WINDOWLENgth	This command sets or queries the window length for the measurement.
MEASUrement:MEAS<x>:XUNIT?	Returns the horizontal scale units of the specified measurement.
MEASUrement:MEAS<x>:YUNIT?	Returns the vertical scale units of the specified measurement.
MEASUrement:MEASRange:MAX	This command sets or queries the global range maximum value.
MEASUrement:MEASRange:MIN	This command sets or queries the global range minimum value.
MEASUrement:MEASRange:STATE	Sets or queries the global range state.
MEASUrement:MINUI	This command sets or queries the minimum number of unit intervals required for BUJ analysis.
MEASUrement:POPUlation:LIMIT:STATE	This command sets or queries the global population limit state.
MEASUrement:POPUlation:LIMIT:VALue	This command sets or queries the global population limit value.
MEASUrement:REFLevels:ABSolute:FALLHigh	Sets or queries the value used as the high reference level of the falling edge.
MEASUrement:REFLevels:ABSolute:FALLLow	Sets or queries the value used as the low reference level of the falling edge.
MEASUrement:REFLevels:ABSolute:FALLMid	Sets or queries the value used as the mid reference level of the falling edge.
MEASUrement:REFLevels:ABSolute:HYSTeresis	Sets or queries the value of the hysteresis of the reference level.
MEASUrement:REFLevels:ABSolute:RISEHigh	Sets or queries the value used as the high reference level of the rising edge.
MEASUrement:REFLevels:ABSolute:RISELow	Sets or queries the value used as the low reference level of the rising edge.
MEASUrement:REFLevels:ABSolute:RISEMid	Sets or queries the value used as the mid reference level of the rising edge.
MEASUrement:REFLevels:ABSolute:TYPE	Sets or queries the reference level type.
MEASUrement:REFLevels:BASETop	Sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels.
MEASUrement:REFLevels:JITTERMODE	Sets or queries how often reference levels are calculated on Jitter measurements.
MEASUrement:REFLevels:METHod	Sets or queries the method used to calculate reference levels.
MEASUrement:REFLevels:MODE	This command sets or queries how often reference levels are calculated.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:REFLevels:PERCent:FALLHigh	Sets or queries the percentage used to calculate the high reference level of the falling edge.
MEASUrement:REFLevels:PERCent:FALLLow	Sets or queries the percentage used to calculate the mid reference level of the falling edge.
MEASUrement:REFLevels:PERCent:FALLMid	Sets or queries the percentage used to calculate the mid reference level of the falling edge.
MEASUrement:REFLevels:PERCent:HYSTeresis	Sets or queries the percentage used to calculate the hysteresis of the reference level.
MEASUrement:REFLevels:PERCent:RISEHigh	Sets or queries the percentage used to calculate the high reference level of the rising edge.
MEASUrement:REFLevels:PERCent:RISELow	Sets or queries the percentage used to calculate the low reference level of the rising edge.
MEASUrement:REFLevels:PERCent:RISEMid	Sets or queries the percentage used to calculate the mid reference level of the rising edge.
MEASUrement:REFLevels:PERCent:TYPE	Sets or queries the reference level percent type.
MEASUrement:REFLevels:TYPE	This command sets or queries the shared reference level method used for sources of measurement calculations.
MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh	Sets or queries the value used as the high reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow	Sets or queries the value used as the low reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:ABSolute:FALLMid	Sets or queries the value used as the mid reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis	Sets or queries the value of the hysteresis of the reference level.
MEASUrement:REF<x>:REFLevels:ABSolute:RISEHigh	Sets or queries the value used as the high reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:ABSolute:RISELow	Sets or queries the value used as the low reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:ABSolute:RISEMid	Sets or queries the value used as the mid reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:ABSolute:TYPE	Sets or queries the reference level type.
MEASUrement:REF<x>:REFLevels:BASETop	Sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels.
MEASUrement:REF<x>:REFLevels:METHod	Sets or queries the method used to calculate reference levels.
MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh	Sets or queries the percentage used to calculate the high reference level of the falling edge.

Table 2-36: Measurement commands (cont.)

Command	Description
MEASUrement:REF<x>:REFLevels:PERCent:FALLLow	Sets or queries the percentage used to calculate the low reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:PERCent:FALLMid	Sets or queries the percentage used to calculate the mid reference level of the falling edge.
MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis	Sets or queries the percentage used to calculate the hysteresis of the reference level.
MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh	Sets or queries the percentage used to calculate the high reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:PERCent:RISELow	Sets or queries the percentage used to calculate the low reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:PERCent:RISEMid	Sets or queries the percentage used to calculate the mid reference level of the rising edge.
MEASUrement:REF<x>:REFLevels:PERCent:TYPE	Sets or queries the reference level percent type.
MEASUrement:STATIstics:CYCLEMode	This command sets or queries whether cycle-cycle statistics are calculated for all measurements.

Miscellaneous command group

Miscellaneous commands do not fit into other categories.

Several commands and queries are common to all devices. The 488.2-1987 standard defines these commands. The common commands begin with an asterisk (*) character.

Table 2-37: Miscellaneous commands

Command	Description
AUTOSAVEPITIMEOUT	Sets or queries the idle time from the programmable interface before auto-save occurs.
AUTOSAVEUITIMEOUT	Sets or queries the idle time from the user interface before auto-save occurs.
AUTOSet	Sets or queries the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform.
AUTOSet:ACQuisition:ENAble	Sets or queries the Autoset acquisition setting adjustment.
AUTOSet:ENAble	Sets or queries the Autoset enable/disable mode.
AUTOSet:HORizontal:ENAble	Sets or queries Autoset's adjustment of horizontal settings.
AUTOSet:TRIGger:ENAble	Sets or queries Autoset's adjustment of trigger settings.
AUTOSet:VERTical:ENAble	sets or queries Autoset's adjustment of vertical settings.
AUTOSet:VERTical:OPTIMize	Sets or queries which vertical settings Autoset will optimize when the display mode is set to Overlay mode.
AUXout:EDGE	Sets or queries the direction in which the trigger output signal will transition when a trigger occurs.
AUXout:SOUrce	Sets or queries the trigger source at the BNC connection.
CLEAR	Clears acquisitions, measurements, and waveforms.
CONNected:REQUested:STATus	Sets the status of the Connected Scope Preference feature.
CONNected:STATus?	Returns the status of the Connected Scope Preference feature.
CONNected:USAGe:TRack:REQUested:STATus	Sets the tracking usage status of the Connected Scope Preference feature.
CONNected:USAGe:TRack:STATus?	Returns the tracking usage status of the Connected Scope Preference feature.
DATE?	Queries the date that the instrument displays.
*DDT	Sets or queries the commands that will be executed by the group execute trigger.
FPAnel:PRESS	Turns off the displayed menu.
FPAnel:TURN	Emulates a knob turn.
HEADer	Sets or queries the Response Header Enable State.
ID?	Returns identifying information about the instrument and its firmware.
*IDN?	Returns the instrument identification code.
LICense?	Queries all license parameters.
LICENSE:APPID?	Returns a comma-separated list of the active application IDs.
LICense:COUNt?	Returns a count of the number of active licenses installed.

Table 2-37: Miscellaneous commands (cont.)

Command	Description
LICense:ERRor?	This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue (alias for ALLEV?).
LICense:GMT?	Returns the GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.
LICense:HID?	Returns the instrument HostID unique identifier.
LICense:INSTall	Accepts a <block data> license and installs it on the instrument.
LICense:ITEM?	Returns the details pertaining to a specific license.
LICense:LIST?	Returns the active license nomenclatures as a comma-separated list of strings.
LICense:VALidate?	Accepts a license nomenclature as an argument and returns the status of the license.
LIC:UNINSTALL?	Returns the exit license information for the user to return to their TekAMS account.
LOCk	Sets or queries the front panel and touchscreen lock state.
*LRN?	Returns a listing of instrument settings.
MAINWindow:BADGe:BRINgtoview	Makes a specified badge visible.
MAINWindow:FONTSize	Sets the font size for UI text elements.
MAINWindow:RRBDisplaystate	Sets the display state of the Results readout bar to ON (displayed) or OFF (not displayed).
NEWpass	Changes the password for user protected data.
PASSWord	Provides access for changing user protected data.
PAUSe	Causes the interface to pause the specified number of seconds before processing any other commands.
REM	Specifies a comment which is ignored by the instrument.
ROSc:SOUrce	Selects or queries the selected source for the time base reference oscillator.
ROSc:STATE?	Returns whether the time base reference oscillator is locked.
SET?	Returns a listing of instrument settings.
SOCKETServer:ENAble	Enables or disables the socket server which supports a telnet or other TCPIP socket connection to send commands and queries to the instrument.
SOCKETServer:PORT	Sets the TCPIP port for the socket server connection.
SOCKETServer:PROTOCol	Sets or queries the protocol for the socket server.
TEKSecure	Initializes both waveform and setup memories.
TIMe	Queries the time displayed by the instrument.
TIMe:ZONe	Sets the time zone to the one specified.
TIMe:ZONe:UTCDELTa	Sets or queries the time zone using the difference between the desired time zone and UTC.
TOTaluptime?	Returns the total number of hours the instrument has been turned on since the NV memory was last programmed.
TOUCHSCReen:STATe	Sets or queries the enabled state of the touch screen.
*TRG	Performs the group execute trigger (GET).

Table 2-37: Miscellaneous commands (cont.)

Command	Description
*TST?	Tests the interface and returns status.
UNDO	Reverts the scope settings to a state before the previous command or user interface action.
UNLock	Unlocks front panel controls.
USBDevice:CONFigure	Used to configure the rear USB port to be off or enabled as a USBTMC device.
VERBose	Sets or queries the verbose state.

Plot command group

Plot commands let you select the type and control the appearance of your plots.

Table 2-38: Plot commands

Command	Description
PLOT:ADDNew	Adds the specified plot.
PLOT:DELete	Deletes the specified plot.
PLOT:LIST?	Lists all currently defined plots.
PLOT:PLOT<x>:BATHtub:BER	Sets or queries the bathtub BER value.
PLOT:PLOT<x>:BATHtub:XAXISUnits	Sets or queries the X-Axis unit, either unit intervals or seconds.
PLOT:PLOT<x>:BITType	Sets or queries the bit type to display for the specified eye diagram plot.
PLOT:PLOT<x>:EXPORTRaw?	Returns a binary stream of double values containing the x,y and hits value.
PLOT:PLOT<x>:EXTENDuis	Sets or queries number of UIs surrounding the eye boundary UIs that are used for fast eye rendering.
PLOT:PLOT<x>:EYERender	Sets or queries the eye rendering method for the specified plot.
PLOT:PLOT<x>:LABel:COLor	Sets or queries the color of the specified trend label.
PLOT:PLOT<x>:LABel:FONT:BOLD	Sets or queries the bold state of the specified trend label.
PLOT:PLOT<x>:LABel:FONT:ITALic	Sets or queries the italic state of the specified trend label.
PLOT:PLOT<x>:LABel:FONT:SIZE	Sets or queries the font size of the specified trend label.
PLOT:PLOT<x>:LABel:FONT:TYPE	Sets or queries the font type of the specified trend label.
PLOT:PLOT<x>:LABel:FONT:UNDERline	Sets or queries the underline state of the specified trend label.
PLOT:PLOT<x>:LABel:NAMe	Sets or queries the specified trend's label.
PLOT:PLOT<x>:LABel:XPOS	Sets or queries the x-position of the specified trend label.
PLOT:PLOT<x>:LABel:YPOS	Sets or queries the y-position of the specified trend label.
PLOT:PLOT<x>:MASK?	Returns the name of the mask test associated with the specified eye diagram plot.
PLOT:PLOT<x>:NUMBins	Sets or queries the current histogram resolution.
PLOT:PLOT<x>:SOUrce<x>	Sets or queries the measurement source.
PLOT:PLOT<x>:SPECtrum:BASE	Sets or queries the spectrum base. Undefined for non-spectrum plots.
PLOT:PLOT<x>:SPECtrum:DYNRange	Sets or queries the dynamic range value.
PLOT:PLOT<x>:TYPe	Sets or queries the current plot type for the selected plot.

Power command group

Table 2-39: Power commands

Command	Description
POWer:ADDNew	Adds the specified power measurement badge.
POWer:DELete	Deletes the specified power measurement badge.
POWer:POWer<x>:AUTOSet	Executes power autoset for the specified power measurement badge.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:CLRESPONSE:AMP[1-10]Val	Sets or queries the generator amplitude value of the specified configuration step for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:AMPMode	Sets or queries the amplitude mode for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:CONNECTSTATus?	Queries connection status to the external generator used with the specified Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude	Sets or queries the constant amplitude voltage for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:FREQ[1-11]Val	Sets or queries the generator frequency value of the specified configuration step for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:GENerator	Sets or queries the generator source used to send stimulus signals to the DUT, for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:GENerator	Sets or queries the generator source for the specified Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:GENIPADDress	Sets or queries the IP address of the external generator to be used with the specified Control Loop Response measurement.
POWer:POWer<x>:CLRESPONSE:IMPEDance	Sets or queries the vertical termination impedance for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:INPUTSOurce	Sets or queries the input source for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce	Sets or queries the output source for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:PPD	Sets or queries the points per decade (PPD) value for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:STARTFREQuency	Sets or queries the start frequency value for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:STOPFREQuency	Sets or queries the stop frequency value for the Control Loop Response power measurement.
POWer:POWer<x>:CLRESPONSE:TESTCONNection	Tests the connection to the external generator used with the specified Control Loop Response power measurement.
POWer:POWer<x>:CYCLEAmp:INPUTSOurce	Sets or queries the input source for cycle amplitude measurement of the specified power measurement number.
POWer:POWer<x>:CYCLEBase:INPUTSOurce	Sets or queries the input source for cycle base measurement of the specified power measurement number.
POWer:POWer<x>:CYCLEMAX:INPUTSOurce	Sets or queries the input source for cycle maximum measurement of the specified power measurement number.
POWer:POWer<x>:CYCLEMin:INPUTSOurce	Sets or queries the input source for cycle minimum measurement of the specified power measurement number.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:CYCLEPKPK:INPUTSOurce	Sets or queries the input source for cycle peak-to-peak measurement of the specified power measurement number.
POWer:POWer<x>:CYCLETop:INPUTSOurce	Sets or queries the input source for cycle top measurement of the specified power measurement number.
POWer:POWer<x>:DIDT:INPUTSOurce	Sets or queries the input source for di/dt measurement of the specified power measurement number.
POWer:POWer<x>:DIDT:SOURCEEDGEType	Sets or queries the edge type for di/dt measurement of the specified power measurement number.
POWer:POWer<x>:DVDT:INPUTSOurce	Sets or queries the input source for dv/dt measurement of the specified power measurement number.
POWer:POWer<x>:DVDT:SOURCEEDGEType	Sets or queries the edge type for dv/dt measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:INPUTType	Sets or queries the input type (AC or DC) for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce	Sets or queries the output 1 current source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce	Sets or queries the output 2 current source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce	Sets or queries the output 3 current source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:ISOUrce	Sets or queries the current source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:NUMOFOutputs	Sets or queries the number of outputs for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:OUTPUT1Type	Sets or queries the Output1 type for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:OUTPUT2Type	Sets or queries the Output2 type for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:OUTPUT3Type	Sets or queries the Output3 type for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:OUTPUTType	Sets or queries the Output type for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce	Sets or queries the output 1 voltage source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce	Sets or queries the output 2 voltage source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce	Sets or queries the output 3 voltage source for efficiency measurement of the specified power measurement number.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:EFFICIENCY:VSOUrce	Sets or queries the voltage source for efficiency measurement of the specified power measurement number.
POWer:POWer<x>:FREQUENCY:EDGe	Sets or queries the edge type for frequency measurement of the specified power measurement number.
POWer:POWer<x>:FREQUENCY:INPUTSOurce	Sets or queries the input source for frequency measurement of the specified power measurement number.
POWer:POWer<x>:GATing	Sets or queries the gating type for the specified power measurement badge.
POWer:POWer<x>:GATing:GLOBal	Sets or queries the gating settings for the specified power measurement badge.
POWer:POWer<x>:HARMONICS:CLASs	Sets or queries the class type for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:CMEThod	Sets or queries the fundamental current method for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:FUNDCURRent	Sets or queries the fundamental current value for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:HORDer	Sets or queries the order value for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:HSOURce	Sets or queries the source type for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:IPOWer	Sets or queries the input power value for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:ISOURce	Sets or queries the current source for SOA measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:LINEFREQUEncy	Sets or queries the value for the line frequency for the power Harmonics measurement.
POWer:POWer<x>:HARMONICS:ODDEVen	Sets or queries the harmonics value analysis format of the specified power measurement number.
POWer:POWer<x>:HARMONICS:PFACtor	Sets or queries the value of power factor for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:POWERRating	Sets or queries the power level for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:RCURRent	Sets or queries the rated current for the harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:STANDard	Sets or queries the test mode for harmonics measurement of the specified power measurement number.
POWer:POWer<x>:HARMONICS:STARTFREQUEncy	Sets or queries the value for the start frequency for the power Harmonics measurement. in the range of 1 Hz to 1 GHz.
POWer:POWer<x>:HARMONICS:UNITs	Sets or queries the harmonics results units of the specified power measurement number.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:HARMONICS:VSOURce	Sets or queries the voltage source for SOA measurement of the specified power measurement number.
POWer:POWer<x>:IMPEDANCE:AMP[1-10]Val	Sets or queries the signal generator amplitude setting of the specified profile step, for the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:AMPMode	Sets or queries the power amplitude mode for the Impedance measurement.
POWer:POWer<x>:IMPEDANCE:CONNECTSTATus?	Queries the instrument's connection status to the external generator, for the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:CONSTAMPlitude	Sets or queries the constant amplitude value for the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:FREQ[1-10]Val	Sets or queries the signal generator start frequency of the specified profile step, for the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:GENerator	Sets or queries the generator source for the Impedance power measurement.
POWer:POWer<x>:IMPEDANCE:GENIPADDress	Sets or queries the external generator IP Address associated with the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:IMPEDANCE	Sets or queries the output impedance of the generator for the specified Impedance power measurement.
POWer:POWer<x>:IMPEDANCE:INPUTSOurce	Sets or queries the source for the Impedance input measurement.
POWer:POWer<x>:IMPEDANCE:OUTPUTSOurce	Sets or queries the source for the Impedance output measurement.
POWer:POWer<x>:IMPEDANCE:PPD	Sets or queries the value for points per decade for the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:STARTFREQuency	Sets or queries the value for the start frequency of the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:STOPFREQuency	Sets or queries the value of the specified Impedance measurement.
POWer:POWer<x>:IMPEDANCE:TESTCONNection	This command tests the connection with the external instrument for the specified Impedance measurement.
POWer:POWer<x>:INDUCTANCE:EDGESource	Sets or queries the edge source for inductance measurement of the specified power measurement number.
POWer:POWer<x>:INDUCTANCE:ISOUrce	Sets or queries the current source for inductance measurement of the specified power measurement number.
POWer:POWer<x>:INDUCTANCE:VSOURce	Sets or queries the voltage source for inductance measurement of the specified power measurement number.
POWer:POWer<x>:INPUTCAP:ISOURce	Sets or queries the inrush current input source of the specified Input Capacitance measurement.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:INPUTCAP:PEAKCURRent	Sets or queries the peak current value of the specified Input Capacitance measurement.
POWer:POWer<x>:INPUTCAP:PEAKVOLTage	Sets or queries the peak voltage value of the specified Input Capacitance measurement.
POWer:POWer<x>:INPUTCAP:VSOURce	Sets or queries the input voltage source of the specified Input Capacitance measurement.
POWer:POWer<x>:INRUSHcurrent:INPUTSOurce	Sets or queries the input source of the specified Inrush Current measurement.
POWer:POWer<x>:INRUSHcurrent:PEAKCURRent	Sets or queries the peak current value of the specified Inrush Current measurement.
POWer:POWer<x>:IVSINTEGRALV:ISOURce	Sets or queries the current source for I vs Integral V measurement of the specified power measurement number.
POWer:POWer<x>:IVSINTEGRALV:VSOURce	Sets or queries the voltage source for I vs Integral V measurement of the specified power measurement number.
POWer:POWer<x>:LABel	Sets or queries the custom name for the specified power measurement badge.
POWer:POWer<x>:LINERIPPLE:INPUTSOurce	Sets or queries the input source for line ripple measurement of the specified power measurement number.
POWer:POWer<x>:LINERIPPLE:LFREQuency	Sets or queries the frequency present for line ripple measurement of the specified power measurement number.
POWer:POWer<x>:MAGNETICLOSS:ISOURce	Sets or queries the current source for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGNETICLOSS:VSOURce	Sets or queries the voltage source for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection	Sets or queries the coil cross section area for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:EDGESOURce	Sets or queries the edge signal source for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:ISOURce	Sets or queries the current source for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:LENgth	Sets or queries the conductor length for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs	Sets or queries the number of primary turns for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC1SOURce	Sets or queries the current source for secondary winding 1 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs	Sets or queries the number of turns of secondary winding 1 for magnetic measurement in the specified power measurement badge

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:MAGPROPERTY:SEC2SOURce	Sets or queries the current source for secondary winding 2 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs	Sets or queries the number of turns of secondary winding 2 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce	Sets or queries the current source for secondary winding 3 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs	Sets or queries the number of turns of secondary winding 3 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC4SOURce	Sets or queries the current source for secondary winding 4 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs	Sets or queries the number of turns of secondary winding 4 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC5SOURce	Sets or queries the current source for secondary winding 5 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs	Sets or queries the number of turns of secondary winding 5 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce	Sets or queries the current source for secondary winding 6 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs	Sets or queries the number of turns of secondary winding 6 for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:SECWINDings	Sets or queries the number of secondary windings for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:UNITs	Sets or queries the units for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:MAGPROPERTY:VSOURce	Sets or queries the voltage source for magnetic measurement of the specified power measurement number.
POWer:POWer<x>:NDUTYCYCLE:EDGEType	Sets or queries the clock edge type for negative duty cycle measurement of the specified power measurement number.
POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce	Sets or queries the input source for negative duty cycle measurement of the specified power measurement number.
POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce	Sets or queries the input source for negative pulse width measurement of the specified power measurement number.
POWer:POWer<x>:PDUTYCYCLE:EDGEType	Sets or queries the clock edge type for positive duty cycle measurement of the specified power measurement number.
POWer:POWer<x>:PDUTYCYCLE:INPUTSOurce	Sets or queries the input source for positive duty cycle measurement of the specified power measurement number.
POWer:POWer<x>:PERIOD:EDGe	Sets or queries the edge type for period measurement of the specified power measurement number.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:PERIOD:INPUTSOurce	Sets or queries the input source for period measurement of the specified power measurement number.
POWer:POWer<x>:POWERQUALITY:CCYCles	Sets or queries the calculate cycles over full cycles settings for the specified power quality measurement badge.
POWer:POWer<x>:POWERQUALITY:FREFerence	Sets or queries the frequency reference type for power quality measurement of the specified power measurement number.
POWer:POWer<x>:POWERQUALITY:ISOURce	Sets or queries the current source for power quality measurement of the specified power measurement number.
POWer:POWer<x>:POWERQUALITY:VSOURce	Sets or queries the voltage source for power quality measurement of the specified power measurement number.
POWer:POWer<x>:PPULSEWIDTH:INPUTSOurce	Sets or queries the input source for positive pulse width measurement in the specified power measurement number.
POWer:POWer<x>:PRESET	Sets or queries the input source for positive pulse width measurement of the specified power measurement number.
POWer:POWer<x>:PSRR:AMP[1-10]Val	Sets or queries the generator amplitude value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:AMPMode	Sets or queries the amplitude mode for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:CONNECTSTATus?	Queries the external instrument's connection status for the specified Power Supply Rejection Ratio (PSRR) measurement.
POWer:POWer<x>:PSRR:CONSTAMPlitude	Sets or queries the constant amplitude voltage for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:FREQ[1-11]Val	Sets or queries the generator frequency value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:GENerator	Sets or queries the generator source used to send stimulus signals to the DUT for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:GENerator	Sets or queries the generator source for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:GENIPADDress	Sets or queries the instrument's IP Address associated with the specified Power Supply Rejection Ratio (PSRR) measurement.
POWer:POWer<x>:PSRR:IMPEDance	Sets or queries the vertical termination impedance for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:INPUTSOurce	Sets or queries the input source for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:OUTPUTSOurce	Sets or queries the output source for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:PPD	Sets or queries the points per decade (PPD) value for the Power Supply Rejection Ratio (PSRR) power measurement.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:PSRR:STARTFREQuency	Sets or queries the start frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:STOPFREQuency	Sets or queries the stop frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.
POWer:POWer<x>:PSRR:TESTCONNection	This command tests the connection with the external instrument for the specified Power Supply Rejection Ratio (PSRR) measurement.
POWer:POWer<x>:RDSON:DEVICEType	Sets or queries the device type for the power drain source on resistance measurement for RDSon measurement of the specified power measurement number.
POWer:POWer<x>:RDSON:ISOURce	Sets or queries the current source for RDSon measurement of the specified power measurement number.
POWer:POWer<x>:RDSON:VSOURce	Sets or queries the voltage source for RDSon measurement of the specified power measurement number.
POWer:POWer<x>:REFLevels:ABSolute:FALLHigh	Sets or queries the falling edge for high reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:FALLLow	Sets or queries the falling edge for low reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:FALLMid	Sets or queries the falling edge for mid reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:HYSTeresis	Sets or queries the absolute hysteresis value for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:RISEHigh	Sets or queries the rising edge for high reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:RISELow	Sets or queries the rising edge for low reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:RISEMid	Sets or queries the rising edge for mid reference level in absolute units for the specified power measurement badge.
POWer:POWer<x>:REFLevels:ABSolute:TYPE	Sets or queries the type of measurement levels when reference level is set to absolute for the specified power measurement badge.
POWer:POWer<x>:REFLevels:BASETop	Sets or queries the reference level base top method for the specified power measurement badge.
POWer:POWer<x>:REFLevels:METHod	Sets or queries the method to configure reference level values for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:FALLHigh	Sets or queries the falling edge for high reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:FALLLow	Sets or queries the falling edge for low reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:FALLMid	Sets or queries the falling edge for mid reference level in percentage for the specified power measurement badge.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:REFLevels:PERCent:HYSTeresis	Sets or queries the hysteresis in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:RISEHigh	Sets or queries the rising edge for high reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:RISELow	Sets or queries the rising edge for low reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:RISEMid	Sets or queries the rising edge for mid reference level in percentage for the specified power measurement badge.
POWer:POWer<x>:REFLevels:PERCent:TYPE	Sets or queries the reference levels for the specified power measurement badge.
POWer:POWer<x>:RESUlts:ALLAcqs:MAXimum?	Queries the maximum value of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:ALLAcqs:MEAN?	Queries the mean value of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:ALLAcqs:MINimum?	Queries the minimum value of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:ALLAcqs:PK2PK?	Queries the peak-to-peak value of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:ALLAcqs:POPUlation?	Queries the population (number of complete cycles) of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:ALLAcqs:STDDev?	Queries the standard deviation value of all acquisitions for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:F1MAG?	Queries the first harmonics magnitude value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:F3MAG?	Queries the third harmonics magnitude value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:FREQUENCY?	Queries the fundamental frequency for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:IRMS?	Queries the RMS current value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:MAXimum?	Queries the maximum value of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:MEAN?	Queries the mean value of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:MINimum?	Queries the minimum value of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:PK2PK?	Queries the peak-to-peak value of the current acquisition for the measurement parameter of the specified power measurement number.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:RESUlts:CURRentacq:POHCL?	Queries the limit of partial odd harmonic current for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:POHCM?	Queries the measured value of partial odd harmonic current for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:POHCS?	Queries the status of partial odd harmonic current for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:POPUlation?	Queries the population (number of complete cycles) of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:RMS?	Queries the RMS value of the source selected for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:STATUS?	Queries the status of the measurement for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:STDDev?	Queries the standard deviation value of the current acquisition for the measurement parameter of the specified power measurement number.
POWer:POWer<x>:RESUlts:CURRentacq:THDF?	Queries the total harmonic distortion (fundamental) value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:THDR?	Queries the total harmonic distortion (RMS) value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:TRPWR?	Queries the true power value for the specified power measurement badge.
POWer:POWer<x>:RESUlts:CURRentacq:VRMS?	Queries the RMS voltage value for the specified power measurement badge.
POWer:POWer<x>:SEQSETup	Runs the power measurement sequence setup.
POWer:POWer<x>:SEQuence	Sets or queries the run state of a single sequence power measurement.
POWer:POWer<x>:SOA:ISOURce	Sets or queries the current source for SOA measurement of the specified power measurement number.
POWer:POWer<x>:SOA:POINT	Sets or queries the X or Y coordinate value for a specified SOA mask point.
POWer:POWer<x>:SOA:RECAllmask	Recalls or queries the recall mask file name of the specified power measurement number.
POWer:POWer<x>:SOA:RECAllmask:FILEName	Sets or queries the file name for saving SOA mask file name of the specified power measurement number.
POWer:POWer<x>:SOA:SAVemask	Saves the mask file as per the name configured and at the configured path or queries the mask file name, path, and file type for the SOA measurement of the specified power measurement number.
POWer:POWer<x>:SOA:SAVemask:AUTOINCrement	Sets or queries the state of auto-increment for saved SOA mask file names of the specified power measurement number.
POWer:POWer<x>:SOA:SAVemask:FILEName	Sets or queries the mask file name for SOA measurement of the specified power measurement number.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:SOA:SAVemask:FOLDer	Sets or queries the mask file folder path for SOA measurement of the specified power measurement number.
POWer:POWer<x>:SOA:VSOURce	Sets or queries the voltage source for SOA measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:DEVICEType	Sets or queries the conduction calculation method for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:GATESOurce	Sets or queries the gate voltage (V_g) for the switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:ILEVELAbs	Sets or queries the current level (Ton-Start & Stop) in absolute units for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:ILEVELPct	Sets or queries the current level (Ton-Start & Stop) in percentage for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:ISOURce	Sets or queries the current source for the switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts	Sets or queries the level units for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:RDSOn	Sets or queries the RDS(on) value for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType	Sets or queries the configuration type for the switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:VCESat	Sets or queries the value for the VCE(sat) value for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:VGLevel	Sets or queries the gate voltage value (V_g Level Ton-Start) for the switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs	Sets or queries the voltage level (Ton-Start & Stop) in absolute units for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct	Sets or queries the voltage level (Ton-Start & Stop) in percentage for switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGLOSS:VSOURce	Sets or queries the voltage source for the switching loss measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGRIPPLE:INPUTSOurce	Sets or queries the input source for switching ripple measurement of the specified power measurement number.
POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency	Sets or queries the switching frequency for switching ripple measurement of the specified power measurement number.
POWer:POWer<x>:TURNOFFtime:FREQuency	Sets or queries the input frequency used by the AC or DC converter of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:INPUTLEVel	Sets or returns the input frequency used by the AC or DC converter of the specified Turn Off Time measurement.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:TURNOFFtime:INPUTSOurce	Sets or queries the input source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:MAXTIMe	Sets or queries the maximum turn off time of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:MAXVoltage	Sets or queries the maximum voltage of the specified Turn OffTime measurement.
POWer:POWer<x>:TURNOFFtime:NUMOUTputs	Sets or queries the number of outputs of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce	Sets or queries the output 1 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage	Sets or queries the output 1 voltage level of the of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce	Sets or queries the output 2 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage	Sets or queries the output 2 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce	Sets or queries the output 3 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage	Sets or queries the output 3 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce	Sets or queries the output 4 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage	Sets or queries the output 4 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce	Sets or queries the output 5 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage	Sets or queries the output 5 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce	Sets or queries the output 6 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage	Sets or queries the output 6 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce	Sets or queries the output 7 source of the specified Turn Off Time measurement.
POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage	Sets or queries the output 7 voltage level of the specified Turn Off Time power measurement.
POWer:POWer<x>:TURNOFFtime:TYPE	Sets or queries the type of AC/DC converter used in the specified Turn Off Time power measurement.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:TURNONtime:FREQuency	Sets or queries the input frequency used by the AC or DC converter of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:INPUTLEVel	Sets or queries the input voltage level of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:INPUTSOurce	Sets or queries the input source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:MAXTIMe	Sets or queries the maximum turn on time of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONTIME:MAXVoltage	Sets or queries the maximum voltage setting of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:NUMOUTputs	Sets or queries the number of outputs for the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT1SOURce	Sets or queries the output 1 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage	Sets or queries the output 1 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce	Sets or queries the output 2 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage	Sets or queries the output 2 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT3SOURce	Sets or queries the output 3 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage	Sets or queries the output 3 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT4SOURce	Sets or queries the output 4 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage	Sets or queries the output 4 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT5SOURce	Sets or queries the output 5 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage	Sets or queries the output 5 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT6SOURce	Sets or queries the output 6 source of the specified Turn On Time measurement.
POWer:POWer<x>:TURNONtime:OUTPUT6VOLTage	Sets or queries the output 6 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:OUTPUT7SOURce	Sets or queries the output 7 source of the specified Turn On Time measurement.

Table 2-39: Power commands (cont.)

Command	Description
POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage	Sets or queries the output 7 voltage level of the specified Turn On Time power measurement.
POWer:POWer<x>:TURNONtime:TYPE	Sets or queries the type of AC/DC converter used in the specified Turn On Time power measurement.
POWer:POWer<x>:TYPe	Sets or queries the measurement type for the specified power measurement badge.

Save and Recall command group

Use the commands in the Save and Recall Command Group to store and retrieve internal waveforms and settings. When you save a setup, you save all the settings of the instrument. When you recall a setup, the instrument restores itself to the state that it was in when you originally saved that setting.

Table 2-40: Save and Recall commands

Command	Description
FACtory	Resets the instrument to factory default settings.
RECAll:SESsion	Restores the state of the instrument from a saved session file.
RECAll:SETUp	Recalls saved instrument settings.
RECAll:WAVEform	Recalls a stored waveform to a reference memory location.
SAVe:EVENTtable:BUS	Saves bus results table to the specified file.
SAVe:EVENTtable:CUSTom	Saves the results table to the specified file path and name.
SAVe:EVENTtable:CUSTom:COMMents	Sets or queries comments to be included in saved results table files.
SAVe:EVENTtable:CUSTom:DATAFormat	Sets or queries the data format to use for saving results table data.
SAVe:EVENTtable:CUSTom:INCLUDEREFs	Sets or queries whether to include displayed reference waveforms with saved results table files.
SAVe:EVENTtable:MEASUrement	Saves data (measurement) results to the specified file.
	Saves a search results table to the specified file.
SAVe:IMAGe	Saves a capture of the screen contents to the specified image file.
SAVe:IMAGe:COMPosition	Sets or queries the color mode for saved images (normal or inverted).
SAVe:IMAGe:VIEWTYpe	Sets or queries the view type for saved images. Currently only FULLScreen is supported.
SAVe:PLOTData	Saves the plot data of the currently selected plot in a specified file. Supported file format is CSV.
SAVe:REPOrt	Saves a report to the specified file or, if no argument is specified, uses the folder and file name specified by the related commands.
SAVe:REPOrt:COMMents	Sets or queries the comments to be included in saved report files.
SAVe:SESsion	Saves the state of the instrument, including reference waveforms, to a saved session file.
SAVe:SETUp	Saves the current instrument state to the specified file.
SAVe:SETUp:INCLUDEREFs	Sets or queries whether displayed reference waveforms are to be included in saved setups.
SAVe:WAVEform	Saves the specified waveform(s) to the specified destination file(s).
SAVe:WAVEform:GATing	Sets the method to save a specified part of the waveform data.
SAVe:WAVEform:GATing:RESAMPLErate	Saves the waveform data at a sample interval.
SAVe:WAVEform:SOURCELIst?	Returns a list of the available waveforms that can be specified as the source for the SAVe:WAVEform command.

Save On command Group

Use this group of commands to program the instrument to save images, measurements, waveforms, or the instrument setup, on triggers that you select. These commands still function, however the **Act On Event** commands are preferred. Please see the Act On Event section for continued development and enhancements. (See Table 2-18.)

Table 2-41: Save On commands

Command	Description
SAVEON:FILE:DEST	Sets or queries the location where files are saved.
SAVEON:FILE:NAME	Sets or queries the file name to use when SAVEON:TRIGer is ON.
SAVEON:IMAGe:FILEFormat	Sets or queries the file format to be used for saved image files.
SAVEON:IMAGe	Sets or queries whether to save a screen capture when a trigger occurs.
SAVEON:TRIGger	Sets or queries whether to save a file when a trigger occurs.
SAVEON:WAVEform	Sets or queries whether to save a waveform when a limit test failure, mask failure, or trigger occurs.
SAVEON:WAVEform:FILEFormat	Sets or queries the file format for saving waveform.
SAVEON:WAVEform:SOURce	Sets or queries the sources for saving waveforms.

Search and Mark command group

Use search and mark commands to seek out and identify information in waveform records that warrant further investigation.

Table 2-42: Search and Mark commands

Command	Description
SEARCH:ADDNew	Adds the specified search.
SEARCH:DELete	Deletes the specified search.
SEARCH:DELETEALL	Deletes all the active instances of measurements defined in the scope application.
SEARCH:LIST?	Lists all currently defined searches.
SEARCH:SEARCH<x>:COPy	Copies the search criteria to or from the trigger.
SEARCH:SEARCH<x>:NAVigate	Sets the navigation action for search marks.
SEARCH:SEARCH<x>:TOTAL?	Queries the total number of found search marks for this search.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:CONDition	Specifies a field or condition for an ARINC429 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:ERRTYPe	Sets or queries the error type when searching on an ARINC429 bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:QUALifier	Sets or queries the qualifier to be used when searching on label data for an ARINC429 bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:VALue	Sets or queries the low value when searching on an ARINC429 label field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue	Sets or queries the high value when searching on an ARINC429 label field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SDI:VALue	Sets or queries the when searching on an ARINC429 SDI field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SSM:VALue	Sets or queries the when searching on an ARINC429 SSM field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:QUALifier	Sets or queries the qualifier to be used when searching on data in the DATA field for an ARINC429 bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:VALue	Sets or queries the low value when searching on an ARINC429 data field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:HIVALue	Sets or queries the high value when searching on an ARINC429 data field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition	Sets or queries the condition (word select / frame sync, or matching data) to be used when searching on an audio bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HITDMVALue	Sets or queries the binary data string for the high data word to be used when searching on an TDM audio bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HIVALue	Sets or queries the binary data string for the high data word to be used when searching on an audio bus signal.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:OFFSet	Sets or queries the data offset value (TDM channel) to be used when searching on a TDM type audio bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:QUALifier	Sets or queries the qualifier to be used when searching on an audio bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:TDMVALue	Sets or queries the binary data string for the single or low data word to be used when searching on an TDM audio bus signal
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:VALue	Sets or queries the binary data string for the single or low data word to be used when searching on an audio bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:WORD	Sets or queries the alignment of the data (left, right or either) to be used when searching on a non-TDM type audio bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition	Sets or queries the field or condition for which to search an AutoEthernet.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue	Sets or queries the binary data value to be used when searching on an AutoEthernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:OFFSet	Sets or queries the data offset value, in bytes, to use when searching on the AutoEthernet data field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe	Sets or queries the length of the data string, in bytes, to use when searching on the AutoEthernet bus signal.
SEARCH:SEARCHx>:TRIGger:A:BUS:AUTOETHERnet:DATa:VALue	Sets or queries the binary value to use when searching on the AutoEthernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:DESTinationaddr:VALue	Sets or queries the binary destination address value to use when searching on an AutoEthernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:PROTOcol:VALue	Sets or queries the binary protocol value to use when searching on the Ethernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:SOUrceaddr:VALue	Sets or queries the binary source address value to use when searching on an AutoEthernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:DESTination:VALue	Sets or queries the binary MAC address destination value to use when searching on an AutoEthernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:SOUrce:VALue	Sets or queries the binary MAC address source value to use when searching on an AutoEthernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:HIVALue	Sets or queries the binary MAC length high value to use when searching on an AutoEthernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:VALue	Sets or queries the MAC length value to use when searching on an Ethernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QTAG:VALue	Sets or queries the binary Q-tag information to use when searching on an AutoEthernet bus signal.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QUALifier	Sets or queries the qualifier to be used when searching on an AutoEthernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:ACKnum:VALue	Sets or queries the binary ack number value to use when searching on an AutoEthernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:DESTinationport:VALue	Sets or queries the binary destination port value to use when searching on the AutoEthernet TCP header destination port number.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SEQnum:VALue	Sets or queries the TCP header sequence number value to use when searching on the AutoEthernet TCP header destination port number.
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SOUrceport:VALue	Sets or queries the binary source port value to use when searching on an AutoEthernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition	Sets or queries the CAN bus trigger condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:DIRection	Sets the CAN search type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:OFFSet	Sets or queries the data offset value, in bytes, to use when searching on the CAN data field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:QUALifier	Sets or queries the CAN bus trigger data qualifier.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:SIZe	Sets or queries the CAN bus trigger data size.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue	Sets or queries the binary data value to be used when searching on a CAN bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType	Sets or queries the type of error condition for a CAN bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit	Sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit	Sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEtype	Sets or queries CAN bus trigger frame type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODe	Sets or queries the CAN bus trigger identifier mode.
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue	Sets or queries CAN bus trigger identifier value.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:BLUe:VALue	Specifies the blue data string used for DPHY triggering if the trigger condition is on RGB packet.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:CONDition	Sets or queries the trigger condition for an DPHY bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:SIZe	Specifies the length of the data string in bytes to be used for an DPHY trigger if the trigger condition is DATA.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:VALue	Specifies the binary data string used for DPHY triggering if the trigger condition is DATA.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ERRor:TYPe	Sets or queries the error type for DPHY bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand	Sets or queries the escape mode command type for DPHY bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:GREen:VALue	Specifies the green data string used for DPHY triggering if the trigger condition is on RGB packet.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:MODe:TYPe	Sets or queries the mode type for DPHY bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:LIST	Sets or queries the packet list for DPHY bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:TYPe	Sets or queries the packet type for DPHY bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:NUMBer	Specifies the pixel number to be used for DPHY triggering if the trigger condition is on pixel number pattern.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:SEARCHOPTion	Sets or queries the search pattern type for DPHY bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:RED:VALue	Specifies the red data string used for DPHY triggering if the trigger condition is on RGB packet.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:WORDCOUNt:VALue	Specifies the word count data string used for DPHY triggering if the trigger condition is set on any pixel packet.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CB	Specifies the Cb data string used for DPHY triggering if the trigger condition is on YCbCr packet.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CR	Specifies the Cr data string used for DPHY triggering if the trigger condition is on YCbCr packet.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:Y	Specifies the Y data string used for DPHY triggering if the trigger condition is on YCbCr packet.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:U	Specifies the U data string used for DPHY triggering if the trigger condition is on YUV packet.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:V	Specifies the V data string used for DPHY triggering if the trigger condition is on YUV packet.
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:Y	Specifies the Y data string used for DPHY triggering if the trigger condition is on YUV packet.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition	Specifies a field or condition within an Ethernet frame to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue	Sets or queries the binary data value to be used when searching on an Ethernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet	Specifies the data offset value, in bytes, to use when searching on the Ethernet data field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:QUALifier	Sets the qualifier to be used when searching on an Ethernet bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe	Specifies the number of contiguous TCP/IPv4/MAC client bytes to use when searching on the Ethernet data field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue	Specifies the binary value to use when searching on the Ethernet data field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:DESTinationaddr:VALue	Specifies the 32-bit value to use when searching on the Ethernet IPv4 header address destination field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:PROTOcol:VALue	Specifies the 8-bit value to use when searching on the Ethernet IPv4 header protocol field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:SOUrceaddr:VALue	Specifies the 32-bit value to use when searching on the Ethernet IPv4 header address source field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:DESTination:VALue	Specifies the 48-bit value to use when searching on the Ethernet MAC address destination field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:SOUrce:VALue	Specifies the 48-bit value to use when searching on the Ethernet MAC address source field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:HIVALue	Specifies the 16-bit high value to use when searching on the Ethernet MAC length/type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:VALue	Specifies the 16-bit value to use when searching on the Ethernet MAC length/type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:QTAG:VALue	Specifies the 32-bit value to use when searching on the Ethernet Q-Tag field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:ACKnum:VALue	Specifies the 32-bit value to use when searching on the Ethernet TCP header acknowledgement number.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:DESTinationport:VALue	Specifies the 16-bit value to use when searching on the Ethernet TCP header destination port number.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SEQnum:VALue	Specifies the 32-bit value to use when searching on the Ethernet TCP header sequence number.
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SOUrceport:VALue	Specifies the 16-bit value to use when searching on the Ethernet TCP header source port number.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:HIVALue	Sets the high binary address value to be used when triggering on a eUSB bus signal.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:VALue	Sets the binary address value to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:CONDition	Sets or queries the search type for the specified search.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:HIVALue	Sets the high binary data value to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:OFFSet	Sets the byte offset to look for a data pattern at, in bytes, to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:QUAlifier	Sets the qualifier to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:SIZe	Sets the length of the data string, in bytes, to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:TYPe	Specifies the eUSB trigger type to be valid on any, data0, data1, data2, or mdata condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:VALue	Sets the binary data value to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATABITS:SIZe	Sets the number of Sync bits, in bytes, to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDPoint:VALue	Sets the endpoint binary value to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDSEARch	Specifies eUSB Search EOP Bits to be On and Off.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe	Sets the number of EOP bits, in bytes, to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:QUAlifier	Sets the qualifier to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe	Sets the number of EOP bits, in bits, to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe	Sets the number of EOP bits, in bits, to be used when triggering on a EUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ERRType	Specifies the eUSB trigger type to be valid on a PID, Token CRC5, Data CRC16, or Bit stuffing condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:HANDSHAKEType	Specifies the eUSB trigger type to be valid on any, ack, nak, stall, or nyet condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue	Sets the binary Register Access Protocol address value to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:COMMand	Specifies the eUSB Register Access Protocol search option

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:DATa:VALue	Sets the binary Register Access Protocol data value to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:OPTion	Specifies the eUSB Register Access Protocol search option.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SOFFRAMENUMber	Sets the binary frame number to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPECIALType	Specifies the eUSB trigger type to be valid on any, err, split, ping, or reserved condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:ET:VALue	Specifies the eUSB trigger type endpoint condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue	Sets the binary hub address value to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue	Sets the binary port address value to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SC:VALue	Specifies the eUSB trigger type start/complete condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SE:VALue	Specifies the eUSB trigger type start/complete condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNC:QUAlifier	Sets the qualifier to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe	Sets the number of sync bits, in bits, to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe	Sets the number of sync bits, in bits, to be used when triggering on a eUSB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCSEARch	Specifies eUSB Search Sync Bits to be On and Off.
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:TOKENType	Specifies the eUSB trigger type to be valid on any, sof, out, in, or setup condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CONDition	Sets or queries FlexRay bus search trigger condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue	Sets or queries the FlexRay bus search cycle count value.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:VALue	Sets or queries the FlexRay bus search cycle count value.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue	Sets or queries the FlexRay bus search cycle count value.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier	Sets or queries the FlexRay bus search trigger cycle count qualifier.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue	Sets or queries the FlexRay bus search trigger data value.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet	Sets or queries the FlexRay bus search trigger data offset.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier	Sets or queries the FlexRay bus search trigger data qualifier.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:SIZe	Sets or queries the FlexRay bus search trigger data size.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:EOFTYPE	Sets or queries the FlexRay bus search trigger end of file type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE	Sets or queries the FlexRay bus search trigger error type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue	Sets or queries the high value when searching on a FlexRay bus frame id field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:QUALifier	Sets the qualifier to be used when searching on a FlexRay bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:VALue	Sets the low value when searching on a FlexRay bus id field.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEType	Sets or queries the FlexRay bus search trigger frame type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC	Sets or queries the FlexRay bus search trigger header CRC.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount	Sets or queries the FlexRay bus search trigger header cycle count.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID	Sets or queries the FlexRay bus search trigger header frame id.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits	Sets or queries the FlexRay bus search header Indicator Bits.
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth	Sets or queries the FlexRay bus search trigger header payload length.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe	Sets or queries the address mode for the specified I2C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:VALue	Sets or queries the address string when the search condition for the specified search is Address or AddressData, to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition	Sets or queries the trigger condition for the specified I2C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:DIRection	Sets or queries the direction of the data for the I2C bus trigger search to determine where to place a mark.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:SIZe	Sets or queries the length of the data string in bytes used for the specified I2C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue	Sets or queries the data value of the data token for the specified I2C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:MODe	Sets or queries the address mode for the specified I3C bus trigger search to determine where to place a search mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:VALue	Sets or queries the I3C address value for which to search, to determine where to place a search mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCR:VALue	Sets or queries the binary field string used for the I3C search if the trigger condition is BCR.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCRType:VALue	Sets or queries the binary field string used for the I3C bus search if the trigger condition is BCRTYPE.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BPACKets	Sets or queries the I3C Broadcast Packets search to Broadcast or Response.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BRGTID:VALue	Ssets or queries the binary field string used for the I3C trigger if the trigger condition is SETBRGTGT packet ID.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CCODe:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is SETBRGTGT packet ID.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CONDition	Sets or queries the trigger condition for the specified I3C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:DIRection	Sets or queries the direction of the data for the I3C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:SIZe	Sets or queries the length of the data string in bytes used for the specified I3C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:VALue	Sets or queries the data value of the data token for the specified I3C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCR:VALue	Sets or queries the binary field string used for the I3C bus search if the trigger condition is DCR.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCRType:VALue	Sets or queries the binary field string used for the I3C bus search if the trigger condition is DCRTYPE.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DWORd:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is DATAWORDS.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ERRORTYPe	Sets or queries the error type condition for the specified I3C bus search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:EVENTBYTe:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is FREQUENCYBYTE.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:FREQBYTe:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is GETSTATUSLSB.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSLSb:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is GETSTATUSLSB.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSMSb:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is GETSTATUSMSB.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ID:VALue	Sets or queries the binary field string used for the I3C search if the trigger condition is ENTDAA packet ID.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:INACCBYTe:VALue	Sets or queries the binary field string used for the I3C search if the trigger condition is INACCURACYBYTE.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXREAD:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is MAXREAD.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXRTURN:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is MAXREADTURN.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXWRITe:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is MAXWRITE.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:PACKets	Sets or queries the I3C Direct Packets to Direct or Response.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SADDress:VALue	Sets or queries the binary field string used for the I3C bus searchif the trigger condition is SLAVEADDRESS.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:BROADCASTPacket	Sets or queries the SDR broadcast packets for the specified I3C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:DIRECTPacket	Sets or queries the SDR direct packet types for the specified I3C bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATEBYTe:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is STATEBYTE.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATic:ADDRess	Sets or queries the binary field string used for the I3C search if the trigger condition is STATICADDRESS.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SUPPORTBYTe:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is SUPPORTEDBYTE.
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:TESTMODe:VALue	Sets or queries the binary field string used for the I3C trigger if the trigger condition is TESTMODE.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue	Specifies the high data value to be used in a LIN search.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:QUALifier	Sets or queries the LIN bus search trigger data qualifier.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZe	Sets or queries the LIN bus search trigger data size.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue	Sets or queries the LIN bus search trigger data value.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE	Sets or queries the LIN bus search trigger error type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue	Sets or queries the LIN bus search trigger identifier value.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition	Specifies a field or condition for an Manchester bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:SIZe	Specifies the number of bits per word for the specified Manchester bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:VALue	Specifies the binary string used for Manchester triggering if the trigger condition is data when searching on a Manchester signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:ERRor:TYPe	Sets or queries the error to be used when searching on a Manchester signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:HEADER:VALue	Specifies the binary string used for Manchester triggering if the trigger condition is header when searching on a Manchester signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:packetOffData:VALue	Specifies the binary string used for Manchester triggering if the trigger condition is DATA.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:SYNC:VALue	Specifies the binary string used for Manchester triggering if the trigger condition is Sync when searching on a Manchester signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:TRAILER:VALue	Specifies the binary string used for Manchester triggering if the trigger condition is Trailer when searching on a Manchester signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ADDress:VALue	Specifies the binary address string used for MDIO triggering if the trigger condition is address.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:CONDition	Sets or queries the trigger condition for MDIO bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DATa:VALue	Specifies the binary data string used for MDIO triggering if the trigger condition is DATA.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DEVicetype:VALue	Specifies the binary device type string used for MDIO triggering if the trigger condition is Device type.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ERRType	Sets or queries the error type to be used when triggering on MDIO data.This applies when the MDIo bus trigger condition is set to ERRor.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:OPCode:VALue	Specifies the binary op-code string used for MDIO triggering if the trigger condition is op-code .
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:PHYSicaladdress:VALue	Specifies the binary physical address string used for MDIO triggering if the trigger condition is physical address.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:REGisteraddress:VALue	Specifies the binary register address string used for MDIO triggering if the trigger condition is register address.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition	Sets or queries the field or condition for a MIL-STD-1553 bus to search on.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe	Sets or queries the type of error condition for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit	Sets or queries the value of the command word Transmit / Receive bit for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:PARity	Sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:COUNt	Sets or queries the value of the command word "word count" field for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:SUBADdress	Sets or queries the value of the command word subaddress field for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:QUALifier	Sets or queries the qualifier to be used when searching on command word addresses for a MIL-STD-1553 bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:VALue	Sets or queries the low value when searching on command word addresses for a MIL-STD-1553 bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:HIVALue	Sets or queries the high value when searching on command word addresses for a MIL-STD-1553 bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:PARity	Sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue	Sets or queries the value when searching on data words for a MIL-STD-1553 bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:PARity	Sets or queries the value of the status word parity bit for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:QUALifier	Sets or queries the qualifier to be used when searching on status word addresses for a MIL-STD-1553 bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:VALue	Sets or queries the low value when searching on status word addresses for a MIL-STD-1553 bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:HIVALue	Sets or queries the high value when searching on status word addresses for a MIL-STD-1553 bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:ME	Sets or queries the value of the message error bit (ME bit, bit 9) in a status word for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:INSTR	Sets or queries the value of the instrumentation bit (INSTR bit, bit 10) in a status word for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SRQ	Sets or queries the value of the status word service request bit (SRQ bit, bit 11) in a status word for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BCR	Sets or queries the value of the broadcast command received bit (BCR bit, bit 15) in a status word for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BUSY	Sets or queries the value of the busy bit (BUSY bit, bit 16) in a status word for a MIL-STD-1553 bus to search on.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SUBSF	Ssets or queries the value of the subsystem flag bit (SUBSF bit, bit 17) in a status word for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:DBCA	Sets or queries the value of the dynamic bus control acceptance bit (DBCA bit, bit 18) in a status word for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:TF	Sets or queries the value of the terminal flag bit (TF bit, bit 19) in a status word for a MIL-STD-1553 bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:NRZ:CONDition	Specifies a field or condition for an NRZ bus to search on.
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:SIZe	Specifies the length of the data string, in bytes, to use when searching on the NRZ signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:VALue	Sets or queries the binary data value to be used when searching on a NRZ signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:PARallel:DATa:VALue	Sets or queries the data value for a parallel bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:BLOCKDATa:VALue	Sets or queries the binary block data string used for PSI5 triggering if the trigger condition is block ID.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:CONDition	Sets or queries the trigger condition for an PSI5 bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:ECU:SENSor:VALue	Sets or queries the binary data string used for PSI5 triggering if the trigger condition is Data in ECU to Sensor communication mode.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:REGion:A:VALue	Sets or queries the binary data string used for PSI5 triggering if the trigger condition is data region A.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:REGion:B:VALue	Sets or queries the length of the data string in bytes to be used for an PSI5 trigger if the trigger condition is data region B.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATABITs	Sets or queries the binary register address string used for PSI5 triggering if the trigger condition is data in ECU to Sensor communication mode.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:ERRORTYPe	Sets or queries the error type for an PSI5 bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:FUNCTIONCODETYPe	Sets or queries the binary data string used for PSI5 triggering if the trigger condition is a Function Code.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:REGister:ADDRess:VALue	Sets or queries the binary register address string used for PSI5 triggering if the trigger condition is Register Address.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:REGISTERADDRess	Sets or queries the register address for PSI5 bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:SENSor:ADDRess:VALue	Sets or queries the binary sensor address string used for PSI5 triggering if the trigger condition is Sensor Address.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:SENSORSTATus:TYPe	Sets or queries the sensor status for PSI5 bus.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:STATus:VALue	Sets or queries the binary status string used for PSI5 triggering if the trigger condition is status.
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:VALue	Sets or queries the data string value for the specified RS232c bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition	Sets or queries the condition for an RS232C bus search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:SIZe	Sets or queries the length of the data string in bytes to be used for an RS232 bus search to determine where to place a mark when the search condition is Data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition	Sets or queries the type of data search for an 8B10b bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:DISParity	Sets or queries the disparity to be used when searching on a 8B10b signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:ERRor	Sets or queries the error type to be used when searching on a 8B10b signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue	Sets or queries the control symbol value to be used when searching on 8b10b signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue	Specifies the 8 bit value to be used when searching on an 8b10b signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue	Specifies the 10 bit value to be used when searching on an 8b10b signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:MODe	Sets or queries the mode of data search to be used when searching on 8b10b signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg	Specifies the 10 bit RD positive value to be used when searching on an 8b10b signal
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos	Specifies the 10 bit RD negative value to be used when searching on an 8b10b signal
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TYPe	Sets or queries the type of symbol when searching on 8b10b signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ADDRess:TYPe	Sets or queries the Address in SDLC bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:CONDition	Sets or queries the trigger condition for a SDLC bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:SIZe	Specifies the length of the data string in bytes to be used for SDLC triggering if the trigger condition is DATA.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:VALue	Specifies the binary data string used for SDLC triggering if the trigger condition is DATA.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ERROR:TYPe	Sets or queries the Error Type in SDLC bus.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:FRAMe:TYPe	Sets or queries the Frame Type in SDLC bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:STADDress:VALue	Specifies the binary Station Address string used for SDLC triggering if the trigger condition is Address.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:SUPervisory:FRAMETYPe	Sets or queries the Supervisory Frame Type in SDLC bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:UNNumbered:FRAMETYPe	Sets or queries the Unnumbered Frame Type in SDLC bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition	Sets or queries the search condition for a SENT bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType	Sets or queries the error type to be used when searching on SENT data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC	Sets or queries the CRC error type to be used when searching on SENT data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue	Sets or queries the high binary fast channel 1 value to use when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier	Sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 1.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue	Sets or queries the binary fast channel 1 value to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue	Sets or queries the high binary fast channel 2 value to use when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier	Sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 2.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue	Sets or queries the binary fast channel 2 value to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:HIVALue	Sets or queries the high binary fast message counter value to use when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:QUALifier	Sets or queries the qualifier to be used when searching on SENT fast packet bus data for the secure format counter.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue	Sets or queries the binary fast message counter value to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue	Sets or queries the binary fast message inverted nibble value to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue	Sets or queries the binary status value to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier	Sets or queries the qualifier to be used when searching on SENT pause pulses.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue	Sets or queries the maximum number of pause clock ticks to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue	Sets or queries the minimum number of pause clock ticks to be used when searching on a SENT bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue	Sets or queries the high binary Slow channel data value to use when searching on SENT Slow packet bus data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier	Sets or queries the qualifier to be used when searching on SENT Slow packet bus data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue	Sets or queries the binary slow channel data value to be used when searching on SENT Slow packet bus data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue	Sets or queries the binary Slow identifier value to be used when searching on SENT Slow packet bus data.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce	Sets or queries the bus source for the specified bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition	S
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe	Sets or queries the control character type to use when searching on a SpaceWire bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe	Sets or queries the control code type to use when searching on a SpaceWire bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe	Sets or queries the length of the data string, in bytes, to use when searching on a SpaceWire bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:VALue	Sets or queries the binary data value to be used when searching on an SpaceWire bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:ERRORTYPe	Sets or queries the error type for which to search on the SpaceWire bus signal.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:TIMECode:VALue	Sets or queries the binary data string used for SpaceWire search if the search condition is Time Code.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition	Sets or queries the trigger condition for the specified SPI bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZe	Sets or queries the length of the data string in bytes used for the specified SPI bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue	Sets or queries the data value of the data token for the specified SPI bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCETYpe	Sets or queries trigger Source for SPI bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition	This command sets or queries the search condition for an SPMI bus.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:VALue	This command specifies the binary data string used for SPMI when the search condition is MASTERREAd, MASTERWRIte, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, LONGEXTREGWRIte, or REG0WRIte.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:MASTERADDRess:VALue	This command sets or queries the binary data string that identifies the master address used in SPMI search.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPonse	This command sets or queries whether or not to search for No Response frames.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:REGISTERADDRess:VALue	This command sets or queries the binary data string that identifies the register address used in SPMI triggering.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:SLAVEADDRess:VALue	This command sets or queries the binary data string that identifies the slave address used in SPMI.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:RESPonse	Sets or queries the command response of respective command type for SVID bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:TYPe	Sets or queries the command type for SVID bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:CONDition	Sets or queries the trigger condition for SVID bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:ERRor:TYPe	Sets or queries the error type for SVID bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:TYPe	Sets or queries the payload type for SVID bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLOAD:VALue	Sets or queries the payload value to be used when search is on SVID bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:SLAVE:ADDRESS	Sets or queries the slave address to be used when search is on SVID bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:HIVALue	Sets or queries the address value for normal token to be used with In Range and Out of Range qualifiers for the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:VALue	Sets or queries the address value for normal token for the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition	Sets or queries the search condition for the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:HIVALue	Sets or queries the data value for data token used with In Range and Out of Range qualifiers for the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:OFFSet	Sets or queries data offset for the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:SIZe	Sets or queries the length of the data string in bytes for the specified USB bus trigger search to determine where to place a mark.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:TYPe	Sets or queries the data packet type for the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:VALue	Sets or queries the data value for data token for the specified bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALue	Sets or queries the endpoint value for normal token for the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE	Sets or queries the error type for the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HANDSHAKEType	Sets or queries the handshake type for the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SOFFRAMENUMber	Sets or queries the frame number string to use for the Start of Frame for the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPECIALType	Sets or queries the PID value for the USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:ET:VALue	Sets or queries the Endpoint Type value for the specified USB bus trigger split token field search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:HUB:VALue	Sets or queries the hub address of the specified USB bus trigger on split token field search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:PORT:VALue	Sets or queries the port address for the specified USB bus trigger on split token field search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SC:VALue	Sets or queries the Start/Complete value for the specified USB bus trigger on split token field search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SE:VALue	Sets or queries the Start/End value for the specified USB bus trigger on split token field search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TOKENType	Sets or queries the token type when the specified USB bus trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe	Sets or queries the slope for an edge trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod	This command sets or queries the DDRRead search burst detection method.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLatency	This command sets or queries the DDR read logic state burst latency value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLENGTH	This command sets or queries the DDR read logic state burst length value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSActive	This command sets or queries the DDR Read chip select active state.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSLevel	This command sets or queries the DDR Read chip select level value.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode	This command sets or queries the DDR Read chip select mode.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSSource	This command sets or queries the DDR Read search chip select source.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATARate	Sets or queries the DDR read search data rate for DDR3 and LPDRR3 standards of the specified search.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATASource	Sets or queries the DDR read data source when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis	Sets or queries the DDR read hysteresis reference level value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol	This command sets or queries the DDR read logic source 1 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol	This command sets or queries the DDR read logic source 2 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol	This command sets or queries the DDR read logic source 3 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol	This command sets or queries the DDR read logic source 4 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin	Sets or queries the DDR read margin reference level value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MAXCAS	This command sets or queries the DDR read chip maximum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MINCas	This command sets or queries the DDR read chip minimum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:POSTAMBLE:LENGth	Sets or queries the DDR read postamble length when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE	Sets or queries the DDR read preamble type when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH	Sets or queries the DDR read data high reference level value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW	Sets or queries the DDR read data low reference level value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID	Sets or queries the DDR read data mid reference level value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH	Sets or queries the DDR read reference level strobe high value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW	Sets or queries the DDR read reference level strobe low value, when the search type is DDR READ.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:MID	Sets or queries the DDR read reference level strobe mid value, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode	Sets or queries the DDR read reference level mode to auto or manual, when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard	Sets or queries the DDR read search standard as DDR3 or LPDDR3.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource	Sets or queries the DDR read strobe source when the search type is DDR READ.
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:TOLERance	This command sets or queries the DDR read logic state tolerance value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod	This command sets or queries the DDR Read/Write search burst detection method
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLatency	This command sets or queries the DDR read/write logic state burst latency value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLENGTH	This command sets or queries the DDR read/write logic state burst length value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSActive	This command sets or queries the DDR Read/Write chip select active state.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSLevel	This command sets or queries the DDR Read/Write chip select level value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode	This command sets or queries the DDR read/write chip select mode.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSSource	This command sets or queries the DDR Read/Write search chip select source.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate	Sets or queries the DDR read/write data rate for DDR3 and LPDRR3 standards of the specified search.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource	Sets or queries the DDR read/write data source when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis	Sets or queries the DDR read/write hysteresis reference level value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol	This command sets or queries the DDR read/write logic source 1 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol	This command sets or queries the DDR read/write logic source 2 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol	This command sets or queries the DDR read/write logic source 3 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol	This command sets or queries the DDR read/write logic source 4 symbol.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin	Sets or queries the DDR read/write margin reference level value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MAXCAS	This command sets or queries the DDR read/write chip maximum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MINCas	This command sets or queries the DDR read/write chip minimum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth	Sets or queries the DDR read/write postamble length when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE	Sets or queries the DDR read/write preamble type when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH	Sets or queries the DDR read/write data high reference level value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW	Sets or queries the DDR read/write data low reference level value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID	Sets or queries the DDR read/write data mid reference level value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH	Sets or queries the DDR read/write reference level strobe high value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW	Sets or queries the DDR read/write reference level strobe low value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID	Sets or queries the DDR read/write reference level strobe mid value, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode	Sets or queries the DDR read/write reference level mode to auto or manual, when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard	Sets or queries the DDR read/write search standard as DDR3 or LPDDR3.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STROBESource	Sets or queries the DDR read/write strobe source when the search type is DDR READWRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:TOLERance	This command sets or queries the DDR read/write logic state tolerance value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod	This command sets or queries the DDRWRITE search burst detection method.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLatency	This command sets or queries the DDR write logic state burst latency value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLENGTH	This command sets or queries the DDR write logic state burst length value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSActive	This command sets or queries the DDR write chip select active state.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSLevel	This command sets or queries the DDR Write chip select level value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode	This command sets or queries the DDR Write chip select mode.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSSource	This command sets or queries the DDR Write search chip select source.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATARate	Sets or queries the DDR write search data rate for DDR3 and LPDRR3 standards of the specified search.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATASource	Sets or queries the DDR write data source when the search type is DDR Write.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis	Sets or queries the DDR write hysteresis reference level value when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol	This command sets or queries the DDR write logic source 1 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol	This command sets or queries the DDR write logic source 2 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol	This command sets or queries the DDR write logic source 3 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol	This command sets or queries the DDR write logic source 4 symbol.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin	Sets or queries the DDR write margin reference level value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MAXCAS	This command sets or queries the DDR write chip select maximum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MINCas	This command sets or queries the DDR write chip select minimum Column Access Strobe (CAS) value.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth	Sets or queries the DDR write postamble length when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE	Sets or queries the DDR write preamble type when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH	Sets or queries the DDR write data high reference level value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW	Sets or queries the DDR write data low reference level value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:MID	Sets or queries the DDR write data mid reference level value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH	Sets or queries the DDR write reference level strobe high value, when the search type is DDR WRITE.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW	Sets or queries the DDR write reference level strobe low value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID	Sets or queries the DDR write reference level strobe mid value, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode	Sets or queries the DDR write reference level mode to auto or manual, when the search type is DDR WRITE.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard	Sets or queries the DDR write search standard as DDR3 or LPDDR3.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STROBESource	Sets or queries the DDR write strobe source when the search type is DDR Write.
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:TOLERance	This command sets or queries the DDR write logic state tolerance value.
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe	Sets or queries the slope for an edge trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce	Sets or queries the source waveform for an edge trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCtion	Sets or queries the logic operator for a pattern or state trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:USEClockedge	Sets whether or not Logic search uses a clock source.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn	Sets or queries the condition setting for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth	Sets or queries the width setting for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:EDGE	Sets or queries the clock slope setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce	Sets or queries the clock source setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime	Sets or queries the hold time setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime	Sets or queries the setup time setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe	Sets or queries the time setting for a timeout trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:DELTATime	Sets or queries the transition time setting for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:WHEn	Sets or queries the condition setting for a transition trigger search to determine where to place a mark.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:TYPe	Sets or queries the trigger type setting for a search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth	Sets or queries the width setting for a window search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition	Sets or queries the LIN bus search trigger condition.
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:QUALifier	Sets or queries the qualifier to be used when searching on a USB bus signal.
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold	Sets or queries the source threshold level for an edge trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCk:THReshold	Sets or queries the logic clock threshold for a logic trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:DELTatime	Specifies the Logic search delta time value.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce	Sets or queries the channel to use as the clock source for logic trigger.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:CH<x>	Sets or queries the voltage level to use for logic trigger search.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:MATH<x>	Sets the voltage level to use for logic trigger search.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:REF<x>	Sets the voltage level to use for logic trigger search.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x>_D<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:MATH<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:POLarity	Sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic search type.
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:WHEn	Sets or queries the condition for generating an A or B logic search with respect to the defined input pattern.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit	Specifies the upper limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOGICQUALification	Specifies whether or not to use logic qualification for a pulse width search.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit	Specifies the lower limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity	Specifies the polarity for a pulse width search.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce	Sets and queries the source for the pulsewidth search input.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold	Sets or queries the source threshold level for a pulse width trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn	Specifies to search for a pulse with a specified width.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification	Specifies whether or not to use logic qualification for a runt search.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity	Specifies the polarity for the runt search.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce	Sets and queries the source for the Runt search input.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH	Sets or queries the source threshold HIGH level for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW	Sets or queries the source threshold LOW level for a runt trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:THReshold	Sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x>	Sets or queries the voltage level to use for setup & hold trigger search.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x>	Sets or queries the voltage level to use for setup & hold trigger search.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x>	Sets or queries the voltage level to use for setup & hold trigger search.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>_D<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:MATH<x>	Sets or queries the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x>	Sets and returns the conditions used for generating an A logic pattern.
SEARCH:SEARCH<x>:TRIGger:A:STATE	Sets or queries the enabled state of the search.
SEARCH:SEARCH<x>:TRIGger:A:STOPAcq	Sets or queries whether acquisitions are stopped when a search hit is found.

Table 2-42: Search and Mark commands (cont.)

Command	Description
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification	Sets whether or not to use logic qualification for a timeout search.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity	Sets or queries the polarity to be used for a Timeout search.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce	Sets and queries the source for timeout search input.
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold	Sets or queries the source threshold level for a timeout trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:LOGICQUALification	Specifies whether or not to use logic qualification for a transition search.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:POLarity	Specifies the polarity for the transition search.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:SOUrce	Sets and queries the source for the transition search input.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:HIGH	Sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:LOW	Sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:CROSSIng	Sets or queries the window trigger threshold crossing of the selected trigger Source.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:LOGICQUALification	Specifies or queries whether or not to use logic qualification for a window search.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity	Sets or queries the window trigger threshold crossing of the selected trigger Source.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOUrce	Sets and queries the source for the window search input.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH	Sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW	Sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark.
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn	Sets or queries the window search event.
SEARCH:SELected	Sets or queries the selected search.
SEARCHTABle	This command adds or deletes a new search event table in an Option 5-WIN (Microsoft Windows 10 OS) TekExpress compliance testing application.

Self Test command group

The Self Test commands control the selection and execution of diagnostic tests.

Table 2-43: Self Test commands

Command	Description
DIAg:LOOP:OPTion	Sets or queries the type of looping desired.
DIAg:LOOP:OPTion:NTIMes	Sets or queries how many loops to run.
DIAg:LOOP:STOP	Stops diagnostics looping.
DIAg:MODe	Sets or queries the diagnostics mode.
DIAg:RESUlt?	Returns the diagnostics results.
DIAg:RESUlt:FLAg?	Returns the status of the diagnostics (single area).
DIAg:RESUlt:LOG?	Returns the status of the diagnostic area.
DIAg:SELect	Selects or queries an available diagnostic area.
DIAg:STATE	Sets the instrument operating state.
TOUCHSCReen:CALibrate	Starts the touch screen calibration procedure.

Spectrum View command group

The Spectrum View commands control the selection and execution of spectrum commands.

Table 2-44: Spectrum View commands

Command	Description
CH<x>:SV:CENTERFrequency	Sets or queries the center frequency of the specified spectrum trace channel for Spectrum View.
CH<x>:SV:POSition	Sets or queries the Spectrum Time setting of the specified spectrum trace channel in the Spectrum View.
CH<x>:SV:SPANABovebw?	Queries whether the stop frequency for the specified spectrum trace channel is above the scope input bandwidth.
CH<x>:SV:SPANBELowdc?	Queries whether the start frequency for the specified spectrum trace channel is below 0 Hz.
CH<x>:SV:STARTFrequency?	Queries the start frequency for the specified spectrum trace channel in Spectrum View
CH<x>:SV:STATE	Sets or queries the on/off setting of data acquisition for the specified spectrum trace channel.
CH<x>:SV:STOPFrequency?	Queries the stop frequency of the specified spectrum trace channel for Spectrum View
SV:CH<x>:RF_AVErage:NUMAVg	Sets or queries the number of acquisitions to be used when creating the Average trace for the specified spectrum trace channel in the Spectrum View.
SV:CH<x>:RF_MAGnitude:FORMat	Sets or queries the format of the Magnitude vs. Time trace for the specified channel.
SV:CH<x>:RF_PHASe:REFerence:DEGrees	Sets or queries the channel Phase Reference value in degrees.
SV:CH<x>:RF_PHASe:REFerence:POSition	Sets or queries whether the channel Phase Reference is located at the Trigger position or at the Cursor A position.
SV:CH<x>:RF_PHASe:REFerence:TIMe	Sets or queries the channel Phase Reference time in seconds.
SV:CH<x>:RF_PHASe:WRAP:DEGrees	Sets or queries the Phase Wrap limit for the Phase vs. Time trace for the specified channel.
SV:CH<x>:RF_PHASe:WRAP:STATE	Sets or queries whether Phase Wrap is applied to the Phase vs. Time trace for the specified channel.
SV:CH<x>:SELect:RF_AVErage	Sets or queries whether the Average trace is displayed for the specified spectrum trace channel in the Spectrum View.
SV:CH<x>:SELect:RF_FREQuency	Sets or queries whether the Frequency vs. Time trace for the specified channel is displayed in the Waveform View.
SV:CH<x>:SELect:RF_MAGnitude	Sets or queries whether the Magnitude vs. Time trace for the specified channel is displayed in the Waveform View.
SV:CH<x>:SELect:RF_MAXHold	Sets or queries whether the Max Hold trace is displayed for the specified spectrum trace channel in the Spectrum View.
SV:CH<x>:SELect:RF_MINHold	Sets or queries whether the Min Hold trace is displayed for the specified spectrum trace channel in the Spectrum View.
SV:CH<x>:SELect:RF_NORMal	Sets or queries whether the Normal trace is displayed for the specified spectrum trace channel in the Spectrum View.

Command	Description
SV:CH<x>:SELect:RF_PHASe	Sets or queries whether the Phase vs. Time trace for the specified channel is displayed in the Waveform View.
SV:CH<x>:SELTrace	Sets or queries the spectrum trace type to show for the specified channel in the Spectrum View.
SV:CH<x>:SQUELCH:STATE	Sets or queries whether Squelch is enabled for the RF vs Time traces for the specified channel.
SV:CH<x>:SQUELCH:THReshold	Sets or queries the Squelch threshold value for the RF vs Time traces for the specified channel.
SV:CH<x>:UNIts	Sets or queries the vertical units to be used in the Spectrum View for the specified channel for absolute logarithmic amplitudes.
SV:LOCKCenter	Sets or queries whether the Center Frequency value is locked across all channels in the Spectrum View.
SV:LOCKSpectrum	Sets or queries whether the Spectrum Time value is locked across all spectrum trace channels in the Spectrum View.
SV:MARKER:PEAK:EXCURsion	Sets or queries the minimum peak excursion value, in dB, for the Spectrum View trace peak markers.
SV:MARKER:PEAK:MAXimum	Sets or queries the maximum number of Spectrum View trace peaks that can have automatic markers placed on them.
SV:MARKER:PEAK:STATE	Sets or queries the visibility of peak markers for Spectrum View traces.
SV:MARKER:PEAK:THReshold	Sets or queries the threshold value of the peak markers available for frequency domain traces.
SV:MARKER:REFERence	Sets the Center Frequency of the currently selected Spectrum View channel to the frequency indicated by the Reference Marker
SV:MARKER:REFERence:AMPLITUDE?	Queries the actual amplitude (vertical) value of the Reference Marker in user-set units.
SV:MARKER:REFERence:FREQuency?	Queries the frequency of the Reference Marker, in Hz, when the Spectrum View trace markers are on.
SV:MARKER:TYPe	Sets or queries the peak marker type (either DELTa or ABSolute).
SV:RBW	Sets or queries the resolution bandwidth (RBW) when the RBW mode has been set to Manual.
SV:RBWMode	Sets or queries the resolution bandwidth (RBW) mode, either Automatic or Manual.
SV:RF_PHASe:REFerence:MASTer	Sets or queries the channel used as the Master Phase Reference.
SV:SPAN	Sets or queries the span setting for all channels in the Spectrum View.
SV:SPANRBWRatio	This command specifies the ratio of the span to the resolution bandwidth (RBW) that will be used when the RBW Mode is set to AUTO.
SV:WINDOW	Sets or queries the window type used by the windowing function of the Spectrum View.

Status and Error command group

Use the commands in the Status and Error command Group to determine the status of the instrument and control events.

Several commands and queries used with the instrument are common to all devices. The IEEE Std 488.2-1987 defines these commands and queries. The common commands begin with an asterisk (*) character.

Table 2-45: Status and Error commands

Command	Description
ALLEv?	Returns all events and their messages.
BUSY?	Returns instrument status.
*CLS	Clears status.
DESE	Sets or queries the bits in the Device Event Status Enable Register.
*ESE	Sets or queries the bits in the Event Status Enable Register.
*ESR?	Returns the contents of the Standard Event Status Register.
EVENT?	Returns event code from the event queue.
EVMsg?	Returns event code, message from the event queue.
EVQty?	Returns number of events that are enabled in the queue.
*OPC	Generates the operation complete message in the standard event status register when all pending operations are finished Or returns “1” when all current operations are finished.
*OPT?	Returns a comma separated list of installed options as an arbitrary ASCII string (no quotes).
*PSC	Sets or queries the power on status flag.
*PUD	Sets or queries a string of protected user data.
*RST	Resets the instrument to factory default settings.
*SRE	Sets or queries the bits in the Service Request Enable Register.
*STB?	Returns the contents of the Status Byte Register.
*WAI	Prevents the instrument from executing further commands until all pending operations finish.

Trigger command group

Use the commands in the Trigger Command Group to control all aspects of triggering for the instrument.

There are two triggers: A and B. Where appropriate, the command set has parallel constructions for each trigger.

You can set the A or B triggers to edge mode. Edge triggering lets you display a waveform at or near the point where the signal passes through a voltage level of your choosing.

You can also set A or B triggers to pulse or logic modes. With pulse triggering, the instrument triggers whenever it detects a pulse of a certain width or height. Logic triggering lets you logically combine the signals on one or more channels. The instrument then triggers when it detects a certain combination of signal levels. The trigger types of Pulse Width, Timeout, Runt, Window, and Rise/Fall Time can be further qualified by a logic pattern. This is referred to as logic qualification.

Table 2-46: Trigger commands

Command	Description
TRIGger	Forces a trigger event to occur or returns current trigger parameters for the instrument.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition	Specifies a field or condition for an ARINC429 bus to trigger on.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:ERRTYPe	Sets or queries the error type when triggering on an ARINC429 bus signal.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:QUALifier	Sets or queries the qualifier to be used when triggering on label data for an ARINC429 bus signal.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:VALue	Sets or queries the low value when triggering on an ARINC429 label field.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue	Sets or queries the high value when triggering on an ARINC429 label field.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue	sSets or queries the when triggering on an ARINC429 SDI field.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue	Sets or queries the when triggering on an ARINC429 SSM field.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:QUALifier	Sets or queries the qualifier to be used when triggering on data in the DATA field for an ARINC429 bus signal.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:VALue	Sets or queries the low value when triggering on an ARINC429 data field.
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:HIVALue	Sets or queries the high value when trigger on an ARINC429 data field.
TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition	Sets the condition (word select, start of frame, or matching data) to be used when triggering on an audio bus signal.
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue	Sets the upper word value to be used when triggering on an audio bus signal.

Table 2-46: Trigger commands (cont.)

Command	Description
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDMVALue	Sets or queries the binary data string for the high data word to be used when triggering on an TDM audio bus signal.
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet	Sets the data offset value to be used when triggering on an audio bus signal.
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier	Sets the qualifier (<, >, =, <=, >=, not =, in range, out of range) to be used when triggering on an audio bus signal.
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:TDMVALue	Sets or queries the binary data string for the single or low data word to be used when triggering on an TDM audio bus signal.
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue	Sets the lower word value to be used when triggering on an audio bus signal.
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD	Sets the alignment of the data (left, right or either) to be used to trigger on an audio bus signal.
TRIGger:{A|B}:BUS:B<x>:CAN:CONDition	Sets the condition (start of frame, frame type, identifier, matching data, EOF, missing ACK field, bit-stuffing error) to be used when triggering on a CAN bus signal.
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:DIRection	Sets the data direction (read, write or “nocare”) to be used to search on a CAN bus signal.
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:OFFSet	Sets or queries the data offset value, in bytes, to use when triggering on the CAN data field.
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:QUALifier	Sets the qualifier (<, >, =, not =, <=, >=) to be used when triggering on a CAN bus signal.
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe	Sets the length of the data string, in bytes, to be used when triggering on a CAN bus signal.
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue	Sets the binary data value to be used when triggering on a CAN bus signal.
TRIGger:{A|B}:BUS:B<x>:CAN:ERRType	Sets or queries the type of error condition for a CAN bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:CAN:FD:BRSBit	Sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:CAN:FD:ESIBit	Sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:CAN:FRAMEtype	Sets the frame type (data, remote, error or overload) to be used when triggering on a CAN bus signal.
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:MODe	Sets the addressing mode (standard or extended format) to be used when triggering on a CAN bus signal.
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:VALue	Sets the binary address value to be used when triggering on a CAN bus signal.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition	Specifies a field or condition within an Ethernet frame to trigger on.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue	When the Ethernet trigger condition is set to `DATa`, and the qualifier is set to either `INrange` or`OUTrange`, this command specifies the upper data value of the range.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet	When the Ethernet trigger condition is set to `DATa`, this command specifies where in the data field to look for the data trigger value.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:QUALifier	Sets or queries the qualifier to be used when triggering on an Ethernet bus signal.

Table 2-46: Trigger commands (cont.)

Command	Description
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe	When the Ethernet trigger condition is set to DATa, this command specifies the number of contiguous TCP/IPv4/MAC client data bytes to trigger on.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue	When the Ethernet trigger condition is set to DATa, and the qualifier is set to LESSthan, MOREthan, EQual, UNEQual, LESSEQual or MOREEQual, this command specifies the value to trigger on.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue	When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 32–bit destination address that is to be used in the trigger.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue	When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 8–bit protocol field that is to be used in the trigger.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue	When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 32-bit source address that is to be used in the trigger.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:VALue	When the Ethernet trigger condition is set to MACADDress, this command specifies the 48–bit MAC destination address that is to be used in the trigger.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:SOUrce:VALue	When the Ethernet trigger condition is set to MACADDress, this command specifies the 48–bit MAC source address value that is to be used in the trigger.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue	When the Ethernet trigger condition is set to MACLENgth, and the qualifier is set to INrange or OUTrange, this command specifies the upper data value of the range.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue	When the Ethernet trigger condition is set to MACLENgth, and the qualifier is set to LESSthan, MOREthan, EQual, UNEQual, LESSEQual or MOREEQual, this command specifies the 16–bit value to trigger on.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:VALue	When the Ethernet trigger condition is set to QTAG, this command specifies the 32-bit Q-Tag value to trigger on.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue	When the Ethernet trigger condition is set to TCPHeader, this command specifies the 32–bit acknowledgement number that is to be used in the trigger.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue	When the Ethernet trigger condition is set TCPHeader, this command specifies the 16–bit destination port address value that is to be used in the trigger.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue	When the Ethernet trigger condition is set to TCPHeader, this command specifies the 32–bit sequence number that is to be used in the trigger.
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue	When the Ethernet trigger condition is set to TCPHeader, this command specifies the 16–bit source port address that is to be used in the trigger.
TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition	Specifies the condition to use when triggering on a FlexRay bus signal (start of frame, frame type, ID, cycle count, header, data, ID and data, EOF, error).
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue	Specifies the high value when triggering on a FlexRay bus cycle count field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:QUALifier	Specifies the qualifier (<, >, =, <=, >=, ≠, in range, out of range) to use when triggering on the FlexRay bus cycle count field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue	Specifies the low value when triggering on the FlexRay bus cycle count field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue	Specifies the high value when triggering on the FlexRay bus data field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:OFFSet	Specifies the offset of the data string, in bytes, when triggering on the FlexRay bus data field.

Table 2-46: Trigger commands (cont.)

Command	Description
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:QUALifier	Specifies the qualifier (<, >, =, <=, >=, ≠, in range, out of range) to use when triggering on the FlexRay bus data field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:SIZe	Specifies the length of the data string, in bytes, when triggering on the FlexRay bus data field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:VALue	Specifies the low value when triggering on the FlexRay bus data field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:EOFTYPE	Specifies the end of file type (static, dynamic or any) when triggering on the FlexRay bus EOF field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:ERRTYPE	Specifies the error type when triggering on the FlexRay bus signal.
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue	Specifies the high value when triggering on the FlexRay bus frame ID field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:QUALifier	Specifies the qualifier to use when triggering on the FlexRay bus frame ID field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALue	Specifies the low value when triggering on the FlexRay bus frame ID field.
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEType	Specifies the frame type (normal, payload, null, sync or startup) when triggering on the FlexRay bus signal.
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CRC	Specifies the CRC portion of the binary header string when triggering on the FlexRay bus signal.
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CYCLEcount	Specifies the cycle count portion of the binary header string when triggering on the FlexRay bus header.
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:FRAMEID	Specifies the frame ID portion of the binary header string when triggering on the FlexRay bus header.
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:INDBits	Specifies the indicator bits portion of the binary header string when triggering on the FlexRay bus header.
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:PAYLength	Specifies the payload length portion of the binary header string when triggering on the FlexRay bus header.
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:MODe	Specifies the I²C address mode to 7 or 10-bit.
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:VALue	Specifies the binary address string used for the I²C trigger if the trigger condition is ADDRESS or ADDRANDDATA.
TRIGger:{A|B}:BUS:B<x>:I2C:CONDition	Specifies the trigger condition for an I²C trigger.
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:DIRection	Specifies the I²C trigger type to be valid on a Read, Write, or No Care condition.
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:SIZe	Specifies the length of the data string in bytes to be used for an I2C trigger if the trigger condition is DATA or ADDRANDDATA.
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:VALue	Specifies the binary data string used for I2C triggering if the trigger condition is DATA or ADDRANDDATA.
TRIGger:{A|B}:BUS:B<x>:LIN:CONDition	Specifies the trigger condition for LIN.
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:HIVALue	Specifies the high data value string used for a LIN bus trigger when the trigger condition is DATA or IDANDDATA and the data qualifier is INRANGE or OUTRANGE.

Table 2-46: Trigger commands (cont.)

Command	Description
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:QUALifier	Specifies the LIN data qualifier. This only applies if the trigger condition is IDANDDATA or DATA.
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe	Specifies the length of the data string in bytes to be used for LIN trigger.
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:VALue	Specifies the binary data string to be used for LIN trigger condition if trigger condition is ID or IDANDDATA.
TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE	Specifies the error type be used for LIN trigger.
TRIGger:{A|B}:BUS:B<x>:LIN:IDentifier:VALue	Specifies the binary address string used for LIN trigger if the trigger condition is ID or IDANDDATA.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:CONDition	Sets or queries the field or condition for a MIL-STD-1553 bus to trigger on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:ERRTYPe	Sets or queries the type of error condition for a MIL-STD-1553 bus to trigger on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:TRBit	Sets or queries the value of the command word Transmit / Receive bit for a MIL-STD-1553 bus to trigger on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:PARity	Sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:COUNt	Sets or queries the value of the command word "word count" field for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:SUBADdress	Sets or queries the value of the command word "subaddress" field for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:QUALifier	Sets or queries the qualifier to be used when triggering on command word addresses for a MIL-STD-1553 bus.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:VALue	Sets or queries the low value when triggering on command word addresses for a MIL-STD-1553 bus.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:HIVALue	Sets or queries the high value when triggering on command word addresses for a MIL-STD-1553 bus.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:PARity	Sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:VALue	Sets or queries the value when triggering on data words for a MIL-STD-1553 bus.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:PARity	Sets or queries the value of the status word parity bit for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:QUALifier	Sets or queries the qualifier to be used when triggering on status word addresses for a MIL-STD-1553 bus.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:VALue	Sets or queries the low value when triggering on status word addresses for a MIL-STD-1553 bus.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:HIVALue	Sets or queries the high value when triggering on status word addresses for a MIL-STD-1553 bus.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME	Sets or queries the value of the message error bit (ME bit, bit 9) in a status word for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:INSTR	Sets or queries the value of the instrumentation bit (INSTR bit, bit 10) in a status word for a MIL-STD-1553 bus to triggering on.

Table 2-46: Trigger commands (cont.)

Command	Description
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SRQ	Sets or queries the value of the status word service request bit (SRQ bit, bit 11) in a status word for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BCR	Sets or queries the value of the broadcast command received bit (BCR bit, bit 15) in a status word for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BUSY	Sets or queries the value of the busy bit (BUSY bit, bit 16) in a status word for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SUBSF	Sets or queries the value of the subsystem flag bit (SUBSF bit, bit 17) in a status word for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:DBCA	Sets or queries the value of the dynamic bus control acceptance bit (DBCA bit, bit 18) in a status word for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:TF	Sets or queries the value of the terminal flag bit (TF bit, bit 19) in a status word for a MIL-STD-1553 bus to triggering on.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier	Sets or queries the qualifier to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:LESSLimit	Sets or queries the lower limit to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus.
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:MORELimit	Sets or queries the upper limit to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus.
TRIGger:{A|B}:BUS:B<x>:PARallel:DATa:VALue	Specifies the binary data string used for a Parallel Bus trigger.
TRIGger:{A|B}:BUS:B<x>:RS232C:CONDition	Specifies the condition for an RS-232C trigger.
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:SIZe	Sets or queries the length of the data string in bytes to be used for an RS232 trigger.
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:VALue	Sets or queries the data value of the data token for an RS232 trigger.
TRIGger:{A|B}:BUS:B<x>:SENT:CONDition	Sets or queries the trigger condition for a SENT bus.
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType	Sets or queries the error type to be used when triggering on SENT data.
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType:CRC	Sets or queries the CRC error type to be used when triggering on SENT data.
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue	Sets or queries the high binary Fast channel 1 value to use when triggering on a SENT bus signal.
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier	Sets or queries the qualifier to be used when triggering on SENT fast packet bus data for device channel 1.
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue	Sets or queries the binary fast channel 1 value to be used when triggering on a SENT bus signal.
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue	Sets or queries the high binary fast channel 2 value to use when triggering on a SENT Fast bus signal.
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier	Sets or queries the qualifier to be used when triggering on SENT Fast packet bus data for device channel 2.
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALue	Sets or queries the binary Fast channel 2 value to be used when triggering on a SENT bus signal. The trigger condition must be set to FAST.

Table 2-46: Trigger commands (cont.)

Command	Description
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:HIVALue	Sets or queries the high binary Fast message counter value to be used when triggering on a SENT bus signal.
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:QUALifier	Sets or queries the qualifier to be used when triggering on SENT Fast packet bus data for the secure format counter.
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue	Sets or queries the binary fast message counter value to be used when triggering on a SENT bus signal.
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue	Sets or queries the binary fast message inverted nibble value to be used when triggering on a SENT bus signal.
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue	Sets or queries the binary status value to be used when triggering on a SENT bus signal.
TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier	Sets or queries the qualifier to be used when triggering on SENT pause pulses.
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue	Sets or queries the high binary Slow channel data value to use when triggering on SENT Slow packet bus data.
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier	This command sets or queries the binary identifier value to use when triggering on Slow packet bus data.
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALue	This command sets or queries the binary slow channel value to use when triggering on Slow packet bus data.
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDentifier:VALue	This command sets or queries the qualifier to use when triggering on SENT Slow packet bus data.
TRIGger:{A|B}:BUS:B<x>:SPI:CONDition	Specifies the trigger condition for a SPI trigger.
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:SIZe	Specifies the length of the data string to be used for a SPI trigger if the trigger condition is DATa.
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue	Specifies the binary data string used for SPI triggering if the trigger condition is DATA.
TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition	Sets or queries the trigger condition for the specified SPMI bus.
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:SIZe	Sets or queries the length of the data string, in bytes, to be used when triggering on an SPMI bus signal.
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue	Sets or queries the binary data string used for SPMI triggering.
TRIGger:{A|B}:BUS:B<x>:SPMI:MASTERADDRess:VALue	Sets or queries the binary data string that identifies the master address used in SPMI triggering if the trigger condition is MASTERREAd, MASTERWRIte, or DEVICEDESCMASTERREAd.
TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse	Sets or queries whether or not to trigger on No Response frames.
TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue	Sets or queries the binary data string that identifies the register address used in SPMI triggering.
TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRess:VALue	Sets or queries the binary data string that identifies the slave address used in SPMI triggering
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:HIVALue	Specifies the binary address string for the upper limit for inside-of-range and outside-of-range qualifiers for the USB trigger.
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue	Specifies the binary address string to be used for USB trigger.

Table 2-46: Trigger commands (cont.)

Command	Description
TRIGger:{A|B}:BUS:B<x>:USB:CONDition	Specifies the trigger condition for the USB trigger.
TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue	Specifies the binary data string for the upper limit for inside-of-range and outside-of-range qualifiers for the USB trigger when the trigger condition is DATAPacket.
TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet	Specifies the data offset in bytes to trigger on.
TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier	Sets the qualifier to be used when triggering on a USB bus signal.
TRIGger:{A|B}:BUS:B<x>:USB:DATa:SIZe	Specifies the number of contiguous data bytes to trigger on.
TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPe	Specifies the data type for when the trigger condition is set to DATAPacket.
TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue	Specifies the binary data string to be used for the USB trigger when the trigger condition is DATAPacket.
TRIGger:{A|B}:BUS:B<x>:USB:ENDPoint:VALue	Specifies the binary endpoint string to be used for the USB trigger.
TRIGger:{A|B}:BUS:B<x>:USB:ERRType	Specifies the error type to be used when the trigger condition is set to ERRor.
TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType	Specifies the handshake type for the USB trigger.
TRIGger:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber	Specifies the binary data string to be used for start of frame number, when the trigger condition is Token Packet and the token type is Start of Frame.
TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType	Specifies the packet ID (PID) for the special packet.
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue	When triggering on a high-speed USB split transaction, this command specifies the split transaction endpoint type value to trigger on.
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:HUB:VALue	When triggering on a high-speed USB split transaction, this command specifies the split transaction hub address value to trigger on.
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue	When triggering on a high-speed USB split transaction, this command specifies the split transaction port address value to trigger on.
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue	When triggering on a high-speed USB split transaction, this command specifies whether to trigger on the start or complete phase of the split transaction, based on the Start/Complete bit field value.
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SE:VALue	When triggering on a high-speed USB split transaction, this command specifies the split transaction start/end bit value to trigger on.
TRIGger:{A|B}:BUS:B<x>:USB:TOKENType	Sets or queries the token type for the USB trigger.
TRIGger:{A|B}:BUS:SOUrce	Sets or queries the source for a bus trigger.
TRIGger:{A|B}:EDGE:COUPling	Sets or queries the type of coupling for the edge trigger.
TRIGger:{A|B}:EDGE:SLOpe	Sets or queries the slope for the edge trigger.
TRIGger:{A|B}:EDGE:SOUrce	Sets or queries the source for the edge trigger.
TRIGger:{A|B}:LEVel:CH<x>	Sets or queries the level for the trigger for the channel.
TRIGger:{A|B}:LOGIc:DELTatime	Sets or queries the Logic trigger delta time value.
TRIGger:{A|B}:LOGIc:FUNCtion	Sets or queries the logical combination of the input channels for the logic trigger.

Table 2-46: Trigger commands (cont.)

Command	Description
TRIGger:{A|B}:LOGIc:INPut:CLOCk:SOUrce	Sets or queries the channel to use as the clock source.
TRIGger:{A|B}:LOGIc:POLarity	Sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic trigger type.
TRIGger:{A|B}:LOGIc:USECLockedge	Sets whether or not Logic trigger type uses clock source.
TRIGger:{A|B}:LOGIc:WHEn	Sets or queries the condition for generating a logic trigger with respect to the defined input pattern.
TRIGger:{A|B}:LOGICPattern:{CH<x>|CH<x>_D<x>}	Sets or queries the Logic Pattern that is used along with the Define Logic choice to determine when the logic trigger occurs.
TRIGger:{A|B}:LOWerthreshold:CH<x>	Sets or queries CH<x> lower trigger level.
TRIGger:{A|B}:PULSEWidth:LOGICQUALification	Sets whether or not to use logic qualification for a pulse width trigger.
TRIGger:{A|B}:PULSEWidth:HIGHLimit	Sets or queries the upper limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values.
TRIGger:{A|B}:PULSEWidth:LOWLimit	Sets or queries the lower limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values.
TRIGger:{A|B}:PULSEWidth:POLarity	Sets or queries the polarity for a pulse width trigger.
TRIGger:{A|B}:PULSEWidth:SOUrce	Sets or queries the source waveform for a pulse width trigger.
TRIGger:{A|B}:PULSEWidth:WHEn	Sets or queries to trigger when a pulse is detected with a width (duration) that is less than, greater than, equal to, or unequal to a specified value.
TRIGger:{A|B}:RUNT:LOGICQUALification	Sets or queries whether or not to use logic qualification for a runt trigger.
TRIGger:{A|B}:RUNT:POLarity	Sets or queries the polarity for the runt trigger.
TRIGger:{A|B}:RUNT:SOUrce	Sets or queries the source waveform for the runt trigger.
TRIGger:{A|B}:RUNT:WHEn	Sets or queries the type of pulse width the trigger checks for when it detects a runt.
TRIGger:{A|B}:RUNT:WIDth	Sets or queries the width, in seconds, for a runt trigger.
TRIGger:{A|B}:SETHold:CLOCk:EDGE	Sets or queries the clock edge polarity for setup and hold triggering.
TRIGger:{A|B}:SETHold:CLOCk:SOUrce	Sets or queries the clock source for the setup and hold triggering.
TRIGger:{A|B}:SETHold:HOLDTime	Sets or queries the hold time for setup and hold violation triggering.
TRIGger:{A|B}:SETHold:SETTime	Sets or queries the setup time for setup and hold violation triggering.
TRIGger:{A|B}:SETHOLDLogicval:{CH<x>|CH<x>_D<x>}	Sets or queries whether the specified channel is included (INCLude) or not included (DONTInclude) in the Setup & Hold trigger input configuration.
TRIGger:{A|B}:TIMEOut:LOGICQUALification	Sets or queries whether or not to use logic qualification for a timeout trigger.
TRIGger:{A|B}:TIMEOut:POLarity	When triggering using the `TIMEOut` trigger type, this commands specifies the polarity to be used.
TRIGger:{A|B}:TIMEOut:SOUrce	When triggering using the `TIMEOut` trigger type, this command specifies the source.
TRIGger:{A|B}:TIMEOut:TIMe	When triggering using the `TIMEOut` trigger type, this command specifies the timeout time, in seconds.
TRIGger:{A|B}:TRANsition:DELTatime	Sets or queries the delta time used in calculating the transition value for the transition trigger.
TRIGger:{A|B}:TRANsition:POLarity	Sets or queries the polarity for the transition trigger.

Table 2-46: Trigger commands (cont.)

Command	Description
TRIGger:{A|B}:TRANsition:SOUrce	Sets or queries the source for a transition trigger.
TRIGger:{A|B}:TRANsition:WHEn	Sets or queries whether to check for a transitioning signal that is faster or slower than the specified delta time.
TRIGger:{A|B}:TRANsition:LOGICQUALification	Sets or queries whether or not to use logic qualification for a transition trigger.
TRIGger:{A|B}:TYPe	Sets or queries the type of A or B trigger.
TRIGger:{A|B}:UPPerthreshold:CH<x>	Sets or queries the CH<x> upper trigger level.
TRIGger:{A|B}:WINdow:CROSSIng	Sets or queries the window trigger threshold crossing of the selected trigger Source.
TRIGger:{A|B}:WINdow:LOGICQUALification	Specifies whether or not to use logic qualification for a window trigger.
TRIGger:{A|B}:WINdow:SOUrce	Sets or queries the source for a window trigger.
TRIGger:{A|B}:WINdow:WHEn	Sets or queries the window trigger event.
TRIGger:{A|B}:WINdow:WIDth	Sets or queries the minimum width for a window violation.
TRIGger:{A|B|B:RESET}	Sets the A, B, or B Reset trigger level automatically to 50% of the range of the minimum and maximum values of the trigger input signal.
TRIGger:A:HOLDoff:BY	Sets or queries the type of holdoff for the A trigger.
TRIGger:A:HOLDoff:TIMe	Sets or queries the A trigger holdoff time.
TRIGger:A:LOGICQUALification	Sets or queries the type of logic qualification to perform.
TRIGger:A:MODe	Sets or queries the A trigger mode.
TRIGger:AUXLevel	Sets or queries the Auxiliary Input voltage level to use for an edge trigger.
TRIGger:B:BY	Sets or queries B trigger time or event qualifiers.
TRIGger:B:EVENTS:COUNt	Sets or queries the number of events that must occur before the B trigger occurs.
TRIGger:B:RESET	Sets the B reset trigger level to 50%.
TRIGger:B:RESET:EDGE:COUPling	Sets or queries the trigger coupling for a sequential edge trigger reset when the Source is set to an analog channel.
TRIGger:B:RESET:EDGE:LEVel	Sets the voltage level to use for an Edge Reset trigger when triggering on an analog channel waveform.
TRIGger:B:RESET:EDGE:SLOpe	Sets or queries the trigger slope for a sequential edge trigger reset.
TRIGger:B:RESET:EDGE:SOUrce	Sets or queries the trigger source for the A→B sequential edge trigger reset feature.
TRIGger:B:RESET:TIMEOut:TIMe	Sets or queries the reset timer for a sequential time out trigger reset.
TRIGger:B:RESET:TYPe	Sets or queries the type of A→B sequential trigger reset.
TRIGger:B:STATE	Queries the current state of the triggering system.
TRIGger:B:TIMe	Sets or queries the B trigger delay time.
TRIGger:STATE?	Queries the current state of the triggering system.
VISual:AREA<x>:ASPEctratio	Sets or queries whether the aspect ratio of the specified Visual Trigger area is locked.
VISual:AREA<x>:FLIP:HORizontal	Flips the specified Visual Trigger area horizontally around its center point.
VISual:AREA<x>:FLIP:VERTical	Flips the specified Visual Trigger area vertically around its center point.
VISual:AREA<x>:HEIGht	Sets or queries the height of the specified Visual Trigger area.
VISual:AREA<x>:HITType	Sets or queries the area hit logic true condition for the specified Visual Trigger area.

Table 2-46: Trigger commands (cont.)

Command	Description
VISual:AREA<x>:RESET	Sets the specified Visual Trigger area shape to a default-sized triangle.
VISual:AREA<x>:ROTAtion	Sets or queries the rotation angle of the specified Visual Trigger area.
VISual:AREA<x>:SHAPE	Sets or queries the current shape of the area.
VISual:AREA<x>:SOUrce	Sets or queries the signal source for the specified Visual Trigger area.
VISual:AREA<x>:VERTICES	Sets or queries the X and Y vertex coordinate values for all vertices of the specified Visual Trigger area.
VISual:AREA<x>:WIDTH	Sets or queries the width of the specified Visual Trigger area.
VISual:AREA<x>:XPOSition	Sets or queries the horizontal (X-axis) center position of the specified Visual Trigger area.
VISual:AREA<x>:YPOSition	Sets or queries the vertical (Y-axis) center position of the specified Visual Trigger area.
VISual:DELETEALL	Deletes all Visual Trigger areas.
VISual:ENAble	Sets or queries the status (on or off) of the Visual Trigger mode.
VISual:EQUation	Sets or queries the Visual Trigger area combination logic equation.
VISual:SHOWAReas	Shows or hides all Visual Trigger areas.
VISual:SHOWCRiteria	Sets or queries display of the area names and hit criteria for all visual trigger areas.
VISual:SHOWEQuation	Shows or hides the Visual Trigger area combination logic equation.

Vertical command group

Use the commands in the Vertical Command Group to control the vertical setup of all live (channel) waveforms for acquisition and to control the display of channel, reference, and math waveforms. Analog channels are available when analog probes are attached to FlexChannel inputs.

You can replace VOLts with SCAle in the vertical commands. This provides program compatibility with earlier models of Tektronix instruments.

Table 2-47: Vertical commands

Command	Description
CH<x>?	Returns vertical parameters for the specified channel.
CH<x>:BANdwidth	Sets or queries the bandwidth of the specified channel.
CH<x>:BANdwidth:FILTer:OPTIMIZation	Sets or queries the channel filter shape.
CH<x>:CLIPping?	Queries whether the specified channel's input signal is clipping (exceeding) the channel vertical scale setting.
CH<x>:COUPling	Sets or queries the coupling setting for the specified channel.
CH<x>:DESKew	Sets or queries the deskew time for the specified channel.
CH<x>:LABel:COLor	Sets or queries the color of the specified channel label.
CH<x>:LABel:FONT:BOLD	Sets or queries the bold state of the specified channel label.
CH<x>:LABel:FONT:ITALic	Sets or queries the italic state of the specified channel label.
CH<x>:LABel:FONT:SIZE	Sets or queries the font size of the specified channel label.
CH<x>:LABel:FONT:TYPE	Sets or queries the font type of the specified channel label.
CH<x>:LABel:FONT:UNDERline	Sets or queries the underline state of the specified channel label.
CH<x>:LABel:NAMe	Defines or queries the label for the channel waveform.
CH<x>:LABel:XPOS	Sets or queries the X display coordinate for the channel waveform label.
CH<x>:LABel:YPOS	Sets or queries the Y display coordinate for the channel waveform label.
CH<x>:OFFSet	Sets or queries the channel offset.
CH<x>:POSition	Sets or queries the vertical position for the specified analog channel.
CH<x>:PRObe?	Returns all information concerning the probe that is attached to the specified channel.
CH<x>:PRObe:AUTOZero	Autozeros the probe attached to the specified channel.
CH<x>:PRObe:COMPensate	Starts the compensation procedure for passive probes (TPP).
CH<x>:PRObe:DEGAUSS	Starts a degauss cycle of the probe attached to the specified channel.
CH<x>:PRObe:DEGAUSS:STATE?	Queries whether the probe attached to the specified channel requires a degauss operation.
CH<x>:PRObe:FORCEDRange	Sets the attached probe to its default range, or it queries the default range of the probe.
CH<x>:PRObe:GAIN?	Returns the gain of the probe that is attached to the specified channel.
CH<x>:PRObe:ID?	Returns the type and serial number of the probe that is attached to the specified channel.

Table 2-47: Vertical commands (cont.)

Command	Description
CH<x>:PRObe:ID:SERnumber?	Returns the serial number of the probe that is attached to the specified channel.
CH<x>:PRObe:ID:TYPe?	Returns the type of probe that is attached to the specified channel.
CH<x>:PRObe:INPUTMode	Sets or queries the input mode of the probe that is attached to the specified channel.
CH<x>:PRObe:INPUTMode:AOFFSet	Sets or queries the A mode offset value of the probe that is attached to the specified channel.
CH<x>:PRObe:INPUTMode:BOFFSet	Sets or queries the B mode offset value of the probe that is attached to the specified channel.
CH<x>:PRObe:INPUTMode:COFFSet	Sets or queries the common mode offset value of the probe that is attached to the specified channel.
CH<x>:PRObe:INPUTMode:DOFFSet	Sets or queries the differential mode offset value of the probe that is attached to the specified channel.
CH<x>:PRObe:RESistance?	Queries the resistance of the probe that is attached to the specified channel.
CH<x>:PRObe:SET	Sets or queries aspects of probe accessory user interfaces, for example probe attenuation factors.
CH<x>:PRObe:STATus?	Queries the probe unsigned integer error value.
CH<x>:PRObe:UNIts?	Returns the units of the probe that is attached to the specified channel.
CH<x>:PROBECal?	Returns the probe calibration status.
CH<x>:PROBECOntrol	Sets or queries the probe range.
CH<x>:PROBEFunc:EXTAtten	Sets the attenuation value for the specified channel to the specified scale factor. Or queries the user-specified attenuation.
CH<x>:PROBEFunc:EXTDBatten	Sets the attenuation value for the specified channel to the specified value in decibels. Or queries the user-specified attenuation in decibels.
CH<x>:PROBEFunc:EXTUnits	Sets or queries the alternate unit for the external attenuator of the specified channel.
CH<x>:PROBEFunc:EXTUnits:STATE	Sets or queries the custom units enable state for the specified channel.
CH<x>:SCAle	Sets or returns the vertical scale for the specified analog channel.
CH<x>:TERmination	Sets channel input termination.
CH<x>:VTERm:BIAS	Sets or queries the termination voltage for the specified channel (if control is available).
CONFIGuration:ANALOg:BANDWidth?	Queries the maximum licensed bandwidth of the instrument.
REF:ADDNew	Adds the specified reference. Argument is of the form "REF<NR1> ", where NR1 ≥ 1.
REF:DELete	Deletes the specified reference.
REF:REF<x>:DESKew	Sets or queries the deskew time for the specified reference.
REF:REF<x>:LABel:COLor	Sets or queries the color of the specified ref label.
REF:REF<x>:LABel:FONT:BOLD	Sets or queries the bold state of the specified reference label.
REF:REF<x>:LABel:FONT:ITALic	Sets or queries the italic state of the specified reference label.

Table 2-47: Vertical commands (cont.)

Command	Description
REF:REF<x>:LABel:FONT:SIZE	Sets or queries the font size of the specified reference label.
REF:REF<x>:LABel:FONT:TYPE	Sets or queries the font type of the specified reference label.
REF:REF<x>:LABel:FONT:UNDERline	Sets or queries the underline state of the specified reference label.
REF:REF<x>:LABel:NAMe	Sets or queries the label of the designated waveform.
REF:REF<x>:LABel:XPOS	Sets or queries the position of the reference waveform label on the X axis.
REF:REF<x>:LABel:YPOS	Sets or queries the position of the reference waveform label on the Y axis.
REF:LIST?	Lists all currently defined references.
REF:REF<x>:SOUrce	Sets or queries the filename used by the given reference.
VERTical:DESKew:FROM:CUSTOMPROPAgation	Sets or queries a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.
VERTical:DESKEW:STATIC	Sets or queries the target channel for performing channel-to-channel deskew adjustment.
VERTical:DESKew:TO:CUSTOMPROPAgation	Sets or queries a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.
VERTical:DESKEW:FROM:SOUrce	Sets or queries the source channel for performing channel-to-channel deskew adjustment.
VERTical:DESKEW:TO:SOUrce	Sets or queries the target channel for performing channel-to-channel deskew adjustment.

Waveform Transfer command group

Use the commands in the Waveform Transfer Command Group to transfer waveform data points from the instrument. Waveform data points are a collection of values that define a waveform. One data value usually represents one data point in the waveform record. When working with envelope waveforms, each data value is either the minimum or maximum of a min/max pair.

Before you transfer waveform data, you must specify the data format, record length, and waveform source.

Data formats. Acquired waveform data uses eight or more bits to represent each data point. The number of bits used depends on the acquisition mode specified when you acquired the data. Data acquired in SAMple or ENVelope mode uses eight bits per waveform data point. Data acquired in AVERage mode uses up to 14 bits per point.

The instrument can transfer waveform data in either ASCII or binary format. You specify the format with the DATa:ENCdg command. The instrument uses signed, 4 byte integers and floating point values; it does not support unsigned floating point values.

ASCII data is represented by signed integer or floating point values. An example ASCII waveform data string can look like this:

CURVE<space>-110,-109,-110,-110,-109,-107,
-109,-107,-106,-105,-103,-100,-97,-90,-84,-80

Use ASCII to obtain more readable and easier to format output than binary. However, ASCII can require more bytes to send the same values than it does with binary. This can reduce transmission speeds.

Binary data can be represented by signed integer or floating point values. The range of the values depends on the byte width specified. When the byte width is one, signed integer data ranges from -128 to 127, and positive integer values range from 0 to 255. When the byte width is two, the values range from -32768 to 32767. When a MATH (or REF that came from a MATH) is used, 32-bit floating point values are used that are four bytes in width.

The defined binary formats specify the order in which the bytes are transferred. The following are the four binary formats:

- RIBinary specifies signed integer data-point representation with the most significant byte transferred first.
- SRIBinary is the same as RIBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.
- RFBinary specifies floating point data-point representation with the most significant byte transferred first.
- SRFBinary is the same as RFBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.

Waveform data and record lengths. You can transfer multiple points for each waveform record. You can transfer a portion of the waveform or you can transfer the entire record. You can use the `DATa:STARt` and `DATa:STOP` commands to specify the first and last data points of the waveform record.

When transferring data from the instrument, you must specify the first and last data points in the waveform record. Setting `DATa:STARt` to 1 and `DATa:STOP` to the record length will always return the entire waveform.

Waveform data locations and memory allocation. The DATa:SOUrce command specifies the waveform source when transferring a waveform from the instrument.

Waveform preamble. Each waveform that you transfer has an associated waveform preamble that contains information such as the horizontal scale, the vertical scale, and other settings in effect when the waveform was created. Refer to the individual WFMOutpre? commands for more information.

Scaling waveform data. Once you transfer the waveform data to the controller, you can convert the data points into voltage values for analysis using information from the waveform preamble.

Transferring waveform data from the instrument.

You can transfer waveforms from the instrument to an external controller using the following sequence:

1. Select the waveform source(s) using `DATa:SOUrce`.
2. Specify the waveform data format using `DATa:ENCdg`.
3. Specify the number of bytes per data point using `WFMOutpre:BYT_Nr`.

NOTE. *MATH waveforms (and REF waveforms that came from a MATH) are always set to four bytes.*

4. Specify the portion of the waveform that you want to transfer using `DATa:STARt` and `DATa:STOP`.
5. Transfer waveform preamble information using `WFMOutpre`.
6. Transfer waveform data from the instrument using `CURVe?`.

Table 2-48: Waveform Transfer commands

Command	Description
CURVe	The query format transfers waveform data from instrument specified by the `DATa:SOUrce` command.
DATa	Sets the format and location of the waveform data that is transferred with the `CURVe` Command. Or queries the format and location of the waveform data that is transferred with the `CURVe?` command.
DATa:ENCdg	Sets or queries the format of outgoing waveform data.
DATa:RESample	Sets or queries the resampling of outgoing waveform data.
DATa:SOUrce	Sets or queries the location of waveform data transferred from the instrument.
DATa:SOUrce:AVAILable?	Returns a list of enumerations representing the source waveforms that are currently available for :CURVe? queries.
DATa:STARt	Sets or queries the starting point in waveform transfer.
DATa:STOP	Sets or queries the ending data point in waveform transfer.
DATa:WIDth	Specifies the width, in bytes per point, for waveform data transferred from the instrument via the CURVe? query.
WAVFrm?	Returns `WFMOutpre?` and `CURVe?` data for the waveform as specified by the `DATA:SOUrce` command.
WFMOutpre?	Returns the waveform formatting data for the waveform specified by the DATA:SOURCE command.
WFMOutpre:ASC_Fmt?	Returns the format for ASCII data transferred from the instrument.
WFMOutpre:BIT_Nr	Sets or queries the number of bits per waveform point that outgoing waveforms contain.

Table 2-48: Waveform Transfer commands (cont.)

Command	Description
WFMOutpre:BN_Fmt	Sets or queries the format of binary data for the waveform.
WFMOutpre:BYT_Nr	Sets or queries the data width for the waveform.
WFMOutpre:BYT_Or	Sets or queries the byte order of waveform points.
WFMOutpre:CENTERFREQuency?	Returns the center frequency of the outgoing waveform.
WFMOutpre:DOMain?	Returns the domain of the outgoing waveform.
WFMOutpre:ENCdg	Sets or queries the type of encoding for outgoing waveforms.
WFMOutpre:NR_Pt?	Returns the number of points for the waveform transmitted in response to a CURVe? query.
WFMOutpre:PT_Fmt?	Returns the point format for the waveform.
WFMOutpre:PT_Off?	Returns the trigger point relative to DATA:START for the waveform.
WFMOutpre:RESample	Sets or queries the resampling of outgoing waveform data.
WFMOutpre:SPAN?	Returns the frequency span of the outgoing waveform.
WFMOutpre:WFId?	Returns a string describing the acquisition parameters for the waveform.
WFMOutpre:WFMTYPe?	Returns the type of the outgoing waveform.
WFMOutpre:XINcr?	Returns the horizontal sampling interval.
WFMOutpre:XUNit?	Returns the horizontal units for the waveform.
WFMOutpre:XZEro?	Returns the (sub-sample) time between the trigger sample and the occurrence of the actual waveform trigger.
WFMOutpre:YOFf?	Returns the vertical offset in digitizing levels for the waveform.
WFMOutpre:YUNit?	Returns the vertical units for the waveform.
WFMOutpre:YZEro?	Returns the vertical offset for the waveform.

Zoom command group

Zoom commands let you expand and position the waveform display horizontally and vertically, without changing the time base or vertical settings.

NOTE. *Zoom commands are available once a view has been added.*

Table 2-49: Zoom commands

Command	Description
DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM	Sets or queries the start of the zoom x axis in the specified plot view.
DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO	Sets or queries the end of the zoom x axis in the specified plot view.
DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM	Sets or queries the start of the zoom y axis in the specified plot view.
DISplay:MATHFFTView<x>:ZOOM:YAXIS:TO	Sets or queries the end of the zoom y axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:XAXIS:FROM	Sets or queries the start of the zoom x-axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:XAXIS:TO	Sets or queries the end of the zoom x-axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:YAXIS:FROM	Sets or queries the start of the zoom y-axis in the specified plot view.
DISplay:PLOTView<x>:ZOOM:YAXIS:TO	Sets or queries the end of the zoom y-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM	Sets or returns the left edge of the zoom x-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO	Sets or queries the right edge of the zoom x-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM	Sets or queries the bottom value of the zoom y-axis in the specified plot view.
DISplay:REFFFTView<x>:ZOOM:YAXIS:TO	Sets or queries the top value of the zoom y-axis in the specified plot view.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition	Sets or queries the horizontal zoom position (of the specified zoom in the specified Waveform View) of the zoomed waveform or zoom waveform in the display, around which the zoom waveform displays.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:SCALe	Sets or queries the zoom horizontal scale factor, of the specified zoom in the specified Waveform View, in which the zoom waveform is displayed.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALe	Sets or queries the zoom window horizontal scale in the specified Waveform View.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe	Sets or queries the specified zoom in the specified Waveform View on or off.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition	Sets or queries the vertical position of the specified zoom in the specified Waveform View, in which the zoom waveform is displayed.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALe	Sets or queries the zoom vertical scale factor of the specified waveform for the specified zoom in the specified Waveform View.
DISplay:WAVEView<x>:Zoom?	Queries the zoom parameters of the specified Waveform View.
DISplay:WAVEView<x>:ZOOM:ZOOM<x>?	Queries the zoom parameters of the specified zoom in the specified Waveform View.

Commands listed in alphabetical order

NOTE. Some of the following commands may not be available on your instrument model. Also, some of the following commands are only available if your instrument has the associated option installed.

ACQuire? (Query Only)

Queries the current acquisition state.

Group Acquisition

Syntax `ACQuire?`

Examples `ACQUIRE?` might return `:ACQUIRE:STATE 1;NUMENV INFINITE;STOPAFTER RUNSTOP;SEQUENCE:MODE NUMACQS;NUMSEQUENCE 1;:ACQUIRE:NUMAVG 16;FASTACQ:STATE 0;PALETTE TEMPERATURE;:ACQUIRE:MODE SAMPLE;SAMPLINGMODE IT`.

ACQuire:FASTAcq:PALEtte

Sets or queries the waveform grading for fast acquisition mode.

Group Acquisition

Syntax
```
ACQuire:FASTAcq:PALEtte
{NORMal|TEMPerature|SPECtral|INVErted}
ACQuire:FASTAcq:PALEtte?
```

Arguments `NORMal` colors traces according to their channel.

`TEMPerature` colors all traces using a multicolored palette, where “intensity” is represented by hue; blue for least frequently hit, red for most frequently hit. All traces share this palette. This is the default color palette.

`SPECtral` colors all traces using a multicolored palette, where “intensity” is represented by hue; red for least frequently hit, blue for most frequently hit. All traces share this palette.

`INVErted` Inverts the normal display hues and lightness levels based on sample intensity. The areas of lowest sample density appear the brightest, while the areas with the highest sample density appear the darkest.

Examples `ACQUIRE:FASTACQ:PALETTE TEMPerature` sets the acquisition fast acquisition palette to Temperature.

`ACQUIRE:FASTACQ:PALETTE?` might return `:ACQUIRE:FASTACQ:PALETTE TEMPERATURE`.

ACQuire:FASTAcq:STATE

Sets or queries the state of fast acquisition mode.

Group Acquisition

Syntax
```
ACQuire:FASTAcq:STATE {<NR1>|OFF|ON}
ACQuire:FASTAcq:STATE?
```

Arguments `<NR1>` = 0 disables FASTAcq; any other value turns this feature on.

`OFF` disables the FASTAcq feature.

`ON` enables the FASTAcq feature.

Examples `ACQUIRE:FASTACQ:STATE ON` turns on fast acquisition mode.

`ACQUIRE:FASTACQ:STATE?` might return `:ACQUIRE:FASTACQ:STATE 0` indicating that fast acquisition mode is off.

ACQuire:MAXSamplerate? (Query Only)

This query returns the maximum real-time sample rate.

Group Acquisition

Syntax `ACQuire:MAXSamplerate?`

Examples `ACQUIRE:MAXSAMPLERATE?` might return `:ACQUIRE:MAXSAMPLERATE 6.25E9`.

ACQuire:MODe

This command sets or queries the selected acquisition mode of the instrument.

Group Acquisition

Syntax

```
ACQuire:MODe {SAMple|PEAKdetect|HIRes|AVErage|ENVelope}
ACQuire:MODe?
```

Related Commands ACQuire:NUMAVg

CURVe

Arguments `SAMple` specifies that the displayed data point value is the first sampled value that is taken during the acquisition interval. In sample mode, all waveform data has 8 bits of precision. You can request 16 bit data with a CURVe query but the lower-order 8 bits of data will be zero. `SAMple` is the default mode.

`PEAKdetect` specifies the display of high-low range of the samples taken from a single waveform acquisition. The high-low range is displayed as a vertical column that extends from the highest to the lowest value sampled during the acquisition interval. `PEAKdetect` mode can reveal the presence of aliasing or narrow spikes.

`HIRes` specifies Hi Res mode where the displayed data point value is the average of all the samples taken during the acquisition interval. This is a form of averaging, where the average comes from a single waveform acquisition. The number of samples taken during the acquisition interval determines the number of data values that compose the average.

`AVErage` specifies averaging mode, in which the resulting waveform shows an average of `SAMple` data points from several separate waveform acquisitions. The instrument processes the number of waveforms you specify into the acquired waveform, creating a running exponential average of the input signal. The number of waveform acquisitions that go into making up the average waveform is set or queried using the ACQuire:NUMAVg command.

`ENVelope` specifies envelope mode, where the resulting waveform shows the `PEAKdetect` range of data points from several separate waveform acquisitions. The number of waveform acquisitions that go into making up the envelope waveform is set or queried using the ACQuire:NUMAVg command.

The instrument acquires data after each trigger event using Sample mode; it then determines the pix map location of each sample point and accumulates it with stored data from previous acquisitions.

A Pix map is a two dimensional array. The value at each point in the array is a counter that reflects the hit intensity. Infinite and noninfinite persist display modes affect how pix maps are accumulated. Zoom, Math, FastAcq, FastFrame, XY, Roll, and Interpolated Time (IT) Sampling Mode are conflicting features to WFMDB acqMode. Turning on one of them generally turns the other one off. Selection of some standard masks (for example, eye masks, which require option MTM) changes the acquisition mode to WFMDB.

Examples

`ACQUIRE:MODE ENVELOPE` sets the acquisition mode to display a waveform that is an envelope of many individual waveform acquisitions.

`ACQUIRE:MODE?` might return `:ACQuire:MODe AVERAGE`, indicating that the displayed waveform is the average of the specified number of waveform acquisitions.

ACQuire:NUMACq? (Query Only)

This query-only command returns the number of waveform acquisitions that have occurred since the last time acquisitions were stopped.

Group

Acquisition

Syntax

```
ACQuire:NUMACq?
```

Related Commands

ACQuire:STATE

Examples

`ACQUIRE:NUMACQ?` might return `:ACQUIRE:NUMACQ 350`, indicating that 350 acquisitions have occurred since executing an ACQuire:STATE RUN command.

ACQuire:NUMAVg

This command sets or queries the number of waveform acquisitions that make up an averaged waveform. Ranges from 2 to 10240.

Group

Acquisition

Syntax

```
ACQuire:NUMAVg <NR1>
ACQuire:NUMAVg?
```

Related Commands

ACQuire:MODe

Arguments

`<NR1>` is the number of waveform acquisitions to average.

Examples

`ACQUIRE:NUMAVG 10` specifies that 10 waveform averages will be performed before exponential averaging starts.

`ACQUIRE:NUMAVG?` might return `:ACQUIRE:NUMAVG 75`, indicating that there are 75 acquisitions specified for averaging.

ACQuire:NUMFRAMESACQuired? (Query Only)

This query returns the number of FastFrame frames which have been acquired.

Group Horizontal

Syntax `ACQuire:NUMFRAMESACQuired?`

Examples `ACQUIRE:NUMFRAMESACQUIRED?` might return `:ACQUIRE:NUMFRAMESACQUIRED 4` indicating 4 frames have been acquired.

ACQuire:SEQuence:CURrent? (Query Only)

In single sequence acquisition mode, this query returns the number of acquisitions or measurements in the sequence completed so far.

Group Acquisition

Syntax `ACQuire:SEQuence:CURrent?`

Examples `ACQUIRE:SEQUENCE:CURRENT?` might return `:ACQUIRE:SEQUENCE:CURRENT 5`.

ACQuire:SEQuence:MODe

In single sequence acquisition, the single sequence stop after count is based on the number of acquisitions.

Group Acquisition

Syntax `ACQuire:SEQuence:MODe NUMACQs`

Arguments `NUMACQs` is the number of acquisitions.

Examples `ACQUIRE:SEQUENCE:MODE?` might return `:ACQUIRE:SEQUENCE:MODE NUMACQS` indicating the acquisition sequence mode is set to `NUMACQS`.

ACQuire:SEQuence:NUMSEQuence

In single sequence acquisition mode, specify the number of acquisitions or measurements that comprise the sequence. The default is 1.

Group Acquisition

Syntax

```
ACQuire:SEQuence:NUMSEQuence <NR1>
ACQuire:SEQuence:NUMSEQuence?
```

Arguments <NR1> is the number of acquisitions or measurements that comprise the sequence.

Examples ACQUIRE:SEQUENCE:NUMSEQUENCE 2 sets the number of acquisition in a sequence is set to 2.

ACQUIRE:SEQUENCE:NUMSEQUENCE? might return ACQUIRE:SEQUENCE:NUMSEQUENCE 1, indicating that the number of acquisition in a sequence is set to 1.

ACQuire:STATE

This command starts or stops acquisitions. When state is set to ON or RUN, a new acquisition will be started. If the last acquisition was a single acquisition sequence, a new single sequence acquisition will be started. If the last acquisition was continuous, a new continuous acquisition will be started.

If RUN is issued in the middle of completing a single sequence acquisition (for example, averaging or enveloping), the acquisition sequence is restarted, and any accumulated data is discarded. Also, the instrument resets the number of acquisitions. If the RUN argument is issued while in continuous mode, a reset occurs and acquired data continues to acquire.

If acquire:stopafter is SEQUENCE, this command leaves the instrument in single sequence, unlike the run/stop button which takes the instrument out of single sequence.

Group Acquisition

Syntax

```
ACQuire:STATE {<NR1>|OFF|ON|RUN|STOP}
ACQuire:STATE?
```

Related Commands ACQuire:STOPAfter

Arguments

<NR1> = 0 stops acquisitions; any other value starts acquisitions.

OFF stops acquisitions.

ON starts acquisitions.

RUN starts acquisitions.

STOP stops acquisitions.

Examples

ACQUIRE:STATE RUN starts the acquisition of waveform data and resets the count of the number of acquisitions.

ACQUIRE:STATE? might return :ACQUIRE:STATE 0, indicating that the acquisition is stopped.

ACQuire:STOPAfter

This command sets or queries whether the instrument continually acquires acquisitions or acquires a single sequence. Pressing SINGLE on the front panel button is equivalent to sending these commands: ACQUIRE:STOPAFTER SEQUENCE and ACQUIRE:STATE 1.

Group

Acquisition

Syntax

```
ACQuire:STOPAfter {RUNSTop|SEQuence}
ACQuire:STOPAfter?
```

Related Commands

ACQuire:STATE

Arguments

RUNSTop specifies that the instrument will continually acquire data, if ACQuire:STATE is turned on.

SEQuence specifies that the next acquisition will be a single-sequence acquisition.

Examples

ACQUIRE:STOPAFTER RUNSTOP sets the instrument to continually acquire data.

ACQUIRE:STOPAFTER? might return :ACQUIRE:STOPAFTER SEQUENCE, indicating that the next acquisition the instrument makes will be of the single-sequence type.

ACTONEVent:ENable

This command enables or disables actions on event (AOE). If AOE saves are limited and the limit has been reache

Group Act On Event

Syntax

```
ACTONEVent:ENable <NR1>
ACTONEVent:ENable?
```

Arguments <NR1> is a number that enables or disables actions on event. The number zero disables the feature and the number one enables the feature.

Examples ACTONEVent:ENable 1 enables actions on event.

ACTONEVent:ENable? might return :ACTONEVent:ENable 0, indicating the actions on event feature is disabled.

ACTONEVent:LIMit

This command sets whether the act on event should limit the number of saves. This prevents the saves from filling the hard drive.

Group Act On Event

Syntax

```
ACTONEVent:LIMit <NR1>
ACTONEVent:LIMit?
```

Arguments <NR1> is a number that enables or disables whether the act on event should limit the number of saves. The number zero disables the feature and the number one enables the feature.

Examples ACTONEVent:LIMit 1 enables the limiting of act on event saves.

ACTONEVent:LIMit? might return :ACTONEVent:LIMit 0, indicating the limiting of act on event saves feature is disabled.

ACTONEVent:LIMITCount

This command sets the limit of act on even saves. The ACTONEVent:LIMit command must be enabled.

Group Act On Event

Syntax
```
ACTONEVent:LIMITCount <NR1>
ACTONEVent:LIMITCount?
```

Arguments <NR1> sets the number of allowed saves for the act on event feature. The number must be a positive integer.

Examples ACTONEVent:LIMITCount 2 sets the number of saves to 2.

ACTONEVent:LIMITCount? might return :ACTONEVent:LIMITCount 4, indicating the limit of act on event saves is 4.

ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE

This command saves a screen capture when a mask test fails.

Group Act On Event

Syntax
```
ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE?
```

Related Commands SAVEONEVent:IMAGe:FILEFormat

SAVEONEVent:FILEName

SAVEONEVent:WAVEform:FILEFormat

SAVEONEVent:WAVEform:SOUrce

Arguments ON enables the save screen capture when a mask test fails feature.

OFF disables the save screen capture when a mask test fails feature.

<NR1> is a number that enables or disables the save screen capture when a mask test fails feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE ON` sets the state of the save screen capture feature to on.

`ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE?` might return `:ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE OFF`, indicating the state of the save screen capture feature is set to off.

ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a mask test fails.

Group

Act On Event

Syntax

```
ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat

SAVEONEVent:FILEName

SAVEONEVent:WAVEform:FILEFormat

SAVEONEVent:WAVEform:SOUrce

Arguments

`ON` enables the save source waveform(s) when a mask test fails feature.

`OFF` disables the save source waveform(s) when a mask test fails feature.

`<NR1>` is a number that enables or disables the save source waveform(s) when a mask test fails feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE ON` sets the state of the save source waveform feature to on.

`ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE?` might return `:ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE OFF`, indicating the state of the save source waveform feature is set to off.

ACTONEVent:MASKFail:ACTION:SRQ:STATE

This command generates an SRQ event when a mask fails.

Group

Act On Event

Syntax

```
ACTONEVent:MASKFail:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKFail:ACTION:SRQ:STATE?
```

Related Commands ACTONEVent:ENable

Arguments

ON enables the generate an SRQ event when a mask fails feature.

OFF disables the generate an SRQ event when a mask fails feature.

<NR1> is a number that enables or disables the generate an SRQ event when a mask fails feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MASKFail:ACTION:SRQ:STATE 1 enables the generate an SRQ event when a mask fails feature to enabled.

ACTONEVent:MASKFail:ACTION:SRQ:STATE? might return :ACTONEVent:MASKFail:ACTION:SRQ:STATE 0, indicating the state of the generate an SRQ event when a mask fails feature is disabled.

ACTONEVent:MASKFail:ACTION:STOPACQ:STATE

This command stops acquisitions when a mask test fails.

Group Act On Event

Syntax

```
ACTONEVent:MASKFail:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKFail:ACTION:STOPACQ:STATE?
```

Related Commands ACTONEVent:ENable

Arguments

ON enables the stop acquisitions when a mask test fails feature.

OFF disables the stop acquisitions when a mask test fails feature.

<NR1> is a number that enables or disables the stop acquisitions when a mask test fails feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MASKFail:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

ACTONEVent:MASKFail:ACTION:STOPACQ:STATE? might return :ACTONEVent:MASKFail:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE

This command saves a screen capture when a mask hit occurs.

Group Act On Event

Syntax

```
ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE?
```

Related Commands SAVEONEVent:IMAGe:FILEFormat

SAVEONEVent:FILEName

SAVEONEVent:WAVEform:FILEFormat

SAVEONEVent:WAVEform:SOUrce

Arguments ON enables the save screen capture when a mask hit occurs feature.

OFF disables the save screen capture when a mask hit occurs feature.

<NR1> is a number that enables or disables the save screen capture when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature to on.

ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE? might return :ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a mask hit occurs.

Group Act On Event

Syntax

```
ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat

SAVEONEVent:FILEName

SAVEONEVent:WAVEform:FILEFormat

SAVEONEVent:WAVEform:SOUrce

Arguments

`ON` enables the save source waveform(s) when a mask hit occurs feature.

`OFF` disables the save source waveform(s) when a mask hit occurs feature.

`<NR1>` is a number that enables or disables the save source waveform(s) when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE ON` sets the state of the save source waveform feature to on.

`ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE?` might return `:ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE OFF`, indicating the state of the save source waveform feature is set to off.

ACTONEVent:MASKHit:ACTION:SRQ:STATE

This command generates an SRQ event when a mask hit occurs.

Group

Act On Event

Syntax

```
ACTONEVent:MASKHit:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKHit:ACTION:SRQ:STATE?
```

Related Commands

ACTONEVent:ENable

Arguments

`ON` enables the generate an SRQ event when a mask hit occurs feature.

`OFF` disables the generate an SRQ event when a mask hit occurs feature.

`<NR1>` is a number that enables or disables the generate an SRQ event when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKHit:ACTION:SRQ:STATE 1` enables the generate an SRQ event when a mask hit occurs feature.

`ACTONEVent:MASKHit:ACTION:SRQ:STATE?` might return `:ACTONEVent:MASKHit:ACTION:SRQ:STATE 0`, indicating the generate an SRQ event when a mask hit occurs feature is disabled.

ACTONEVent:MASKHit:ACTION:STOPACQ:STATE

This command stops acquisitions when a mask hit occurs.

Group

Act On Event

Syntax

```
ACTONEVent:MASKHit:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKHit:ACTION:STOPACQ:STATE?
```

Related Commands

ACTONEVent:ENable

Arguments

`ON` enables the stop acquisitions when a mask hit occurs feature.

`OFF` disables the stop acquisitions when a mask hit occurs feature.

`<NR1>` is a number that enables or disables the stop acquisitions when a mask hit occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKHit:ACTION:STOPACQ:STATE ON` sets the state of the stop acquisitions feature to on.

`ACTONEVent:MASKHit:ACTION:STOPACQ:STATE?` might return `:ACTONEVent:MASKHit:ACTION:STOPACQ:STATE OFF`, indicating the state of the stop acquisitions feature is set to off.

ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE

This command saves a screen capture when a mask test passes.

Group

Act On Event

Syntax

```
ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat

SAVEONEVent:FILEName

SAVEONEVent:WAVEform:FILEFormat

SAVEONEVent:WAVEform:SOUrce

Arguments

ON enables the save screen capture when a mask test passes feature.

OFF disables the save screen capture when a mask test passes feature.

<NR1> is a number that enables or disables the save screen capture when a mask test passes feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature to on.

ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE? might return :ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a mask test passes.

Group

Act On Event

Syntax

```
ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat

SAVEONEVent:FILEName

SAVEONEVent:WAVEform:FILEFormat

SAVEONEVent:WAVEform:SOUrce

Arguments

ON enables the save source waveform(s) when a mask test passes feature.

OFF disables the save source waveform(s) when a mask test passes feature.

<NR1> is a number that enables or disables the save source waveform(s) when a mask test passes feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE ON` sets the state of the save source waveform feature to on.

`ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE?` might return `:ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE OFF`, indicating the state of the save source waveform feature is set to off.

ACTONEVent:MASKPass:ACTION:SRQ:STATE

This command generates an SRQ event when a mask passes.

Group

Act On Event

Syntax

```
ACTONEVent:MASKPass:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKPass:ACTION:SRQ:STATE?
```

Related Commands

ACTONEVent:ENable

Arguments

`ON` enables the generate an SRQ event when a mask passes feature.

`OFF` disables the generate an SRQ event when a mask passes feature.

`<NR1>` is a number that enables or disables the generate an SRQ event when a mask passes feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MASKPass:ACTION:SRQ:STATE 1` enables the generate an SRQ event when a mask passes feature.

`ACTONEVent:MASKPass:ACTION:SRQ:STATE?` might return `:ACTONEVent:MASKPass:ACTION:SRQ:STATE 0`, indicating the generate an SRQ event when a mask passes feature is disabled.

ACTONEVent:MASKPass:ACTION:STOPACQ:STATE

This command stops acquisitions when a mask test passes.

Group

Act On Event

Syntax

```
ACTONEVent:MASKPass:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MASKPass:ACTION:STOPACQ:STATE?
```

Related Commands ACTONEVent:ENable

Arguments ON enables the stop acquisitions when a mask test passes feature.

OFF disables the stop acquisitions when a mask test passes feature.

<NR1> is a number that enables or disables the stop acquisitions when a mask test passes feature. The number zero disables the feature, all other numbers enable the feature.

Examples ACTONEVent:MASKPass:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

ACTONEVent:MASKPass:ACTION:STOPACQ:STATE? might return :ACTONEVent:MASKPass:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE

This command saves a screen capture when the user-set measurement limit is exceeded.

Group Act On Event

Syntax
```
ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE?
```

Related Commands SAVEONEVent:IMAGe:FILEFormat

SAVEONEVent:FILEName

SAVEONEVent:WAVEform:FILEFormat

SAVEONEVent:WAVEform:SOUrce

Arguments ON enables the save screen capture when the measurement limit is exceeded feature.

OFF disables the save screen capture when the measurement limit is exceeded feature.

<NR1> is a number that enables or disables the save screen capture when the measurement limit is exceeded feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature to on.

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE? might return :ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when the user-set measurement limit is exceeded.

Group

Act On Event

Syntax

```
ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE
{ON|OFF|<NR1>}
ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat

SAVEONEVent:FILEName

SAVEONEVent:WAVEform:FILEFormat

SAVEONEVent:WAVEform:SOUrce

Arguments

ON enables the save source waveform(s) when the measurement limit is exceeded feature.

OFF disables the save source waveform(s) when the measurement limit is exceeded feature.

<NR1> is a number that enables or disables the save source waveform(s) when the measurement limit is exceeded feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE? might return :ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.

ACTONEVent:MEASUrement:ACTION:SRQ:STATE

This command generates an SRQ event when any measurement triggers the user-defined measurement limits.

Group Act On Event

Syntax

```
ACTONEVent:MEASUrement:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MEASUrement:ACTION:SRQ:STATE?
```

Related Commands ACTONEVent:ENable

Arguments

`ON` enables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature.

`OFF` disables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature.

`<NR1>` is a number that enables or disables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:MEASUrement:ACTION:SRQ:STATE 1` enables the generate an SRQ event when any measurement triggers the user-defined measurement limits feature.

`ACTONEVent:MEASUrement:ACTION:SRQ:STATE?` might return `:ACTONEVent:MEASUrement:ACTION:SRQ:STATE 0`, indicating the generate an SRQ event when any measurement triggers the user-defined measurement limits feature is disabled.

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE

This command stops acquisitions when the user-set measurement limit is exceeded.

Group Act On Event

Syntax

```
ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE?
```

Related Commands ACTONEVent:ENable

Arguments

ON enables the stop acquisitions when the measurement limit is exceeded feature.

OFF disables the stop acquisitions when the measurement limit is exceeded feature.

<NR1> is a number that enables or disables the stop acquisitions when the measurement limit is exceeded feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE? might return :ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE

This command saves a screen capture when a search event is found.

Group

Act On Event

Syntax

```
ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE?
```

Related Commands

SAVEONEVent:IMAGe:FILEFormat

SAVEONEVent:FILEName

SAVEONEVent:WAVEform:FILEFormat

SAVEONEVent:WAVEform:SOUrce

Arguments

ON enables the save screen capture when a search event is found feature.

OFF disables the save screen capture when a search event is found feature.

<NR1> is a number that enables or disables the save screen capture when a search event is found feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE ON sets the state of the save screen capture feature to on.

ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE? might return :ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE OFF, indicating the state of the save screen capture feature is set to off.

ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) when a search event is found.

Group Act On Event

Syntax

```
ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE?
```

Related Commands SAVEONEVent:IMAGe:FILEFormat

SAVEONEVent:FILEName

SAVEONEVent:WAVEform:FILEFormat

SAVEONEVent:WAVEform:SOUrce

Arguments `ON` enables the save source waveform(s) when a search event is found feature.

`OFF` disables the save source waveform(s) when a search event is found feature.

`<NR1>` is a number that enables or disables the save source waveform(s) when a search event is found feature. The number zero disables the feature, all other numbers enable the feature.

Examples `ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE ON` sets the state of the save source waveform feature to on.

`ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE?` might return `:ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE OFF`, indicating the state of the save source waveform feature is set to off.

ACTONEVent:SEARCH:ACTION:SRQ:STATE

This command generates an SRQ event when any search event is found.

Group Act On Event

Syntax

```
ACTONEVent:SEARCH:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:SEARCH:ACTION:SRQ:STATE?
```

Related Commands ACTONEVent:ENable

Arguments

ON enables the generate an SRQ event when any search event is found feature.

OFF disables the generate an SRQ event when any search event is found feature.

<NR1> is a number that enables or disables the generate an SRQ event when any search event is found feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:SEARCH:ACTION:SRQ:STATE 1 enables the generate an SRQ event when any search event is found feature.

ACTONEVent:SEARCH:ACTION:SRQ:STATE? might return :ACTONEVent:SEARCH:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when any search event is found feature is disabled.

ACTONEVent:SEARCH:ACTION:STOPACQ:STATE

This command stops acquisitions when a search event is found.

Group

Act On Event

Syntax

```
ACTONEVent:SEARCH:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:SEARCH:ACTION:STOPACQ:STATE?
```

Related Commands

ACTONEVent:ENable

Arguments

ON enables the stop acquisitions when a search event is found feature.

OFF disables the stop acquisitions when a search event is found feature.

<NR1> is a number that enables or disables the stop acquisitions when a search event is found feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:SEARCH:ACTION:STOPACQ:STATE ON sets the state of the stop acquisitions feature to on.

ACTONEVent:SEARCH:ACTION:STOPACQ:STATE? might return :ACTONEVent:SEARCH:ACTION:STOPACQ:STATE OFF, indicating the state of the stop acquisitions feature is set to off.

ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE

This command saves a screen capture on a trigger event from a single sequence or sequence of N acquisition. Each acquisition in the sequence of N will perform a save operation. This command replaces SAVe:IMAGe (still valid command, but only an alias for this new command).

Group Act On Event

Syntax

```
ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE {ON|OFF|<NR1>}
ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE?
```

Arguments

`ON` enables the save screen capture on a trigger event feature.

`OFF` disables the save screen capture on a trigger event feature.

`<NR1>` is a number that enables or disables the save screen capture on a trigger event feature. The number zero disables the feature, all other numbers enable the feature.

Examples

`ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE ON` sets the state of the save screen capture feature to on.

`ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE?` might return `:ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE OFF`, indicating the state of the save screen capture feature is set to off.

ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE

This command saves the user set source waveform(s) on a trigger event from a single sequence or sequence of N acquisition. Each acquisition in the sequence of N will perform a save operation. This command replaces SAVEON:WAVEform (still valid command, but only an alias for this new command).

Group Act On Event

Syntax

```
ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE {ON|OFF|<NR1>}
ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE?
```

Arguments

`ON` enables the save source waveform(s) on a trigger event feature.

`OFF` disables the save source waveform(s) on a trigger event feature.

<NR1> is a number that enables or disables the save source waveform(s) on a trigger event feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE ON sets the state of the save source waveform feature to on.

ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE? might return :ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE OFF, indicating the state of the save source waveform feature is set to off.

ACTONEVent:TRIGger:ACTION:SRQ:STATE

This command generates an SRQ event when a trigger event occurs.

Group

Act On Event

Syntax

```
ACTONEVent:TRIGger:ACTION:SRQ:STATE {ON|OFF|<NR1>}
ACTONEVent:TRIGger:ACTION:SRQ:STATE?
```

Related Commands

ACTONEVent:ENable

Arguments

ON enables the generate an SRQ event when a trigger event occurs feature.

OFF disables the generate an SRQ event when a trigger event occurs feature.

<NR1> is a number that enables or disables the generate an SRQ event when a trigger event occurs feature. The number zero disables the feature, all other numbers enable the feature.

Examples

ACTONEVent:TRIGger:ACTION:SRQ:STATE 1 enables the generate an SRQ event when a trigger event occurs feature.

ACTONEVent:TRIGger:ACTION:SRQ:STATE? might return :ACTONEVent:TRIGger:ACTION:SRQ:STATE 0, indicating the generate an SRQ event when a trigger event occurs feature is disabled.

ACTONEVent:TRIGger:ACTION:STOPACQ:STATE

This command stops acquisitions on a trigger event from a single sequence or sequence of N acquisition. Each acquisition in the sequence of N will perform a save operation.

Group Act On Event

Syntax
```
ACTONEVent:TRIGger:ACTION:STOPACQ:STATE {ON|OFF|<NR1>}
ACTONEVent:TRIGger:ACTION:STOPACQ:STATE?
```

Arguments `ON` enables the stop acquisitions on a trigger event feature.

`OFF` disables the stop acquisitions on a trigger event feature.

`<NR1>` is a number that enables or disables the stop acquisitions on a trigger event feature. The number zero disables the feature, all other numbers enable the feature.

Examples `ACTONEVent:TRIGger:ACTION:STOPACQ:STATE ON` sets the state of the stop acquisitions feature to on.

`ACTONEVent:TRIGger:ACTION:STOPACQ:STATE?` might return `:ACTONEVent:TRIGger:ACTION:STOPACQ:STATE OFF`, indicating the state of the stop acquisitions feature is set to off.

AFG:AMPLitude

Sets (or queries) the AFG amplitude in volts, peak to peak.

Conditions Requires option AFG.

Group AFG

Syntax
```
AFG:AMPLitude <NR3>
AFG:AMPLitude?
```

Arguments <NR3> is a floating point number that represents the AFG amplitude, peak to peak, in volts.

Examples `AFG:AMPLITUDE 1.0` sets the AFG amplitude to 1.0 volts, peak to peak.

`AFG:AMPLITUDE?` might return `:AFG:AMPLITUDE 3.0000` indicating the amplitude is set to 3.0 Volts.

AFG:ARBitrary:SOUrce

This command sets or queries the source name for the Arbitrary Waveform. Currently supported sources are either waveform file (.wfm) or text file (.csv).

Conditions Requires option AFG.

Group AFG

Syntax

```
AFG:ARBitrary:SOUrce <QString>
AFG:ARBitrary:SOUrce?
```

Arguments <QString> is the source name.

Examples AFG:ARBITRARY:SOURCE "E:/Waveforms/Square.wfm" sets the source waveform to E:/Waveforms/Square.wfm.

AFG:ARBITRARY:SOURCE? might return "E:/Waveforms/Square.wfm" indicating the source is set to E:/Waveforms/Square.wfm.

AFG:FREQuency

Sets (or queries) the AFG frequency, in Hz.

Conditions Requires option AFG.

Group AFG

Syntax

```
AFG:FREQuency <NR3>
AFG:FREQuency?
```

Arguments <NR3> is the floating point number that represents the AFG frequency, in Hz.

Examples AFG:FREQUENCY 100.0E3 sets the AFG frequency to 100 kHz.

AFG:FREQUENCY? might return :AFG:FREQUENCY 312.5000E+3 indicating the frequency is set to 312.5 kHz.

AFG:FUNCtion

Sets (or queries) which AFG function to execute.

Conditions Requires option AFG.

Group AFG

Syntax

```
AFG:FUNCtion
{SINE|SQUare|PULSe|RAMP|NOISe|DC|SINC|GAUSsian|LORENtz
|ERISe|EDECAy|HAVERSINe|CARDIac|ARBitrary}
AFG:FUNCtion?
```

Arguments

SINE
SQUare
PULSe
RAMP
NOISe
DC – The DC level is controlled by AFG:OFFSet.
SINC (Sin(x)/x)
GAUSsian
LORENtz
ERISe
EDECAy
HAVERSINe
CARDIac
ARBitrary

Examples `AFG:FUNC LOREN` specifies to generate the Lorentz function.

`AFG:FUNCTION?` might return `:AFG:FUNCTION SINE` indicating the AFG function is set to sine.

AFG:HIGHLevel

This command sets (or queries) the high level value of the output waveform, in volts, when using the arbitrary function generator feature.

Conditions Requires option AFG.

Group AFG

Syntax
```
AFG:HIGHLevel <NR3>
AFG:HIGHLevel?
```

Arguments <NR3> is a floating point number that represents the AFG high level value, in volts.

Examples `AFG:HIGHLEVEL 1.0` sets the AFG high level value to 1.0 volts.

`AFG:HIGHLEVEL`? might return `:AFG:HIGHLEVEL 1.5000` indicating the high level is set to 1.5 Volts.

AFG:LOWLevel

This command sets (or queries) the low level value of the output waveform, in volts, when using the arbitrary function generator feature.

Conditions Requires option AFG.

Group AFG

Syntax
```
AFG:LOWLevel <NR3>
AFG:LOWLevel?
```

Arguments `NR3` is the floating point number that represents the AFG low level value, in volts.

Examples `AFG:LOWLEVEL 1.0` sets the AFG low level value to 1.00 volts.

`AFG:LOWLEVEL`? might return `:AFG:LOWLEVEL -1.5000` indicating the low level is set to —1.5 Volts.

AFG:NOISEAdd:PERCent

Sets (or queries) the AFG additive noise level as a percentage. Minimum is 0.0%, maximum is 100.0% and increment is 1.0%.

Conditions Requires option AFG.

Group AFG

Syntax
```
AFG:NOISEAdd:PERCent <NR3>
AFG:NOISEAdd:PERCent?
```

Arguments Floating point number that represents the AFG additive noise level, as a percentage.

Examples `AFG:NOISEADD:PERCENT 50` sets the AFG additive noise level to 50 percent.

`AFG:NOISEADD:PERCENT?` might return `:AFG:NOISEADD:PERCENT 0` indicating that no noise is added.

AFG:NOISEAdd:STATE

Sets (or queries) the AFG additive noise state.

Conditions Requires option AFG.

Group AFG

Syntax
```
AFG:NOISEAdd:STATE {0|1|OFF|ON}
AFG:NOISEAdd:STATE?
```

Arguments `1` or `ON` turns on the AFG additive noise state.

`0` or `OFF` turns it off.

Examples `AFG:NOISEADD:STATE ON` turns on the additive noise state.

`AFG:NOISEADD:STATE?` might return `:AFG:NOISEADD:STATE 0` indicating the noise additive state is off.

AFG:OFFSet

Sets (or queries) the AFG offset value, in volts.

Conditions Requires option AFG.

Group AFG

Syntax
```
AFG:OFFSet <NR3>
AFG:OFFSet?
```

Arguments <NR3> is a floating point number that represents the AFG offset, in volts.

Examples AFG:OFFSET 1.0 sets the AFG offset to 1.0 volts.

AFG:OFFSET? might return AFG:OFFSET 0.0E+0 indicating there is no offset.

AFG:OUTPut:LOAd:IMPEDance

Sets (or queries) the AFG output load impedance.

Conditions Requires option AFG.

Group AFG

Syntax
```
AFG:OUTPut:LOAd:IMPEDance {FIFty|HIGHZ}
AFG:OUTPut:LOAd:IMPEDance?
```

Arguments FIFty sets the output load impedance to 50 Ohms.

HIGHZ sets the output load impedance to the high-impedance state.

Examples AFG:OUTP:LOA:IMPED FIF sets the AFG output load impedance to 50 Ohms.

AFG:OUTPUT:LOAD:IMPEDANCE? might return AFG:OUTPUT:LOAD:IMPEDANCE HIGHZ indicating the load impedance is set to the high impedance state.

AFG:OUTPut:STATE

Sets (or queries) the AFG output state.

Conditions Requires option AFG.

Group AFG

Syntax

```
AFG:OUTPut:STATE {0|1|OFF|ON}
AFG:OUTPut:STATE?
```

Arguments

1 or ON turns on the AFG output state.

0 or OFF turns it off.

Examples

AFG:OUTPUT:STATE ON turns on the AFG output state.

AFG:OUTPUT:STATE? might return AFG:OUTPUT:STATE 1 indicating the AFG output is on.

AFG:PERIod

Sets (or queries) the period of the AFG waveform, in seconds.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:PERIod <NR3>
AFG:PERIod?
```

Arguments

NR3 is the floating point number that represents the AFG period value, in seconds.

Returns

The query response is returned in high precision NR3 format (up to 12 digits with more than 4 trailing 0 digits after the decimal point is omitted).

Examples

AFG:PERIOD 1 sets the AFG period value to 1 second.

AFG:PERIOD? might return :AFG:PERIOD 3.2000E-6 indicating the AFG period is set to 3.2 µs.

AFG:PULse:WIDth

Sets (or queries) the AFG pulse width, in seconds.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:PULse:WIDth <NR3>
AFG:PULse:WIDth?
```

Arguments

NR3 is the floating point number that represents the pulse width, in seconds.

Examples

AFG:PULSE:WIDTH 100.0E-6 sets the AFG pulse width to 100 microseconds.

AFG:PULSE:WIDTH? might return :AFG:PULSE:WIDTH 1.0000E-6 indicating the pulse width is set to 1 µs.

AFG:RAMP:SYMmetry

Sets (or queries) the AFG ramp symmetry in percent. Minimum is 0.0%, maximum is 100.0% and increment is 0.10%.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:RAMP:SYMmetry <NR3>
AFG:RAMP:SYMmetry?
```

Arguments

Floating point number that represents the AFG ramp symmetry, as a percentage.

Examples

AFG:RAMP:SYMMETRY 50.0 sets the AFG ramp symmetry to 50 percent.

AFG:RAMP:SYMMETRY? might return :AFG:RAMP:SYMMETRY 50.0000 indicating the symmetry is set to 50%.

AFG:SQUare:DUty

Sets (or queries) the AFG duty cycle in percent. The minimum is 10.0%, maximum is 90.0% and increment is 0.10%.

Conditions

Requires option AFG.

Group

AFG

Syntax

```
AFG:SQUare:DUty <NR3>
AFG:SQUare:DUty?
```

Arguments

Floating point number that represents the AFG duty cycle, as a percentage.

Examples

`AFG:SQUARE:DUTY 50.0` sets the AFG duty cycle to 50 percent.

`AFG:SQUARE:DUTY?` might return `:AFG:SQUARE:DUTY 50.0000` indicating the duty cycle is set to 50%.

ALIas

This command sets or queries the state of alias functionality, and it is identical to the ALIAS:STATE command.

Group

Alias

Syntax

```
ALIas {OFF|ON|<NR1>}
ALIas?
```

Related Commands

ALIas:STATE

Arguments

`OFF` turns Alias expansion off.

`ON` turns Alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

`<NR1>` = 0 disables Alias mode; any other value enables Alias mode.

Examples

`ALIAS ON` turns the alias feature on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

`ALIAS?` might return `:ALIAS:STATE 1`, indicating that the alias feature is on.

ALIas:CATalog? (Query Only)

This query-only command returns a list of the currently defined alias labels, separated by commas. If no aliases are defined, the query returns the string “”.

Group

Alias

Syntax `ALIas:CATalog?`

Examples `ALIAS:CATALOG?` might return the string `:ALIAS:CATALOG "SETUP1","TESTMENU1","DEFAULT"` showing that there are three aliases named SETUP1, TESTMENU1, and DEFAULT.

ALIas:DEFine

This command assigns a sequence of program messages to an alias label. These messages are then substituted for the alias whenever it is received as a command or query, provided that `ALIas:STATE` is turned on. The query form of this command returns the definitions of a selected alias.

NOTE. *Attempting to give two aliases the same name causes an error. To give a new alias the name of an existing alias, the existing alias must first be deleted.*

Group Alias

Syntax
```
ALIas:DEFine <QString><,>{<QString>|<Block>}
ALIas:DEFine?
```

Related Commands ALIas:STATE

Arguments The first `<QString>` is the alias label.

This label cannot be a command name. Labels must start with a letter and can contain only letters, numbers, and underscores; other characters are not allowed. The label must be less than or equal to 12 characters.

The second `<QString>` or `<Block>` is a complete sequence of program messages.

The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 256 characters.

Examples `ALIAS:DEFINE "ST1",":RECALL:SETUP 'C:/mySetup.set';:AUTOSET EXECUTE"` defines an alias named “ST1” that recalls a previously saved setup and performs an autoset.

`ALIAS:DEFINE? "ST1"` returns `:ALIAS:DEFINE "ST1",#247 :RECALL:SETUP 'C:/mySetup.set';:AUTOSET EXECUTE`.

ALIas:DELETe (No Query Form)

This command removes a specified alias and is identical to `ALIas:DELETe:NAMe`. An error message is generated if the named alias does not exist.

Group Alias

Syntax `ALIas:DELETe <QString>`

Related Commands *ESR?

ALIas:DELETe:ALL

Arguments `<QString>` is the name of the alias to be removed. Using `ALIas:DELETe` without specifying an alias causes an execution error. `<QString>` must be a previously defined value.

Examples `ALIAS:DELETE "SETUP1"` deletes the alias named SETUP1.

ALIas:DELETe:ALL (No Query Form)

This command deletes all existing aliases.

Group Alias

Syntax `ALIas:DELETe:ALL`

Related Commands ALIas:DELETe

ALIas:DELETe:NAMe

Examples `ALIAS:DELETE:ALL` deletes all existing aliases.

ALIas:DELETe:NAMe (No Query Form)

This command removes a specified alias. An error message is generated if the named alias does not exist. This command is identical to `ALIas:DELETe`.

Group Alias

Syntax `ALIas:DELEte:NAMe <QString>`

Arguments `<QString>` is the name of the alias to remove. Using `ALIas:DELEte:NAMe` without specifying an alias causes an exception error.`<QString>` must be a previously defined alias.

Examples `ALIAS:DELETE:NAME "STARTUP"` deletes the alias named STARTUP.

ALIas:STATE

This command turns aliases on or off and is identical to the ALIas command. The query form of this command returns the state of the aliases.

Group Alias

Syntax
```
ALIas:STATE {<NR1>|OFF|ON}
ALIas:STATE?
```

Arguments `<NR1>` = 0 turns off aliases; any other value turns on aliases.

`OFF` turns alias expansion off.

`ON` turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

Examples `ALIAS:STATE OFF` turns off the alias feature.

`ALIAS:STATE?` might return `:ALIAS:STATE ON`, indicating that alias expansion is currently turned on.

ALLEv? (Query Only)

This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. Use the *ESR? query to enable the events to be returned. This command is similar to repeatedly sending *EVMsg? queries to the instrument.

Group Status and Error

Syntax `ALLEv?`

Related Commands *ESR?

EVMsg?

Examples ALLEV? might return :ALLEV 2225,"Measurement error, No waveform to measure; "420,"Query UNTERMINATED;".

AUTOSAVEPITIMEOUT

This command sets or queries the idle time from the programmable interface before auto-save occurs.

Group Miscellaneous

Syntax
```
AUTOSAVEPITIMEOUT <NR1>
AUTOSAVEPITIMEOUT?
```

Arguments <NR1>

Examples AUTOSAVEPITIMEOUT 100000 sets the timeout to 100 seconds.

AUTOSAVEPITIMEOUT? might return :AUTOSAVEPITIMEOUT 180000 indicating the time out value is 180 seconds.

AUTOSAVEUITIMEOUT

This command sets or queries the idle time from the user interface before auto-save occurs.

Group Miscellaneous

Syntax
```
AUTOSAVEUITIMEOUT <NR1>
AUTOSAVEUITIMEOUT?
```

Arguments <NR1>

Examples AUTOSAVEUITIMEOUT 4000 sets the timeout to 4 seconds.

AUTOSAVEUITIMEOUT? might return :AUTOSAVEUITIMEOUT 3000 indicating the time out value is 3 seconds.

AUTOSet (No Query Form)

This command (no query format) sets the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform.

Group Miscellaneous

Syntax

```
AUTOSet EXECute
```

Arguments `EXECute` autosets the displayed waveform; this is equivalent to pressing the front panel Autoset button.

Examples `AUTOSET EXECute` autosets the displayed waveform.

AUTOSet:ACQuisition:ENAble

This command sets or queries the Autoset acquisition setting adjustment. Settings affected may include, but not be limited to, acquisition mode, and FastAcq mode.

Group Miscellaneous

Syntax

```
AUTOSet:ACQuisition:ENAble {ON|1|OFF|0}
AUTOSet:ACQuisition:ENAble?
```

Related Commands AUTOSet:HORizontal:ENAble

AUTOSet:TRIGger:ENAble

AUTOSet:VERTical:ENAble

Arguments `ON` or `1` enables Autoset to change acquisition settings.

`OFF` or `0` disables Autoset from changing acquisition settings.

Examples `AUTOSet:ACQuisition:ENAble OFF` turns off Autoset's adjustment of the acquisition settings.

`AUTOSet:ACQuisition:ENAble?` might return 0, indicating Autoset's adjustment of the acquisition settings is turned off.

AUTOSet:ENAble

This command sets or queries the Autoset enable/disable feature. This is useful for classroom purposes where the instructor wants the students to achieve the desired instrument settings without the benefit of the Autoset feature.

Group Miscellaneous

Syntax

```
AUTOSet:ENAble {ON|1|OFF|0}
AUTOSet:ENAble?
```

Related Commands AUTOSet:ACQuisition:ENAble

AUTOSet:HORizontal:ENAble

AUTOSet:TRIGger:ENAble

AUTOSet:VERTical:ENAble

Arguments `ON` or `1` enables Autoset.

`OFF` or `0` disables Autoset.

Examples `AUTOSet:ENAble ON` turns on the Autoset feature.

`AUTOSet:ENAble?` might return 1, indicating that Autoset is enabled.

AUTOSet:HORizontal:ENAble

This command sets or queries Autoset's adjustment of horizontal settings. Settings affected may include, but not be limited to, horizontal scale, horizontal position, and horizontal delay mode.

Group Miscellaneous

Syntax

```
AUTOSet:HORizontal:ENAble {ON|1|OFF|0}
AUTOSet:HORizontal:ENAble?
```

Related Commands AUTOSet:ACQuisition:ENAble

AUTOSet:TRIGger:ENAble

AUTOSet:VERTical:ENAble

Arguments

ON or 1 enables Autoset to change horizontal settings.

OFF or 0 disables Autoset from changing horizontal settings.

Examples

AUTOSet:HORizontal:ENAble OFF turns off Autoset's adjustment of the horizontal settings.

AUTOSet:HORizontal:ENAble? might return 0, indicating Autoset's adjustment of the horizontal settings is turned off.

AUTOSet:TRIGger:ENAble

This command sets or queries Autoset's adjustment of trigger settings. Settings affected may include, but not be limited to, trigger level, trigger source, and trigger coupling.

Group

Miscellaneous

Syntax

```
AUTOSet:TRIGger:ENAble {ON|1|OFF|0}
AUTOSet:TRIGger:ENAble?
```

Related Commands

AUTOSet:ACQuisition:ENAble

AUTOSet:HORizontal:ENAble

AUTOSet:VERTical:ENAble

Arguments

ON or 1 enables Autoset to change trigger settings.

OFF or 0 disables Autoset from changing trigger settings.

Examples

AUTOSet:TRIGger:ENAble OFF turns off Autoset's adjustment of the trigger settings.

AUTOSet:TRIGger:ENAble? might return 0, indicating Autoset's adjustment of the trigger settings is turned off.

AUTOSet:VERTical:ENAble

This command sets or queries Autoset's adjustment of vertical settings. Settings affected may include, but not be limited to, vertical scale, vertical position, and vertical offset.

Group Miscellaneous

Syntax
```
AUTOSet:VERTical:ENAble {ON|1|OFF|0}
AUTOSet:VERTical:ENAble?
```

Related Commands AUTOSet:ACQuisition:ENAble

AUTOSet:HORizontal:ENAble

AUTOSet:TRIGger:ENAble

Arguments `ON` or `1` enables Autoset to change vertical settings.

`OFF` or `0` disables Autoset from changing vertical settings.

Examples `AUTOSet:VERTical:ENAble OFF` turns off Autoset's adjustment of the vertical settings.

`AUTOSet:VERTical:ENAble?` might return 0, indicating Autoset's adjustment of the vertical settings is turned off.

AUTOSet:VERTical:OPTIMize

This command sets or queries which vertical settings Autoset will optimize when the display mode is set to Overlay mode (all waveforms are in one common graticule in the Waveform View).

Group Miscellaneous

Syntax
```
AUTOSet:VERTical:OPTIMize {RESOlution|VISIBility}
AUTOSet:VERTical:OPTIMize?
```

Related Commands DISplay:WAVEView<x>:VIEWStyle

Arguments `RESOlution` uses as much of the ADC's (Analog to Digital Converter) range as possible to provide the best vertical resolution and measurement accuracy, but waveforms will overlap each other.

`VISIBility` vertically scales and positions waveforms so they are visually separated from each other at the expense of vertical resolution and measurement accuracy.

Examples

`AUTOSet:VERTical:OPTIMize VISIBility` will vertically scale and position all active waveforms so that they are visually separated from each other when autoset is executed.

`AUTOSet:VERTical:OPTIMize?` might return `AUTOSet:VERTical:OPTIMize RESOLUTION`, which indicates that, when executed, autoset will vertically scale/position the active waveforms to maximize the ADC range while in Overlay display mode.

AUXout:EDGE

This command sets or queries the direction in which the Auxiliary Output signal will transition when a trigger occurs.

Group

Miscellaneous

Syntax

```
AUXout:EDGE {RISing|FALling}
AUXout:EDGE?
```

Arguments

`RISing` sets the polarity to the rising edge.

`FALling` sets the polarity to the falling edge.

Examples

`AUXOUT:EDGE RISING` sets the polarity to rising edge.

`AUXOUT:EDGE?` might return `:AUXOUT:EDGE FALLING`, indicating that the polarity is set to falling edge.

AUXout:SOUrce

This command sets or queries the source at the Auxiliary Output BNC connection.

Group

Miscellaneous

Syntax

```
AUXout:SOUrce {ATRIGger|REFOUT|AFG}
AUXout:SOUrce?
```

Arguments

`ATRIGger` sets the source at the BNC connector to the main trigger.

`REFOUT` sets the source at the BNC connector to the reference output.

`AFG` sets the source at the BNC connector to the AFG output.

Examples AUXOUT:SOURCE? might return :AUXOUT:SOURCE ATRIGGER, indicating that the source at the BNC connector is set to the A trigger.

BUS:ADDNew (No Query Form)

This command adds the specified bus. This command creates/adds the bus but does not display it (turn it on). In order to enable bus decoding and see the bus display on screen, send the DISplay:WAVEView<x>:BUS:B<x>:STATE ON command.

Group Bus

Syntax BUS:ADDNew <QString>

Related Commands BUS:B<x>:TYPe

DISplay:WAVEView<x>:BUS:B<x>:STATE

Arguments <QString> is a quoted string of the form "B<NR1>" where NR1 is ≥1.

Examples BUS:ADDNEW "B5";:display:waveview1:bus:b5:state on creates bus 3 with the default type of Parallel, and then turns it on.

BUS:B<x>:ARINC429A:BITRate

This command sets or queries the ARINC429 bit rate for the specified bus. The bus number is specified by x. If you select Custom, use BUS:B<x>:ARINC429A:BITRate:CUSTom to set the bit rate.

Conditions Requires option SR-AERO.

Group Bus

Syntax
```
BUS:B<x>:ARINC429A:BITRate {LOW|HI|CUSTom}
BUS:B<x>:ARINC429A:BITRate?
```

Related Commands B<x> is the number of the bus.

BUS:B<x>:ARINC429A:BITRate:CUSTom

Arguments Arguments specify the bit rate.

Examples `BUS:B1:ARINC429A:BITRATE LOW` sets the bit rate to handle low speed signals.

`BUS:B1:ARINC429A:BITRATE?` might return `:BUS:B1:ARINC429A:BITRATE HI`, indicating the bit rate is set to handle high speed signals.

BUS:B<x>:ARINC429A:BITRate:CUSTom

This command sets or queries the ARINC429 custom bit rate for the specified bus. The bus is specified by x.

Conditions Requires option SR-AERO.

Group Bus

Syntax

```
BUS:B<x>:ARINC429A:BITRate:CUSTom <NR1>
BUS:B<x>:ARINC429A:BITRate:CUSTom?
```

Related Commands BUS:B<x>:ARINC429A:BITRate

Arguments `B<x>` is the number of the bus.

`<NR1>` is the ARINC429 custom bit rate for the specified bus.

Examples `BUS:B1:ARINC429A:BITRATE:CUSTOM 12500` sets the bit rate to 12,500.

`BUS:B1:ARINC429A:BITRATE:CUSTOM?` might return `:BUS:B1:ARINC429A:BITRATE:CUSTOM 100000`, indicating the bit rate is set to 100,000.

BUS:B<x>:ARINC429A:DATAFORmat

This command sets or queries the format of the DATA field for the specified ARINC429 bus. The bus is specified by x.

Conditions Requires option SR-AERO.

Group Bus

Syntax

```
BUS:B<x>:ARINC429A:DATAFORmat {DATA|SDIDATA|SDIDATASSM}
BUS:B<x>:ARINC429A:DATAFORmat?
```

Arguments

B<x> is the number of the bus.

DATA specifies a DATA field width of 19 bits (covering bits 11 through 29 of the 32 bit packet)

SDIDATA specifies a DATA field width of 21 bits (covering bits 9 through 29 of the 32 bit packet)

SDIDATASSM specifies a DATA field width of 23 bits (covering bits 9 through 31 of the 32 bit packet)

Examples

BUS:B1:ARINC429A:DATAFORMAT SDIDATASSM sets Bus 1 ARINC429 DATA field width to 23 bits.

BUS:B1:ARINC429A:DATAFORMAT? might return :BUS:B1:ARINC429A:DATAFORMAT DATA, indicating a DATA field width of 19 bits.

BUS:B<x>:ARINC429A:POLARITY

This command sets or queries the source polarity for the specified ARINC429 bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

```
BUS:B<x>:ARINC429A:POLARITY {NORMal|INVERTed}
BUS:B<x>:ARINC429A:POLARITY?
```

Arguments

B<x> is the number of the bus.

NORMal specifies normal polarity.

INVERTed specifies inverted polarity.

Examples

BUS:B3:ARINC429A:POLARITY INVERTED sets Bus 3 ARINC429 polarity to inverted.

BUS:B2:ARINC429A:POLARITY? might return :BUS:B2:ARINC429A:POLARITY NORMAL, indicating that the Bus 2 ARINC429 polarity is set to normal.

BUS:B<x>:ARINC429A:SOUrce

This command sets or queries the source for the specified ARINC429 bus. The bus is specified by x.

Conditions Requires option SR-AERO.

Group Bus

Syntax

```
BUS:B<x>:ARINC429A:SOUrce {CH<X>|Math<x>|REF<x>}
BUS:B<x>:ARINC429A:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source waveform for the ARINC429 bus.

Math<x> specifies a math waveform as the source waveform for the ARINC429 bus.

REF<x> specifies a reference waveform as the source waveform for the ARINC429 bus.

Examples

BUS:B1:ARINC429A:SOURCE CH1 sets channel 1 as the source for the ARINC429 bus.

BUS:B1:ARINC429A:SOURCE? might return :BUS:B1:ARINC429A:SOURCE MATH2 indicating that the source is set to MATH2.

BUS:B<x>:ARINC429A:THRESHold

This command sets or queries the ARINC429 upper threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-AERO.

Group Bus

Syntax

```
BUS:B<x>:ARINC429A:THRESHold <NR3>
BUS:B<x>:ARINC429A:THRESHold?
```

Related Commands

BUS:B<x>:ARINC429A:SOUrce

Arguments

B<x> is the number of the bus.

<NR3> is the ARINC429 lower threshold for the specified bus.

Examples

`BUS:B3:ARINC429A:THRESHOLD -200e-3` sets the Bus 3 ARINC429 upper threshold to -200 mV.

`BUS:B2:ARINC429A:THRESHOLD?` might return `:BUS:B2:ARINC429A:THRESHOLD -500.0000E-3` indicating the Bus 2 ARINC429 upper threshold is set to -500 mV.

BUS:B<x>:AUDio:BITDelay

This command sets or queries the number of delay bits for the specified AUDIO bus. The bus is specified by x.

NOTE. *This command only applies to the TDM Audio type.*

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:BITDelay <NR1>
BUS:B<x>:AUDio:BITDelay?
```

Arguments

B<x> is the number of the bus.

<NR1> specifies the number of delay bits.

Examples

`BUS:B1:AUDIO:BITDELAY 2` sets the bit delay to 2.

`BUS:B1:AUDIO:BITDELAY?` might return `:BUS:B1:AUDIO:BITDELAY 1` indicating that the number of delay bits is 1.

BUS:B<x>:AUDio:BITOrder

Specifies the bit order for the specified AUDIO bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax

```
BUS:B<x>:AUDio:BITOrder {MSB|LSB}
BUS:B<x>:AUDio:BITOrder?
```

Arguments

`B<x>` is the number of the bus.

`MSB` specifies that the most significant bit will be expected first in the order.

`LSB` specifies that the least significant bit will be expected first in the order.

Examples

`BUS:B1:AUDIO:BITORDER LSB` sets the bit order to LSB.

`BUS:B1:AUDIO:BITORDER?` might return `:BUS:B1:AUDIO:BITORDER MSB` indicating that the MSB is first in the bit order.

BUS:B<x>:AUDio:CLOCk:POLarity

This command sets or queries the clock source polarity for the specified AUDIO bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax

```
BUS:B<x>:AUDio:CLOCk:POLarity {FALL|RISE}
BUS:B<x>:AUDio:CLOCk:POLarity?
```

Arguments

`B<x>` is the number of the bus.

`FALL` sets falling edge as the clock polarity.

`RISE` sets rising edge as the clock polarity.

Examples `BUS:B1:AUDIO:CLOCK:POLARITY FALL` sets the clock polarity to Fall.

`BUS:B1:AUDIO:CLOCK:POLARITY?` might return `:BUS:B1:AUDIO:CLOCK:POLARITY RISE` indicating that the clock polarity is set to Rise.

BUS:B<x>:AUDio:CLOCk:SOUrce

This command sets or queries the clock source waveform for the specified AUDIO bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax

```
BUS:B<x>:AUDio:CLOCk:SOUrce
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:AUDio:CLOCk:SOUrce?
```

Arguments `B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the clock source waveform for the audio bus.

`CH<x>_D<x>` specifies a digital channel as the clock source waveform for the specified audio bus.

`Math<x>` specifies a math waveform as the clock source waveform for the audio bus.

`REF<x>` specifies a reference waveform as the clock source waveform for the audio bus.

`REF<x>_D<x>` specifies a digital reference waveform as the clock source waveform for the specified audio bus.

Examples `BUS:B1:AUDIO:CLOCK:SOURCE CH1_D1` sets D1 of channel 1 as the clock source for the audio bus.

`BUS:B1:AUDIO:CLOCK:SOURCE?` might return `:BUS:B1:AUDIO:CLOCK:SOURCE CH1` indicating that the clock source is set to channel 1.

BUS:B<x>:AUDio:CLOCk:THReshold

This command sets or queries the audio clock source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax

```
BUS:B<x>:AUDio:CLOCk:THReshold <NR3>
BUS:B<x>:AUDio:CLOCk:THReshold?
```

Related Commands BUS:B<x>:AUDio:CLOCk:SOUrce

Arguments B<x> is the number of the bus.

<NR3> is the audio clock source threshold for the specified bus.

Examples BUS:B3:AUDIO:CLOCK:THRESHOLD 500.0E-3 sets the Bus 3 audio Cclock source threshold to 500.0 mV.

BUS:B2:AUDIO:CLOCK:THRESHOLD? might return :BUS:B2:AUDIO:CLOCK:THRESHOLD 1.0 indicates the Bus 2 audio clock source threshold is set to 1.0 V.

BUS:B<x>:AUDio:DATa:POLarity

This command sets or queries the audio data source polarity for the specified audio bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax

```
BUS:B<x>:AUDio:DATa:POLarity {HIGH|LOW}
BUS:B<x>:AUDio:DATa:POLarity?
```

Arguments B<x> is the number of the bus.

HIGH specifies positive data polarity for the audio bus.

LOW specifies negative data polarity for the audio bus.

Examples BUS:B1:AUDIO:DATA:POLARITY LOW sets the data polarity to LOW.

BUS:B1:AUDIO:DATA:POLARITY? might return :BUS:B1:AUDIO:DATA:POLARITY HIGH indicating that the data polarity is set to HIGH.

BUS:B<x>:AUDio:DATa:SIZe

This command sets or queries the number of bits per channel for the specified audio bus. The bus is specified by x.

NOTE. *This command only applies to the TDM Audio type.*

Conditions Requires option SR-AUDIO.

Group Bus

Syntax
```
BUS:B<x>:AUDio:DATa:SIZe <NR1>
BUS:B<x>:AUDio:DATa:SIZe?
```

Arguments B<x> is the number of the bus.

NR1 specifies the number of bits per word.

Examples BUS:B1:AUDIO:DATA:SIZE 8 sets the number of bits per word to 8.

BUS:B1:AUDIO:DATA:SIZE? might return :BUS:B1:AUDIO:DATA:SIZE 24 indicating that the number of bits per word is set to 24.

BUS:B<x>:AUDio:DATa:SOUrce

This command sets or queries the audio data source for the specified audio bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax

```
BUS:B<x>:AUDio:DATa:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:AUDio:DATa:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the data source waveform for the audio bus.

CH<x>_D<x> specifies a digital channel as the data source waveform for the audio bus.

MATH<x> specifies an math waveform as the data source waveform for the audio bus.

REF<x> specifies an reference waveform as the data source waveform for the audio bus.

REF<x>_D<x> specifies a digital reference waveform as the data source waveform for the specified audio bus.

Examples

BUS:B1:AUDIO:DATA:SOURCE CH1_D1 sets the data source to D1 of FlexChannel 1.

BUS:B1:AUDIO:DATA:SOURCE? might return :BUS:B1:AUDIO:DATA:SOURCE CH3 indicating that the data source is set to CH3.

BUS:B<x>:AUDio:DATa:THReshold

This command sets or queries the audio data source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AUDIO.

Group

Bus

Syntax

```
BUS:B<x>:AUDio:DATa:THReshold <NR3>
BUS:B<x>:AUDio:DATa:THReshold?
```

Related Commands

BUS:B<x>:AUDio:DATa:SOUrce

Arguments

B<x> is the number of the bus.

<NR3> is the audio data source threshold for the specified bus.

Examples `BUS:B3:AUDIO:DATA:THRESHOLD` 1.5 sets the Bus 3 audio clock source threshold to 500.0 mV.

`BUS:B2:AUDIO:DATA:THRESHOLD?` might return `:BUS:B2:AUDIO:DATA:THRESHOLD` 500.0000E-3 indicates the Bus 2 audio data source threshold is set to 500 mV.

BUS:B<x>:AUDio:DATa:WORDSize

This command sets or queries the audio bits per word for the specified bus. The bus is specified by x.

Group Bus

Syntax
```
BUS:B<x>:AUDio:DATa:WORDSize <NR1>
BUS:B<x>:AUDio:DATa:WORDSize?
```

Arguments `B<x>` is the number of the bus.

`<NR1>` is the audio bits per word for the specified bus.

Examples `BUS:B1:AUDio:DATa:WORDSize 24` sets the audio bits per word to 24 bits.

`BUS:B1:AUDio:DATa:WORDSize?` might return `:BUS:B1:AUDIO:DATA:WORDSIZE 24` indicating the bits per word is 24.

BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL

This command sets or queries the audio bits of sync width for the specified bus. The bus is specified by x.

NOTE. *This command only applies to the TDM Audio type.*

Conditions Requires option SR-AUDIO.

Group Bus

Syntax
```
BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL <NR1>
BUS:B<x>:AUDio:FRAME:CLOCKBITSPERCHANNEL?
```

Arguments B<x> is the number of the bus.

<NR1> is the audio bits of sync width for the specified bus.

Examples BUS:B1:AUDio:FRAME:CLOCKBITSPERCHANNEL 32 sets the number of bits to 32.

BUS:B1:AUDio:FRAME:CLOCKBITSPERCHANNEL? might return :BUS:B1:AUDIO:FRAME:CLOCKBITSPERCHANNEL 24 indicating there are 24 bits of sync width for the bus.

BUS:B<x>:AUDio:FRAME:SIZe

This command sets or queries the number of audio channels in each frame for the specified AUDIO bus. The bus is specified by x.

NOTE. *This command only applies to the TDM Audio type.*

Conditions Requires option SR-AUDIO.

Group Bus

Syntax
```
BUS:B<x>:AUDio:FRAME:SIZe <NR1>
BUS:B<x>:AUDio:FRAME:SIZe?
```

Arguments B<x> is the number of the bus.

<NR1> specifies the number of channels in each frame.

Examples BUS:B1:AUDIO:FRAME:SIZE 2 sets the frame size to 2.

BUS:B1:AUDIO:FRAME:SIZE? might return :BUS:B1:AUDIO:FRAME:SIZE 8 indicating that the number of channels in each frame is set to 8.

BUS:B<x>:AUDio:TYPe

This command sets or queries the audio format (type) for the specified audio bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax
```
BUS:B<x>:AUDio:TYPe {I2S|LJ|RJ|TDM}
BUS:B<x>:AUDio:TYPe?
```

Arguments `B<x>` is the number of the bus.

`I2S` specifies the I²S audio format.

`LJ` specifies the left-justified audio format.

`RJ` specifies the right-justified audio format.

`TDM` specifies the time-division multiplexing audio format.

Examples `BUS:B1:AUDIO:TYPE RJ` sets right-justified as the audio format.

`BUS:B1:AUDIO:TYPE?` might return `:BUS:B1:AUDIO:TYPE I2S` indicating that the audio format is set to I²S.

BUS:B<x>:AUDio:WORDSel:POLarity

This command sets or queries the word select source polarity for the specified audio bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax
```
BUS:B<x>:AUDio:WORDSel:POLarity {NORMal|INVERTed}
BUS:B<x>:AUDio:WORDSel:POLarity?
```

Arguments `B<x>` is the number of the bus.

`NORMal` specifies positive polarity.

`INVERTed` specifies negative polarity.

Examples `BUS:B1:AUDIO:WORDSEL:POLARITY NORMal` sets normal as the word select polarity.

`BUS:B1:AUDIO:WORDSEl:POLARITY?` might return `:BUS:B1:AUDIO:WORDSEL:POLARITY NORMAL` indicating that the word select polarity is set to normal.

BUS:B<x>:AUDio:WORDSel:SOUrce

This command sets or queries the audio word select source waveform for the specified audio bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax

```
BUS:B<x>:AUDio:WORDSel:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:AUDio:WORDSel:SOUrce?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the word select source waveform.

`CH<x>_D<x>` specifies a digital channel as the word select source waveform.

`MATH<x>` specifies an math waveform as the word select source waveform.

`REF<x>` specifies an reference waveform as the word select source waveform.

`REF<x>_D<x>` specifies a digital reference waveform as the word select source waveform for the specified audio bus.

Examples

`BUS:B1:AUDIO:WORDSEL:SOURCE CH1` sets CH1 as the word select source.

`BUS:B1:AUDIO:WORDSEL:SOURCE CH2_D3` sets D3 of FlexChannel 2 as the word select source.

`BUS:B1:AUDIO:WORDSEL:SOURCE?` might return `:BUS:B1:AUDIO:WORDSEL:SOURCE CH2` indicating that the word select source is set to CH2.

BUS:B<x>:AUDio:WORDSel:THReshold

This command sets or queries the audio word select source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-AUDIO.

Group Bus

Syntax

```
BUS:B<x>:AUDio:WORDSel:THReshold <NR3>
BUS:B<x>:AUDio:WORDSel:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the audio word select source threshold for the specified bus.

Examples

BUS:B4:AUDIO:WORDSEL:THRESHOLD 300.0E-3 sets the Bus 4 audio word select source threshold to 300 mV.

BUS:B1:AUDIO:WORDSEL:THRESHOLD? might return :BUS:B1:AUDIO:WORDSEL:THRESHOLD 1.25 indicates the Bus 1 audio word select source threshold is set to 1.25 V.

BUS:B<x>:AUTOETHERnet:DATAMINUSTHRESHOLD

This command sets or queries the AutoEthernet D– source threshold level for the specified bus. This threshold only applies when the AutoEthernet signal type is single ended.

Conditions

Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:DATAMINUSTHRESHOLD <NR3>
BUS:B<x>:AUTOETHERnet:DATAMINUSTHRESHOLD?
```

Arguments

Bus<x> is the Bus number.

NR3 specifies the AutoEthernet D– source threshold level for the specified bus, in volts. This threshold only applies when the AutoEthernet signal type is single ended.

Examples

BUS:B3:AUTOETHERnet:DATAMINUSTHRESHOLD 250.0E-3 sets the Bus 3 AutoEthernet D– source threshold level to 250.0000 mV.

BUS:B2:AUTOETHERnet:DATAMINUSTHRESHOLD? might return BUS:B2:AUTOETHERnet:DATAMINUSTHRESHOLD 1.0, indicating the Bus 2 AutoEthernet D– source threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:DATAPLUSTHRESHold

This command sets or queries the AutoEthernet D+ source threshold level for the specified bus. This threshold only applies when the AutoEthernet signal type is single ended.

Conditions Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:DATAPLUSTHRESHold <NR3>
BUS:B<x>:AUTOETHERnet:DATAPLUSTHRESHold?
```

Arguments `Bus<x>` is the Bus number.

`NR3` specifies the AutoEthernet D+ source threshold for the specified bus, in volts. This threshold only applies when the AutoEthernet signal type is single ended.

Examples `BUS:B3:AUTOETHERnet:DATAPLUSTHRESHOLD 250.0E-3` sets the Bus 3 AutoEthernet D+ source threshold level to 250.0000 mV.

`BUS:B2:AUTOETHERnet:DATAPLUSTHRESHOLD?` might return `BUS:B2:AUTOETHERnet:DATAPLUSTHRESHOLD 1.0`, indicating the Bus 2 AutoEthernet D+ source threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:LOWDATAMINus

This command sets or queries the AutoEthernet D– source low threshold level for the specified bus. This threshold only applies when the AutoEthernet signal type is single ended.

Conditions Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:LOWDATAMINus <NR3>
BUS:B<x>:AUTOETHERnet:LOWDATAMINus?
```

Arguments

Bus<x> is the Bus number.

NR3 specifies the AutoEthernet D– source low threshold level for the specified bus, in volts. This threshold only applies when the AutoEthernet signal type is single ended.

Examples

BUS:B1:AUTOETHERnet:LOWDATAMINus 250.0E-3 sets the Bus 1 AutoEthernet D– source low threshold level to 250.0000 mV.

BUS:B2:AUTOETHERnet:LOWDATAMINus? might return BUS:B2:AUTOETHERnet:LOWDATAMINus 1.0, indicating the Bus 2 AutoEthernet D– source threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:LOWDATAPLUS

This command sets or queries the AutoEthernet D+ source low threshold level for the specified bus. This threshold only applies when the AutoEthernet signal type is single ended.

Conditions

Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:LOWDATAPLUS <NR3>
BUS:B<x>:AUTOETHERnet:LOWDATAPLUS?
```

Arguments

Bus<x> is the Bus number.

NR3 specifies the AutoEthernet D+ source low threshold level for the specified bus, in volts. This threshold only applies when the AutoEthernet signal type is single ended.

Examples

BUS:B3:AUTOETHERnet:LOWDATAPLUS 250.0E-3 sets the Bus 3 AutoEthernet D+ source low threshold level to 250.0000 mV.

BUS:B2:AUTOETHERnet:LOWDATAPLUS? might return BUS:B2:AUTOETHERnet:LOWDATAPLUS 1.0, indicating the Bus 2 AutoEthernet D+ source low threshold level is set to 1.0 V.

BUS:B<x>:AUTOETHERnet:LOWTHRESHold

This command sets or queries the AutoEthernet Data source Low threshold level for the specified bus. This threshold only applies when the AutoEthernet signal type is differential.

Conditions Requires option SR-AUTOENET

The AutoEthernet signal type is differential.

Group Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:LOWTHRESHold <NR3>
BUS:B<x>:AUTOETHERnet:LOWTHRESHold?
```

Related Commands BUS:B<x>:AUTOETHERnet:SIGNALTYpe

Arguments `Bus<x>` is the Bus number.

`NR3` specifies the AutoEthernet DATA source Low threshold level for the specified bus, in volts. This threshold only applies when the AutoEthernet signal type is differential.

Examples `BUS:B1:AUTOETHERnet:LOWTHRESHold -200e-3` sets the bus 1 low threshold to -200 mV.

`BUS:B3:AUTOETHERnet:LOWTHRESHold?` might return `BUS:B3:AUTOETHERnet:LOWTHRESHOLD -500.0000E-3`, indicating the threshold is set to -500 mV for bus 3.

BUS:B<x>:AUTOETHERnet:SIGNALTYpe

This command sets or queries the AutoEthernet signal type for the specified bus.

Conditions Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:AUTOETHERnet:SIGNALTYpe?
```

Arguments

`Bus<x>` is the Bus number.

`SINGLE` specifies single-ended signals.

`DIFF` specifies differential signals.

Examples

`BUS:B4:AUTOETHERnet:SIGNALTYpe SINGLE` specifies the AutoEthernet signal type as single-ended for Bus B4.

`BUS:B3:AUTOETHERnet:SIGNALTYpe?` might return `BUS:B3:AUTOETHERnet:SIGNALTYpe DIFF`, indicating that the AutoEthernet signal type is differential for Bus 3.

BUS:B<x>:AUTOETHERnet:SOUrce

This command sets or queries the AutoEthernet data (SDATA) source for the specified bus. This command controls the source channel when the signal type is differential.

Conditions

Requires option SR-AUTOENET

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:SOUrce {CH<x>|MATH<x>|REF<x<>}
BUS:B<x>:AUTOETHERnet:SOUrce?
```

Related Commands

BUS:B<x>:AUTOETHERnet:THRESHold

Arguments

`Bus<x>` is the Bus number.

`CH<x>` specifies an analog channel as the AutoEthernet data source for differential input, where <x> is the channel number.

`MATH<x>` specifies a math channel as the AutoEthernet data source for differential input, where <x> is the math waveform number.

`REF<x>` specifies a reference waveform as the AutoEthernet data source for differential input, where <x> is the reference waveform number.

Examples

`BUS:B5:AUTOETHERnet:SOURCE CH4` specifies to use the channel 4 waveform as the source for AutoEthernet data on bus 5.

BUS:B1:AUTOETHERnet:SOURCE? might return BUS:B1:AUTOETHERnet:SOURCE? CH2, indicating that channel 2 is the currently specified source for AutoEthernet data on bus 1.

BUS:B<x>:AUTOETHERnet:SOUrce:DMINus

This command sets or queries the AutoEthernet D– source for the specified bus. This command specifies the source channel to use when the signal type is single ended.

Conditions Requires option SR-AUTOENET

Group Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:SOUrce:DMINus {CH<x>|MATH<x>|REF<x<>}
BUS:B<x>:AUTOETHERnet:SOUrce:DMINus?
```

Arguments

Bus<x> is the Bus number.

CH<x> specifies an analog channel as the source for the D– signal, where <x> is the channel number.

MATH<x> specifies a math channel as the source for the D– signal, where <x> is the math waveform number.

REF<x> specifies a reference waveform as the source for the D– signal, where <x> is the reference waveform number.

Examples

BUS:B2:AUTOETHERnet:SOUrce:DMINus CH5 sets the D– source to channel 5 for bus 2.

BUS:B1:AUTOETHERnet:SOUrce:DMINus? might return BUS:B1:AUTOETHERnet:SOURCE:DMINus CH6, indicating the D– source is set to channel 6 for bus 1.

BUS:B<x>:AUTOETHERnet:SOUrce:DPLUs

This command sets or queries the AutoEthernet D+ source for the specified bus. This command specifies the source channel to use when the signal type is single ended.

Conditions Requires option SR-AUTOENET

Group Bus

Syntax
```
BUS:B<x>:AUTOETHERnet:SOUrce:DPLUs {CH<x>|MATH<x>|REF<x<>}
BUS:B<x>:AUTOETHERnet:SOUrce:DPLUs?
```

Arguments

`Bus<x>` is the Bus number.

`CH<x>` specifies an analog channel as the source for the D+ signal, where <x> is the channel number.

`MATH<x>` specifies a math channel as the source for the D+ signal, where <x> is the math waveform number.

`REF<x>` specifies a reference waveform as the source for the D+ signal, where <x> is the reference waveform number.

Examples

`BUS:B2:AUTOETHERnet:SOUrce:DPLUs CH5` sets the D+ source to channel 5 for bus 2.

`BUS:B1:AUTOETHERnet:SOUrce:DPLUs?` might return `BUS:B1:AUTOETHERnet:SOURCE:DPLUS CH8`, indicating the D+ source is set to channel 8 for bus 1.

BUS:B<x>:AUTOETHERnet:THRESHold

This command sets or queries the AutoEthernet DATA source High threshold level for the specified bus.

Conditions Requires option SR-AUTOENET

Group Bus

Syntax
```
BUS:B<x>:AUTOETHERnet:THRESHold <NR3>
BUS:B<x>:AUTOETHERnet:THRESHold?
```

Related Commands BUS:B<x>:AUTOETHERnet:SOUrce

Arguments

`Bus<x>` is the Bus number.

`NR3` specifies the AutoEthernet DATA source High threshold level for the specified bus, in volts.

Examples

`BUS:B4:AUTOETHERnet:THRESHOLD 1.0` sets the Bus 4 AutoEthernet Data source High threshold level to 1.0 V.

`BUS:B3:AUTOETHERnet:THRESHOLD?` might return `BUS:B3:AUTOETHERnet:THRESHOLD 225.00000E-3`, indicating the Bus 3 AutoEthernet Data source High threshold level is set to 225.0 mV.

BUS:B<x>:AUTOETHERnet:TYPe

This command sets or queries the AutoEthernet standard speed.

Conditions

Requires option SR-AUTOENET

The AutoEthernet signal type is single ended.

Group

Bus

Syntax

```
BUS:B<x>:AUTOETHERnet:TYPe {HUNDREDBASET1}
BUS:B<x>:AUTOETHERnet:TYPe?
```

Arguments

`Bus<x>` is the Bus number.

`HUNDREDBASET1` specifies the AutoEthernet speed as 100Base-T1.

Examples

`BUS:B1:AUTOETHERnet:TYPE HUNDREDBASET1` specifies the AutoEthernet speed as 100Base-T1.

`BUS:B3:AUTOETHERnet:TYPE?` might return `BUS:B3:AUTOETHERnet:TYPE HUNDREDBASET1`, indicating that the AutoEthernet speed is 100Base-T1 for Bus 3.

BUS:B<x>:CAN:BITRate

This command sets or queries the CAN bit rate. The bus number is specified by x. If you select Custom, use BUS:B<x>:CAN:BITRate:VALue to set the bit rate.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:BITRate
{RATE10K|RATE100K|RATE1M|RATE125K|RATE153K|
```

RATE20K|RATE25K|RATE250K|RATE31K|RATE33K|RATE400K|RATE50K|
RATE500K|RATE62K| RATE68K|RATE800K|RATE83K|RATE92K|CUSTom}
BUS:B<x>:CAN:BITRate?

Related Commands BUS:B<x>:CAN:BITRate:VALue

Arguments B<x> is the number of the bus.

Arguments specify the bit rate.

Examples BUS:B1:CAN:BITRate RATE1M sets the CAN bit rate to 1 Mb.

BUS:B1:CAN:BITRate? might return :BUS:B1:CAN:BITRATE RATE500K, indicating the CAN bit rate is set to 500 k.

BUS:B<x>:CAN:BITRate:VALue

This command sets or queries CAN bit rate when Custom is selected by BUS:B<x>:CAN:BITRate. The bus number is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
BUS:B<x>:CAN:BITRate:VALue <NR3>
BUS:B<x>:CAN:BITRate:VALue?

Related Commands BUS:B<x>:CAN:BITRate

Arguments B<x> is the number of the bus.

<NR3> specifies the CAN bit rate.

Returns <NR1> is the bit rate.

Examples BUS:B1:CAN:BITRate:VALue 400.0E+3 sets the bit rate to 400 k.

BUS:B1:CAN:BITRate:VALue? might return :BUS:B1:CAN:BITRATE:VALUE 500000, indicating that the bit rate is set to 500,000.

BUS:B<x>:CAN:FD:BITRate

This command sets or queries the increased data phase bit rate used by CAN FD packets on the specified CAN bus. The bus is specified by x. If you select Custom, use `BUS:B<x>:CAN:FD:BITRate:CUSTom` to set the bit rate.

Conditions Requires option SR-AUTO.

Group Bus

Syntax

```
BUS:B<x>:CAN:FD:BITRate
{RATE1M|RATE2M|RATE3M|RATE4M|RATE5M|RATE6M|
RATE7M|RATE8M|RATE9M|RATE10M|RATE11M|RATE12M|RATE13M|
RATE14M|RATE15M|RATE16M|CUSTom}
BUS:B<x>:CAN:FD:BITRate?
```

Related Commands BUS:B<x>:CAN:FD:BITRate:CUSTom

Arguments `B<x>` is the number of the bus.

Arguments specify the bit rate.

Examples `BUS:B1:CAN:BITRATE RATE1M` sets the CAN FD bit rate to 1 Mbps.

`BUS:B1:CAN:BITRATE?` might return `:BUS:B1:CAN:BITRATE RATE4M`, indicating that the CAN FD bit rate is 4 Mbps.

BUS:B<x>:CAN:FD:BITRate:CUSTom

This command sets or queries the custom bit rate for the increased data phase of CAN FD packets on the specified CAN bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax

```
BUS:B<x>:CAN:FD:BITRate:CUSTom <NR1>
BUS:B<x>:CAN:FD:BITRate:CUSTom?
```

Related Commands BUS:B<x>:CAN:BITRate

Arguments B<x> is the number of the bus.

<NR1> is the custom FD bit rate for the specified bus.

Examples BUS:B1:CAN:FD:BITRATE:CUSTOM 1000000 sets the bit rate to 1,000,000.

BUS:B1:CAN:FD:BITRATE:CUSTOM? might return :BUS:B1:CAN:FD:BITRATE:CUSTOM 4000000, indicating the bit rate is set to 4,000,000.

BUS:B<x>:CAN:SAMPLEpoint

This command sets or queries the sample point for the specified CAN bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:CAN:SAMPLEpoint <NR1>
BUS:B<x>:CAN:SAMPLEpoint?
```

Arguments B<x> is the number of the bus.

<NR1> is the sample point, in percent, for the specified CAN bus.

Examples BUS:B1:CAN:SAMPLEpoint 40 sets the sample point to 40%.

BUS:B1:CAN:SAMPLEpoint? might return :BUS:B1:CAN:SAMPLEPOINT 50 indicating the sample point is at 50%.

BUS:B<x>:CAN:SIGNal

This command sets or queries the signal type for the specified CAN bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax

```
BUS:B<x>:CAN:SIGNal {DIFFerential|CANH|CANL|RX|TX}
BUS:B<x>:CAN:SIGNal?
```

Arguments

B<x> is the number of the bus.

Arguments are the CAN bus signal types.

Examples

BUS:B1:CAN:SINal CANL sets the signal type to CANL.

BUS:B1:CAN:SINal? might return :BUS:B1:CAN:SIGNAL CANH indicating the signal type is CANH.

BUS:B<x>:CAN:SOUrce

This command sets or queries the CAN source channel for the specified CAN bus. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:CAN:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:CAN:SOUrce?
```

Arguments

B<x> is the number of the bus.

Arguments specify a channel.

Examples

BUS:B1:CAN:SOUrce MATH6 sets the source channel to MATH6.

BUS:B1:CAN:SOUrce? might return :BUS:B1:CAN:SOURCE CH1, indicating that the CAN source channel is CH1.

BUS:B<x>:CAN:STANDard

This command sets or queries which CAN standard specification to analyze the specified CAN bus with. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group Bus

Syntax
```
BUS:B<x>:CAN:STANDard {CAN2X|FDISO|FDNONISO}
BUS:B<x>:CAN:STANDard?
```

Arguments `B<x>` is the number of the bus.

`CAN2X` sets the CAN bus standard to CAN 2.0.

`FDISO` sets the CAN bus standard to ISO CAN FD (11898-1:2015).

`FDNONISO` sets the CAN bus standard to non-ISO CAN FD (Bosch:2012).

Examples `BUS:B1:CAN:STANDARD FDISO` sets the CAN standard to ISO CAN FD.

`BUS:B1:CAN:STANDARD?` might return `:BUS:B1:CAN:STANDARD CAN2X`, indicating that the CAN standard is CAN 2.0.

BUS:B<x>:CAN:THReshold

This command sets or queries the source channel threshold for the specified CAN bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:CAN:THReshold <NR3>
BUS:B<x>:CAN:THReshold?
```

Arguments `B<x>` is the number of the bus.

`<NR3>` is the source channel threshold for the specified CAN bus.

Examples `BUS:B1:CAN:THReshold 5` sets the threshold to 5 V.

`BUS:B1:CAN:THReshold?` might return `:BUS:B1:CAN:THRESHOLD 0.0E+0` indicating the threshold is set to 0.0 V.

BUS:B<x>:DISplay:FORMat

This command sets or queries how the data is represented in the bus form for the specified bus. The bus is specified by x.

Group Bus

Syntax

```
BUS:B<x>:DISplay:FORMat
{HEX|BINARY|MIXEDASCII|MIXEDHEX|ASCII|DECIMAL|MIXED}
BUS:B<x>:DISplay:FORMat?
```

Arguments B<x> is the number of the bus.

NOTE. *Different bus types support only a subset of these arguments.*

HEX specifies hexadecimal format.

BINARY specifies binary format.

MIXEDASCII specifies mixed ASCII format.

MIXEDHEX specifies mixed hexadecimal format.

ASCII specifies ASCII format.

DECIMAL specifies decimal format.

MIXED specifies mixed format.

Examples BUS:B1:DISplay:FORMat BINARY sets the format to binary.

BUS:B1:DISplay:FORMat? might return :BUS:B1:DISPLAY:FORMAT HEX indicating the format is set to hexadecimal.

BUS:B<x>:DISplay:LAYout

This command sets or queries what to display for the specified bus. The bus is specified by x.

Group Bus

Syntax

```
BUS:B<x>:DISplay:LAYout {BUS|BUSANDWAVEFORM}
BUS:B<x>:DISplay:LAYout?
```

Arguments

B<x> is the number of the bus.

BUS specifies displaying the bus form only.

BUSANDWAVEFORM specifies displaying the bus form and the constituent source waveform(s). This argument is not available for some bus types and some bus configurations

Examples

BUS:B1:DISplay:LAYout BUS specifies displaying the bus form only.

BUS:B1:DISplay:LAYout? might return :BUS:B1:DISPLAY:LAYOUT BUS indicating the bus will be displayed without displaying its constituent source waveform(s).

BUS:B<x>:DPHY:CLOCk:SOUrce

This command sets or queries the DPHY Clock source for the specified bus line. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x<>}
BUS:B<x>:DPHY:CLOCk:SOUrce?
```

Arguments

Bus<x> is the Bus number.

CH<x> specifies an analog channel as the source, where <x> is the channel number.

MATH<x> specifies a math channel as the source, where <x> is the math waveform number.

REF<x> specifies a reference waveform as the source, where <x> is the reference waveform number.

Examples

BUS:B1:DPHY:CLOCk:SOUrce Ch1 sets the source to channel 1 for bus 1.

BUS:B1:DPHY:CLOCk:SOUrce? might return :BUS:B1:DPHY:CLOCk:SOUrce Ch1, indicating the source is set to channel 1 for bus 1.

BUS:B<x>:DPHY:CLOCk:THRESHold

This command sets or queries the DPHY Clock source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-DPHY.

Group Bus

Syntax

```
BUS:B<x>:DPHY:CLOCk:THRESHold <NR3>
BUS:B<x>:DPHY:CLOCk:THRESHold?
```

Arguments B<x> is the number of the bus.

<NR3> is the DPHY clock high threshold for the specified bus. The argument range is -8V to +8V.

Examples `BUS:B4:DPHY:CLOCk:THRESHold 1.0` sets the Bus 4 DPHY threshold to 1.0 V.

`BUS:B3:DPHY:CLOCk:THRESHold?` might return `:BUS:B3:DPHY:CLOCk:THRESHold 100.00000E-3`, indicating the Bus 3 DPHY clock threshold is set to 100.0 mV.

BUS:B<x>:DPHY:DMINus:DATATHRESHold

This command sets or queries the DPHY D- source data threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-DPHY.

Group Bus

Syntax

```
BUS:B<x>:DPHY:DMINus:DATATHRESHold <NR3>
BUS:B<x>:DPHY:DMINus:DATATHRESHold?
```

Arguments B<x> is the number of the bus.

<NR3> is the DPHY D- source data threshold for the specified bus. The argument range is -8V to +8V.

Examples `BUS:B1:DPHY:DMINus:DATATHRESHold -160.00000E-3` sets the Bus 3 DPHY D- source data threshold to 160.0 mV.

`BUS:B3:DPHY:DMINus:DATATHRESHold?` might return `:BUS:B3:DPHY:DMINus:DATATHRESHold 150.00000E-3`, indicating the Bus 1 DPHY D- source data threshold is set to 150.0 mV.

BUS:B<x>:DPHY:DMINus:LPTHRESHold

This command sets or queries the DPHY D- source low power threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-DPHY.

Group Bus

Syntax

```
BUS:B<x>:DPHY:DMINus:LPTHRESHold <NR3>
BUS:B<x>:DPHY:DMINus:LPTHRESHold?
```

Arguments `B<x>` is the number of the bus.

`<NR3>` is the DPHY D- source low power threshold for the specified bus. The argument range is -8V to +8V.

Examples `BUS:B3:DPHY:DMINus:LPTHRESHold -2.5` sets the Bus 3 DPHY D- source low power threshold to -2.5 V.

`BUS:B2:DPHY:DMINus:LPTHRESHold?` might return `:BUS:B2:DPHY:DMINus:LPTHRESHold -1.2`, indicating the Bus 1 DPHY D- source low power threshold is set to -1.2 V.

BUS:B<x>:DPHY:DMINus:SOUrce

This command sets or queries the DPHY D- source for the specified bus line. The bus is specified by x.

Conditions Requires option SR-DPHY

Group Bus

Syntax

```
BUS:B<x>:DPHY:DMINus:SOUrce {CH<x>|MATH<x>|REF<x<>}
BUS:B<x>:DPHY:DMINus:SOUrce?
```

Arguments

Bus<x> is the Bus number.

CH<x> specifies an analog channel as the source, where <x> is the channel number.

MATH<x> specifies a math channel as the source, where <x> is the math waveform number.

REF<x> specifies a reference waveform as the source, where <x> is the reference waveform number.

Examples

`BUS:B1:DPHY:DMINus:SOUrce Ch3` sets the source to channel 3 for bus 1.

`BUS:B1:DPHY:DMINus:SOUrce?` might return `:BUS:B1:DPHY:DMINus:SOUrce Ch3`, indicating the source is set to channel 3 for bus 1.

BUS:B<x>:DPHY:DPlus:DATATHRESHold

This command sets or queries the D+ source data threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-DPHY.

Group

Bus

Syntax

```
BUS:B<x>:DPHY:DPlus:DATATHRESHold <NR3>
BUS:B<x>:DPHY:DPlus:DATATHRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the DPHY D+ source data threshold for the specified bus. The argument range is -8V to +8V.

Examples

`BUS:B2:DPHY:DPlus:DATATHRESHold 100.00000E-3` sets the Bus 2 DPHY D+ source data threshold to 100.0 mV.

`BUS:B1:DPHY:DPlus:DATATHRESHold?` might return `:BUS:B1:DPHY:DPlus:DATATHRESHold 150.00000E-3`, indicating the Bus 1 DPHY D+ source data threshold is set to 150.0 mV.

BUS:B<x>:DPHY:DPlus:LPTHRESHold

This command sets or queries the DPHY D+ source low power threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-DPHY.

Group Bus

Syntax

```
BUS:B<x>:DPHY:DPlus:LPTHRESHold <NR3>
BUS:B<x>:DPHY:DPlus:LPTHRESHold?
```

Arguments B<x> is the number of the bus.

<NR3> is the DPHY D+ source low power threshold for the specified bus. The argument range is -8V to +8V.

Examples BUS:B3:DPHY:DPlus:LPTHRESHold 1.2 sets the Bus 3 DPHY D+ source low power threshold to 1.2 V.

BUS:B2:DPHY:DPlus:LPTHRESHold? might return :BUS:B2:DPHY:DPlus:LPTHRESHold 1.2, indicating the Bus 1 DPHY D+ source low power threshold is set to 1.2 V.

BUS:B<x>:DPHY:DPlus:SOUrce

This command sets or queries the DPHY D+ source for the specified bus line. The bus is specified by x.

Conditions Requires option SR-DPHY

Group Bus

Syntax

```
BUS:B<x>:DPHY:DPlus:SOUrce {CH<x>|MATH<x>|REF<x><>}
BUS:B<x>:DPHY:DPlus:SOUrce?
```

Arguments Bus<x> is the Bus number.

CH<x> specifies an analog channel as the source, where <x> is the channel number.

MATH<x> specifies a math channel as the source, where <x> is the math waveform number.

REF<x> specifies a reference waveform as the source, where <x> is the reference waveform number.

Examples

BUS:B1:DPHY:DPlus:SOUrce Ch2 sets the source to channel 2 for bus 1.

BUS:B1:DPHY:DPlus:SOUrce? might return :BUS:B1:DPHY:DPlus:SOUrce CH2, indicatingthe source is set to channel 2 for bus 1.

BUS:B<x>:DPHY:LP:DIRection

This command sets or queries the DPHY bus direction in low power. By default lp direction is set to forward. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group

Bus

Syntax

```
BUS:B<x>:DPHY:LP:DIRection {forward|reverse}
BUS:B<x>:DPHY:LP:DIRection?
```

Arguments

Bus<x> is the Bus number.

forward specifies the direction as forward.

reverse specifies the direction as reverse.

Examples

BUS:B1:DPHY:LP:DIRection reverse sets the Bus 1 DPHY bus direction in low power to reverse.

BUS:B2:DPHY:LP:DIRection? might return :BUS:B2:DPHY:LP:DIRection forward, indicating the Bus 2 DPHY direction in low power is forward.

BUS:B<x>:DPHY:PROTocol:TYPe

This command sets or queries the protocol type for DPHY bus decode. The default type is CSI. The bus is specified by x.

Conditions

Requires option SR-DPHY

Group Bus

Syntax
```
BUS:B<x>:DPHY:PROTocol:TYPe {CSI|DSI}
BUS:B<x>:DPHY:PROTocol:TYPe?
```

Arguments `Bus<x>` is the Bus number.

`CSI` specifies the protocol type as CSI.

`DSI` specifies the protocol type as DSI.

Examples `BUS:B1:DPHY:PROTocol:TYPe DSI` sets the Bus 1 DPHY protocol type to DSI.

`BUS:B2:DPHY:PROTocol:TYPe?` might return `:BUS:B2:DPHY:PROTocol:TYPe CSI`, indicating the Bus 2 DPHY protocol type is set to CSI.

BUS:B<x>:DPHY:SIGNal:ENCoding

This command sets or queries the 8b9b encoding for DPHY bus decode. By default 8b9b encoding is set to false. The bus is specified by x.

Conditions Requires option SR-DPHY

Group Bus

Syntax
```
BUS:B<x>:DPHY:SIGNal:ENCoding {false|true}
BUS:B<x>:DPHY:SIGNal:ENCoding?
```

Arguments `Bus<x>` is the Bus number.

`false` specifies the 8b9b encoding disabled.

`true` specifies 8b9b encoding disabled.

Examples `BUS:B1:DPHY:SIGnal:ENCoding true` sets the Bus 1 DPHY 8b9b encoding to true.

`BUS:B2:DPHY:SIGnal:ENCoding?` might return `::BUS:B2:DPHY:SIGnal:ENCoding true`, indicating the Bus 2 DPHY 8b9b encoding is set to true.

BUS:B<x>:ETHERnet:DATAMINUSTHRESHold

This command sets or queries the Ethernet D- source threshold for the specified bus. This threshold only applies when the Ethernet signal type is single ended. The bus is specified by x.

Conditions Requires option SR-ENET.

Group Bus

Syntax

```
BUS:B<x>:ETHERnet:DATAMINUSTHRESHold <NR3>
BUS:B<x>:ETHERnet:DATAMINUSTHRESHold?
```

Related Commands BUS:B<x>:ETHERNET:SOUrce:DMINus

BUS:B<x>:ETHERNET:SOUrce:DPLUs

BUS:B<x>:ETHERnet:SIGNALTYpe

Arguments B<X> is the number of the bus.

<NR3> is the Ethernet D- source threshold for the specified bus.

Examples BUS:B3:ETHERNET:DATAMINUSTHRESHOLD 250.0E-3 sets the Bus 3 Ethernet DATA Minus source threshold to 250.0000 mV.

BUS:B2:ETHERNET:DATAMINUSTHRESHOLD? might return BUS:B2:ETHERNET:DATAMINUSTHRESHOLD 1.0 indicates the Bus 2 Ethernet D- source threshold is set to 1.0 V.

BUS:B<x>:ETHERnet:DATAPLUSTHRESHold

This command sets or queries the Ethernet D+ source threshold for the specified bus. This threshold only applies when the Ethernet signal type is single ended. The bus is specified by x.

Conditions Requires option SR-ENET.

Group Bus

Syntax

```
BUS:B<x>:ETHERnet:DATAPLUSTHRESHold <NR3>
BUS:B<x>:ETHERnet:DATAPLUSTHRESHold?
```

Related Commands

B<x> is the number of the bus.

BUS:B<x>:ETHERNET:SOUrce:DMINus

BUS:B<x>:ETHERNET:SOUrce:DPLUs

BUS:B<x>:ETHERnet:SIGNALTYpe

Arguments

<NR3> is the Ethernet D+ source threshold for the specified bus.

Examples

BUS:B3:ETHERNET:DATAPLUSTHRESHOLD 250.0E-3 sets the Bus 3 Ethernet D+ source threshold to 250.0000 mV.

BUS:B2:ETHERNET:DATAMINUSTHRESHOLD? might return :BUS:B2:ETHERNET:DATAPLUSTHRESHOLD 1.0 indicating the Bus 2 Ethernet D+ source threshold is set to 1.0 V.

BUS:B<x>:ETHERnet:IPVFOUR

This command sets or queries whether IPV4 packets are available for triggering on Ethernet. The bus is specified by x.

Conditions

Requires option SR-ENET.

Group

Bus

Syntax

```
BUS:B<x>:ETHERnet:IPVFOUR {YES|NO}
BUS:B<x>:ETHERnet:IPVFOUR?
```

Arguments

B<x> is the number of the bus.

YES specifies that IPV4 packets are available.

NO specifies that IPV4 packets are not available.

Examples

BUS:B1:ETHERNET:IPVFOUR NO will specify that Bus 1 does not have IPV4 packets available.

BUS:B3:ETHERNET:IPVFOUR? might return :BUS:B3:ETHERNET:IPVFOUR YES indicating that Bus 3 has IPV4 packets available.

BUS:B<x>:ETHERnet:LOWTHRESHold

This command sets or queries the Ethernet source Low threshold for the specified bus. This threshold only applies when the Ethernet signal type is differential. The bus is specified by x.

Conditions Requires option SR-ENET.

Group Bus

Syntax

```
BUS:B<x>:ETHERnet:LOWTHRESHold <NR3>
BUS:B<x>:ETHERnet:LOWTHRESHold?
```

Related Commands BUS:B<x>:ETHERnet:SIGNALTYpe

Arguments B<x> is the number of the bus.

<NR3> is the Ethernet source Low threshold for the specified bus.

Examples `BUS:B1:ETHERnet:LOWTHRESHold -200e-3` sets the threshold to -200 mV.

`BUS:B1:ETHERnet:LOWTHRESHold?` might return `:BUS:B1:ETHERNET:LOWTHRESHOLD -500.0000E-3` indicating the threshold is set to -500 mV.

BUS:B<x>:ETHERnet:QTAGGING

This command sets or queries whether Q-Tagging packets are available for triggering on Ethernet. The bus is specified by x.

Conditions Requires option SR-ENET.

Group Bus

Syntax

```
BUS:B<x>:ETHERnet:QTAGGING {YES|NO}
BUS:B<x>:ETHERnet:QTAGGING?
```

Arguments B<x> is the number of the bus.

YES specifies that Q-Tagging packets are available.

NO specifies that Q-Tagging packets are not available.

Examples BUS:B1:ETHERnet:QTAGGING YES turns on Q-Tagging.

BUS:B1:ETHERnet:QTAGGING? might return :BUS:B1:ETHERNET:QTAGGING YES indicating that Q-tagging packets are available.

BUS:B<x>:ETHERnet:SIGNALTYpe

This command sets or queries the Ethernet signal type for the specified bus. The bus is specified by x.

Conditions Requires option SR-ENET.

Group Bus

Syntax

```
BUS:B<x>:ETHERnet:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:ETHERnet:SIGNALTYpe?
```

Arguments B<x> is the number of the bus.

SINGLE specifies single-ended signals.

DIFF specifies differential signals.

Examples BUS:B1:ETHERnet:SIGNALTYpe SINGLE specifies single-ended signals.

BUS:B1:ETHERnet:SIGNALTYpe? might return :BUS:B1:ETHERNET:SIGNALTYPE DIFF indicating differential signals are specified.

BUS:B<x>:ETHERnet:SOUrce

This command sets or queries the Ethernet data (SDATA) source for the specified bus. This command controls the source channel when the signal type is differential. The bus number is specified by x.

Conditions Requires option SR-ENET.

Group Bus

Syntax

`BUS:B<x>:ETHERnet:SOUrce {CH<x>|MATH<x>|REF<x>}`

`BUS:B<x>:ETHERnet:SOUrce?`

Related Commands

BUS:B<x>:ETHERnet:THRESHold

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies to use one of the analog channels as the Ethernet data source for differential input.

`MATH<x>` specifies to use a math waveform as the source for Ethernet data differential input

`REF<x>` specifies to use one of the reference waveforms as the Ethernet data source for differential input.

Examples

`BUS:B1:ETHERNET:SOURCE CH4` specifies to use the channel 4 waveform as the source for Ethernet data.

`BUS:B1:ETHERNET:SOURCE?` might return `CH2`, indicating that channel 2 is the currently specified source for Ethernet data.

BUS:B<x>:ETHERNET:SOUrce:DMINus

This command sets or queries the Ethernet D- source for the specified bus. this command specifies the source channel to use when the signal type is single ended. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:ETHERNET:SOUrce:DMINus {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ETHERNET:SOUrce:DMINus?
```

Arguments

`B<x>` is the number of the bus.

`CH<x>`, `MATH<x>` or `REF<x>` set the D- source to the specified signal source.

Examples

`BUS:B1:ETHERNET:SOUrce:DMINus CH1` sets the D- source to channel 1.

`BUS:B1:ETHERNET:SOUrce:DMINus?` might return `:BUS:B1:ETHERNET:SOURCE:DMINUS CH2` indicating the `D-` source is set to channel 2.

BUS:B<x>:ETHERNET:SOUrce:DPLUs

This command sets or queries the Ethernet D+ source for the specified bus. this command specifies the source channel to use when the signal type is single ended. The bus is specified by x.

Group Bus

Syntax
```
BUS:B<x>:ETHERNET:SOUrce:DPLUs {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:ETHERNET:SOUrce:DPLUs?
```

Arguments B<x> is the number of the bus.

CH<x>, MATH<x> or REF<x> set the D+ source to the specified signal source.

Examples BUS:B1:ETHERNET:SOUrce:DPLUs Ch5 sets the D+ source to channel 5.

BUS:B1:ETHERNET:SOUrce:DPLUs? might return :BUS:B1:ETHERNET:SOURCE:DPLUS CH5 indicating the D+ source is set to channel 5.

BUS:B<x>:ETHERnet:THRESHold

This command sets or queries the Ethernet DATA source High threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-ENET.

Group Bus

Syntax
```
BUS:B<x>:ETHERnet:THRESHold <NR3>
BUS:B<x>:ETHERnet:THRESHold?
```

Related Commands BUS:B<x>:ETHERnet:SOUrce

Arguments B<x> is the number of the bus.

<NR3> is the Ethernet DATA source High threshold for the specified bus.

Examples BUS:B4:ETHERNET:THRESHOLD 1.0 sets the Bus 4 Ethernet DATA source High threshold to 1.0 V.

BUS:B3:ETHERNET:THRESHOLD? might return BUS:B3:ETHERNET:THRESHOLD 225.00000E-3 indicates the Bus 3 Ethernet DATA source High threshold is set to 225.0 mV.

BUS:B<x>:ETHERnet:TYPe

This command specifies the Ethernet standard speed. The bus number is specified by x.

Conditions Requires option SR-ENET.

Group Bus

Syntax
```
BUS:B<x>:ETHERnet:TYPe {TENBASET|HUNDREDBASETX}
BUS:B<x>:ETHERnet:TYPe?
```

Arguments B<x> is the number of the bus.

TENBASET specifies the Ethernet speed as 10Base-T.

HUNDREDBASETX specifies the Ethernet speed as 100Base-T.

Examples BUS:B1:ETHERNET:TYPE HUNDREDBASETX specifies the Ethernet speed as 100Base-T.

BUS:B1:ETHERNET:TYPE? might return TENBASET, indicating that 10Base-T is the currently specified Ethernet speed.

BUS:B<x>:EUSB:BITRate

This command sets or queries the eUSB data rate for the specified bus.

Conditions Requires option SR-EUSB2.

Group Bus

Syntax
```
BUS:B<x>:EUSB:BITRate {HIGH|FULL|LOW}
BUS:B<x>:EUSB:BITRate?
```

Arguments `B<X>` is the number of the bus.

`HIGH` sets the bit rate to Bit Rate to 480 Mbps.

`FULL` sets the bit rate to Bit Rate to 12 Mbps.

`LOW` sets the bit rate to Bit Rate to 1.5 Mbps.

Examples `BUS:B1:EUSB:BITRate HIGH` sets the bit rate to 480 Mbps.

`BUS:B1:EUSB:BITRate?` might return `:BUS:B1:EUSB:BITRate FULL`, indicating the bit rate is set to 12 Mbps.

BUS:B<x>:EUSB:DATAMINUS:DATA:THRESHold

This command sets or queries the eUSB D- Input Source Data Threshold for Data line decode for specified bus.

Conditions Requires option SR-EUSB2.

Group Bus

Syntax
```
BUS:B<X>:EUSB:DATAMINUS:DATA:THRESHold <NR3>
BUS:B<X>:EUSB:DATAMINUS:DATA:THRESHold?
```

Arguments `B<X>` is the number of the bus.

`<NR3>` is the EUSB Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples `BUS:B1:EUSB:DATAMINUS:DATA:THRESHold 1.0` sets the bus 1 eUSB threshold to 1.0 V.

`BUS:B1:EUSB:DATAMINUS:DATA:THRESHold?` might return `:BUS:B1:EUSB:DATAMINUS:DATA:THRESHold -8.0`, indicating the bus 1 eUSB threshold is set to -8.0V.

BUS:B<x>:EUSB:DATAMINUSTHRESHold

This command sets or queries the eUSB D- Input Source Data Threshold for Data line decode for specified bus.

Conditions Requires option SR-EUSB2.

Group Bus

Syntax
```
BUS:B<x>:EUSB:DATAMINUSTHRESHold <NR3>
BUS:B<x>:EUSB:DATAMINUSTHRESHold?
```

Arguments B<x> is the number of the bus.

<NR3> is the eUSB DATA Minus source threshold. The argument range is -8 V to +8 V.

Examples `BUS:B1:EUSB:DATAMINUSTHRESHold 0.0` sets the bus 1 eUSB threshold to 0.0 V.

`BUS:B1:EUSB:DATAMINUSTHRESHold?` might return `:BUS:B1:EUSB:DATAMINUSTHRESHold -8.0`, indicating the bus 1 eUSB threshold is set to -8.0 V.

BUS:B<x>:EUSB:DATAPLUS:DATA:THRESHold

This command sets or queries the eUSB D+ Input Source Data Threshold for Data line decode for specified bus.

Conditions Requires option SR-EUSB2.

Group Bus

Syntax
```
BUS:B<x>:EUSB:DATAPLUS:DATA:THRESHold <NR3>
BUS:B<x>:EUSB:DATAPLUS:DATA:THRESHold?
```

Arguments B<x> is the number of the bus.

<NR3> is the eUSB Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples `BUS:B1:EUSB:DATAPLUS:DATA:THRESHold 1.0` sets the bus 1 eUSB threshold to 1.0 V.

`BUS:B1:EUSB:DATAPLUS:DATA:THRESHold?` might return `:BUS:B1:EUSB:DATAPLUS:DATA:THRESHold -8.0`, indicating the bus 1 eUSB threshold is set to -8.0V.

BUS:B<x>:EUSB:DATAPLUSTHRESHold

This command sets or queries the eUSB DATA Plus source threshold for the specified bus.

Conditions Requires option SR-EUSB2.

Group Bus

Syntax

```
BUS:B<x>:EUSB:DATAPLUSTHRESHold <NR3>
BUS:B<x>:EUSB:DATAPLUSTHRESHold?
```

Arguments B<x> is the number of the bus.

<NR3> is the eUSB DATA Plus source threshold. The argument range is -8 V to +8 V.

Examples `BUS:B1:EUSB:DATAPLUSTHRESHold 0.0` sets the bus 1 eUSB threshold to 0.0 V.

`BUS:B1:EUSB:DATAPLUSTHRESHold?` might return `:BUS:B1:EUSB:DATAPLUSTHRESHold -8.0V`, indicating the bus 1 eUSB threshold is set to -8.0 V.

BUS:B<x>:EUSB:LOWTHRESHold

This command sets or queries the eUSB source Low threshold for the specified bus when signal type is Differential.

Conditions Requires option SR-EUSB2.

Group Bus

Syntax

```
BUS:B<x>:EUSB:LOWTHRESHold <NR3>
BUS:B<x>:EUSB:LOWTHRESHold?
```

Arguments B<x> is the number of the bus.

<NR3> is the eUSB Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples `BUS:B1:EUSB:LOWTHRESHold 1.0` sets the bus 1 eUSB threshold to 1.0 V.

`BUS:B1:EUSB:LOWTHRESHold?` might return `:BUS:B1:EUSB:LOWTHRESHold -8.0V`, indicating the bus 1 eUSB threshold is set to -8.0V

BUS:B<x>:EUSB:OPERating:MODe

This command sets or queries the eUSB mode for the specified bus.

Conditions Requires option SR-EUSB2.

Group Bus

Syntax
```
BUS:B<x>:EUSB:OPERating:MODe
{NATive|REPEATERHOSt|REPEATERPERIPHERAL}
BUS:B<x>:EUSB:OPERating:MODe?
```

Arguments `B<x>` is the number of the bus.

`NATive` sets the operating mode as native.

`REPEATERHOSt` sets the operating mode as repeater host.

`REPEATERPERIPHERAL` sets the operating mode as repeater peripheral.

Examples `BUS:B1:EUSB:OPERating:MODe NATive` sets the operating mode as native.

`BUS:B1:EUSB:OPERating:MODe?` might return `:BUS:B1:EUSB:OPERating:MODe NATive`, indicating that operating mode is native.

BUS:B<x>:EUSB:SIGNALTYpe

This command sets or queries the eUSB signal type for the specified bus.

Conditions Requires option SR-EUSB2.

Group Bus

Syntax
```
BUS:B<x>:EUSB:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:EUSB:SIGNALTYpe?
```

Arguments `B<x>` is the number of the bus.

Arguments are the available eUSB signal types.

Examples `BUS:B1:EUSB:SIGNALTYpe SINGLE` sets the eUSB signal type to single for bus 1.

`BUS:B1:EUSB:SIGNALTYpe?` might return `:BUS:B1:EUSB:SIGNALTYpe SINGLE`, indicating the eUSB signal type is single for bus 1.

BUS:B<x>:EUSB:SOUrce:DIFF

This command sets or queries the eUSB Diff source for the specified bus when signal type is Diff.

Conditions Requires option SR-EUSB2.

Group Bus

Syntax
```
BUS:B<x>:EUSB:SOUrce:DIFF {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:EUSB:SOUrce:DIFF?
```

Arguments `B<x>` is the number of the bus.

Arguments are the available eUSB sources.

Examples `BUS:B1:EUSB:SOUrce:DIFF Ch1` sets the channel to 1.

`BUS:B1:EUSB:SOUrce:DIFF?` might return `:BUS:B1:EUSB:SOUrce:DIFF CH1`, indicating channel 1 is the source.

BUS:B<x>:EUSB:SOUrce:DMINus

This command sets or queries the eUSB DataMinus (SDATAMINUS) source for the specified bus.

Conditions Requires option SR-EUSB2.

Group Bus

Syntax

```
BUS:B<x>:EUSB:SOUrce:DMINus {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:EUSB:SOUrce:DMINus?
```

Arguments

B<x> is the number of the bus.

Arguments are the available eUSB sources.

Examples

`BUS:B1:EUSB:SOUrce:DMINus CH1` sets the channel to 1.

`BUS:B1:EUSB:SOUrce:DMINus?` might return `:BUS:B1:EUSB:SOUrce:DMINus CH1`, indicating channel 1 is the source.

BUS:B<x>:EUSB:SOUrce:DPLUs

This command sets or queries the eUSB dataPlus (SDATAPLUS) source for the specified bus.

Conditions

Requires option SR-EUSB2.

Group

Bus

Syntax

```
BUS:B<x>:EUSB:SOUrce:DPLUs {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:EUSB:SOUrce:DPLUs?
```

Arguments

B<x> is the number of the bus.

Arguments are the available eUSB sources.

Examples

`BUS:B1:EUSB:SOUrce:DPLUs CH1` sets the channel to 1.

`BUS:B1:EUSB:SOUrce:DPLUs?` might return `:BUS:B1:EUSB:SOUrce:DPLUs CH1`, indicating channel 1 is the source.

BUS:B<x>:EUSB:THRESHold

This command sets or queries the eUSB source High threshold for the specified bus when signal type is Diff.

Conditions

Requires option SR-EUSB2.

Group Bus

Syntax
```
BUS:B<x>:EUSB:THRESHold <NR3>
BUS:B<x>:EUSB:THRESHold ?
```

Arguments B<x> is the number of the bus.

<NR3> is the eUSB Strobe threshold for the specified bus. The argument range is -8 V to +8 V.

Examples BUS:B1:EUSB:THRESHold 1.0 sets the bus 1 eUSB threshold to 1.0 V.

BUS:B1:EUSB:THRESHold? might return :BUS:B1:EUSB:THRESHold -8.0V, indicating the bus 1 eUSB threshold is set to -8.0V

BUS:B<x>:FLEXray:BITRate

This command sets or queries the FlexRay bus bit rate. The bus is specified by x. If you select Custom, use BUS:B<x>:FLEXray:BITRate:CUSTom to set the bit rate.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:FLEXray:BITRate {CUSTOM|RATE2M|RATE5M|RATE10M}
BUS:B<x>:FLEXray:BITRate?
```

Arguments B<x> is the number of the bus.

Arguments specify the bit rate.

Examples BUS:B1:FLEXRAY:BITRate RATE2M sets the bit rate to 2 Mb.

BUS:B1:FLEXRAY:BITRate? might return :BUS:B1:FLEXRAY:BITRATE RATE10M, indicating the bit rate is 10 Mb.

BUS:B<x>:FLEXray:BITRate:CUSTom

This command sets or queries the FlexRay custom bit rate for the specified bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:FLEXray:BITRate:CUSTom <NR1>
BUS:B<x>:FLEXray:BITRate:CUSTom?
```

Arguments B<x> is the number of the bus.

<NR1> is the FlexRay custom bit rate for the specified bus.

Examples `BUS:B1:FLEXray:BITRate:CUSTom 10000000` sets the bit rate to 10,000,000.

`BUS:B1:FLEXray:BITRate:CUSTom?` might return `:BUS:B1:FLEXRAY:BITRATE:CUSTOM 10000000` indicating the bit rate is set to 10,000,000.

BUS:B<x>:FLEXray:CHannel

This command sets or queries the FlexRay channel type for the specified bus. The bus number is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:FLEXray:CHannel {A|B}
BUS:B<x>:FLEXray:CHannel?
```

Arguments B<x> is the number of the bus.

A specifies the A channel.

B specifies the B channel.

Examples `BUS:B1:FLEXRAY:CHANNEL B` sets the FlexRay channel to B.

`BUS:B1:FLEXRAY:CHANNEL?` might return `:BUS:B1:FLEXRAY:CHANNEL A`, indicating the channel is set to A.

BUS:B<x>:FLEXray:LOWTHRESHold

This command sets or queries the FlexRay data source low threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax

```
BUS:B<x>:FLEXray:LOWTHRESHold <NR3>
BUS:B<x>:FLEXray:LOWTHRESHold?
```

Arguments

<NR3> is the FlexRay data source low threshold for the specified bus.

B<x> is the number of the bus.

Examples

`BUS:B1:FLEXray:LOWTHRESHold 50.0e-3` sets the threshold to 50 mV.

`BUS:B1:FLEXray:LOWTHRESHold?` might `:BUS:B1:FLEXray:LOWTHRESHOLD 0.0E+0` indicating the threshold is set to 0.0 V.

BUS:B<x>:FLEXray:SIGnal

This command sets or queries the FlexRay signal type for the specified bus. The bus number is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax

```
BUS:B<x>:FLEXray:SIGnal {BDIFFBP|BM|TXRX}
BUS:B<x>:FLEXray:SIGnal?
```

Arguments

B<x> is the number of the bus.

BDIFFBP sets the FlexRay signal type to BDIFFBP.

BM sets the FlexRay signal type to BM.

TXRX sets the FlexRay signal type to TXRX.

Examples `BUS:B1:FLEXRAY:SIGNAL BM` sets the FlexRay channel type to BM.

`BUS:B1:FLEXRAY:SIGNAL?` might return `:BUS:B1:FLEXRAY:SIGNAL BDIFFBP`, indicating the FlexRay channel type is set to BDIFFBP.

BUS:B<x>:FLEXray:SOUrce

This command sets or queries the Flexray bus data source for the specified bus when the signal type is BDIFFBP or BM. The bus number is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:FLEXray:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:FLEXray:SOUrce?
```

Arguments `B<x>` is the number of the bus.

Arguments are the available FlexRay sources.

Examples `BUS:B1:FLEXRAY:SOUrce MATH4` sets the FlexRay source to Math 4.

`BUS:B1:FLEXRAY:SOUrce?` might return `:BUS:B1:FLEXRAY:SOURCE CH1`, indicating the source is channel 1.

BUS:B<x>:FLEXray:SOUrce:TXRX

This command sets or queries the FlexRay TxRx data source for the specified bus when the signal type is TXRX. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:FLEXray:SOUrce:TXRX
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
```

Arguments Arguments are the available sources.

`B<x>` is the number of the bus.

Examples `BUS:B1:FLEXray:SOUrce:TXRX CH1` sets the TXRX source the channel 1.

`BUS:B1:FLEXray:SOUrce:TXRX?` might return `:BUS:B1:FLEXRAY:SOURCE:TXRX CH1_D0` indicating the TXRX source is set to CH1_D0.

BUS:B<x>:FLEXray:THRESHold

This command sets or queries the FlexRay data source high threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:FLEXray:THRESHold <NR3>
BUS:B<x>:FLEXray:THRESHold?
```

Arguments `B<x>` is the number of the bus.

`<NR3>` is the FlexRay data source high threshold for the specified bus.

Examples `BUS:B1:FLEXray:THRESHold 50.0-3` sets the high threshold to 50 mV.

`BUS:B1:FLEXray:THRESHold?` might return `:BUS:B1:FLEXRAY:THRESHOLD 0.0E+0` indicating the high threshold is set to 0.0 V.

BUS:B<x>:FLEXray:TXRXTHRESHold

This command sets or queries the FlexRay data source TxRx threshold for the specified bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax
```
BUS:B<x>:FLEXray:TXRXTHRESHold <NR3>
BUS:B<x>:FLEXray:TXRXTHRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the TxRx threshold.

Examples

BUS:B1:FLEXray:TXRXTHRESHold 50.0e-3 sets the threshold to 50 mV.

BUS:B1:FLEXray:TXRXTHRESHold? might return :BUS:B1:FLEXRAY:TXRXTHRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

BUS:B<x>:I2C:CLOCk:SOUrce

This command sets or queries the I2C clock (SCLK) source for the specified bus. The bus is specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:I2C:CLOCk:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:I2C:CLOCk:SOUrce?
```

Related Commands

BUS:B<x>:I2C:DATa:SOUrce

BUS:B<x>:I2C:RWINADDR

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel to use as the I2C SCLK source.

CH<x>_D<x> specifies a digital channel to use as the I2C SCLK source.

MATH<x> specifies a math waveform to use as the I2C SCLK source.

REF<x> specifies a reference waveform to use as the I2C SCLK source.

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified I2C bus.

Examples

BUS:B1:I2C:CLOCK:SOURCE CH1 sets the I2C SCLK source to CH1.

BUS:B1:I2C:CLOCK:SOURCE? might return :BUS:B1:I2C:CLOCK:SOURCE CH4_D5, indicating that the I2C SCLK source is set to D5 of FlexChannel 4.

BUS:B<x>:I2C:CLOCk:THReshold

This command sets or queries the I2C Clock (SCLK) source threshold for the specified bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax

```
BUS:B<x>:I2C:CLOCk:THReshold <NR3>
BUS:B<x>:I2C:CLOCk:THReshold?
```

Arguments B<x> is the number of the bus.

<NR3> is the I2C Clock (SCLK) source threshold for the specified bus.

Examples `BUS:B1:I2C:CLOCk:THReshold 50.0e-3` sets the threshold to 50 mV.

`BUS:B1:I2C:CLOCk:THReshold?` might return `:BUS:B1:I2C:CLOCK:THRESHOLD 0.0E+0` indicating the threshold is set to 0 V.

BUS:B<x>:I2C:DATa:SOUrce

This command sets or queries the I2C data (SDA) source for the specified I2C bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax

```
BUS:B<x>:I2C:DATa:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:I2C:DATa:SOUrce?
```

Related Commands BUS:B<x>:I2C:CLOCk:SOUrce

BUS:B<x>:I2C:RWINADDR

Arguments

CH<x> specifies an analog channel to use as the I2C SDA source.

CH<x>_D<x> specifies a digital channel to use as the I2C SDA source.

MATH<x> specifies a math waveform to use as the I2C SDA source.

REF<x> specifies a reference waveform to use as the I2C SDA source.

REF<x>_D<x> specifies a digital reference waveform as the data source waveform for the specified I2C bus.

Examples

BUS:B1:I2C:DATA:SOURCE CH1_D5 sets the I2C SDA source to CH1_D5.

BUS:B1:I2C:DATA:SOURCE? might return :BUS:B1:I2C:DATA:SOURCE MATH1, indicating that the I2C SDA source is set to MATH1.

BUS:B<x>:I2C:DATa:THReshold

This command sets or queries the I2C Data (SDA) source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:I2C:DATa:THReshold <NR3>
BUS:B<x>:I2C:DATa:THReshold?
```

Arguments

B<x> is the number of the bus.

<NR3> is the I2C Data (SDA) source threshold for the specified bus.

Examples

BUS:B1:I2C:DATa:THReshold 50.0e-3 sets the threshold to 50 mV.

BUS:B1:I2C:DATa:THReshold? might return :BUS:B1:I2C:DATA:THRESHOLD 0.0E+0 indicating the threshold is set to 0 V.

BUS:B<x>:I2C:RWINADDR

This command sets or queries the manner in which seven-bit I2C addresses are represented in the busform display of the specified bus. The bus is specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax
```
BUS:B<x>:I2C:RWINADDR {0|1}
BUS:B<x>:I2C:RWINADDR?
```

Related Commands BUS:B<x>:I2C:CLOCk:SOUrce

BUS:B<x>:I2C:DATa:SOUrce

Arguments B<x> is the number of the bus.

0 displays seven-bit slave addresses as integers in the range of 0 to 127, with the state of the R/W* bit from the LSB of the slave address byte. For example, the slave address byte of 0b10100101 is displayed as the value 0x52 R.

1 displays the entire slave address byte as a number, with the R/W* signal as its LSB (bit 0) and the slave address in bits 7..1. For example, the slave address byte of 0b10100101 is displayed as the value 0xA5 R.

Examples BUS:B1:I2C:RWINADDR 0 displays seven-bit slave addresses as integers in the range of 0 to 127.

BUS:B1:I2C:RWINADDR? might return :BUS:B1:I2C:RWINADDR 1, indicating that the entire slave address byte is displayed as a number, with the R/W* signal as its LSB (bit 0) and the slave address in bits 7..1.

BUS:B<x>:I3C:CLOCk:SOUrce

This command sets or queries the I3C clock (SCLK) source for the specified bus.

Requires purchase and installation of option SRI3C.

Group Bus

Syntax
```
BUS:B<x>:I3C:CLOCk:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:I3C:CLOCk:SOUrce?
```

Related Commands BUS:B<x>:I3C:CLOCk:THReshold

Arguments B<x> specifies the bus number.

CH<x> specifies the analog channel to use as the I3C SCLK source.

CH<x>_D<x> specifies the digital channel to use as the I3C SCLK source.

Math<x> specifies the math waveform to use as the I3C SCLK source.

REF<x> specifies the reference waveform to use as the I3C SCLK source.

REF<x>_D<x> specifies the digital reference waveform to use as the I3C SCLK source.

Examples

BUS:B5:I3C:CLOCk:SOURCE CH1 sets the Bus 5 I3C SCLK source to Channel 1.

BUS:B2:I3C:CLOCk:SOURCE? might return BUS:B2:I3C:CLOCK:SOURCE CH4_D5, indicating that the Bus 2 I3C SCLK source is set to Digital signal 5 of Channel 4.

BUS:B<x>:I3C:CLOCk:THReshold

This command sets or queries the I3C clock (SCLK) source threshold level for the specified bus.

Requires purchase and installation of option SRI3C.

Group

Bus

Syntax

```
BUS:B<x>:I3C:CLOCk:THReshold <NR3>
BUS:B<x>:I3C:CLOCk:THReshold?
```

Related Commands

BUS:B<x>:I3C:CLOCk:SOUrce

Arguments

B<x> specifies the bus number.

<NR3> is the threshold value for I3C Clock (SCLK) source of the specified bus.

Examples

BUS:B3:I3C:CLOCk:THReshold 50.0e-3 sets the I3C clock threshold to 50 mV for Bus 3.

BUS:B3:I3C:CLOCk:THReshold? might return BUS:B1:I3C:CLOCK:THRESHOLD 0.0E+0, indicating the threshold is set to 0 V.

BUS:B<x>:I3C:DATa:SOUrce

This command sets or queries the I3C data clock (SDA) source for the specified bus.

Requires purchase and installation of option SRI3C.

Group Bus

Syntax

```
BUS:B<x>:I3C:DATa:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:I3C:DATa:SOUrce?
```

Related Commands BUS:B<x>:I3C:DATa:THReshold

Arguments B<x> specifies the bus number.

CH<x> specifies the analog channel to use as the I3C SDA source.

CH<x>_D<x> specifies the digital channel to use as the I3C SDA source.

Math<x> specifies the math waveform to use as the I3C SDA source.

REF<x> specifies the reference waveform to use as the I3C SDA source.

REF<x>_D<x> specifies the digital reference waveform to use as the I3C SDA source.

Examples BUS:B5:I3C:DATA:SOURCE CH1 sets the Bus 5 I3C SDA clock source to Channel 1.

BUS:B2:I3C:DATA:SOURCE? might return BUS:B2:I3C:DATA:SOURCE MATH1, indicating that the Bus 2 I3C SDA source is set to Math waveform 1.

BUS:B<x>:I3C:DATa:THReshold

This command sets or queries the I3C clock (SDA) data threshold level for the specified bus.

Requires purchase and installation of option SRI3C.

Group Bus

Syntax

```
BUS:B<x>:I3C:DATa:THReshold
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:I3C:DATa:THReshold?
```

Related Commands BUS:B<x>:I3C:DATa:SOUrce

Arguments B<x> specifies the bus number.

<NR3> is the threshold value for I3C Data Clock (SDA) source of the specified bus.

Examples BUS:B3:I3C:DATA:THReshold 50.0e-2 sets the I3C data threshold to 500 mV for Bus 3.

BUS:B3:I3C:DATA:THReshold? might return BUS:B1:I3C:DATA:THRESHOLD 0.0E+0, indicating the Bus 1 I3C SDA threshold is set to 0 V.

BUS:B<x>:LABel:COLor

This command sets or queries the color of the specified bus label. The bus is specified by x.

Group Bus

Syntax
```
BUS:B<x>:LABel:COLor <QString>
BUS:B<x>:LABel:COLor?
```

Arguments B<x> is the number of the bus.

<QString> is the bus label color. To return the color to the default color, send an empty string as in this example: :BUS:B1:LABEL:COLOR "".

Examples BUS:B1:LABel:COLor "#FF0000" sets the label color to red.

BUS:B1:LABel:COLor? might return :BUS:B1:LABEL:COLOR "#FF0000" indicating the color is red.

BUS:B<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified bus label. The bus is specified by x.

Group Bus

Syntax
```
BUS:B<x>:LABel:FONT:BOLD {ON|OFF|1|0}
BUS:B<x>:LABel:FONT:BOLD?
```

Arguments

B<x> is the number of the bus.

ON displays the label in bold font.

OFF does not display the label in bold font.

1 displays the label in bold font.

0 does not display the label in bold font.

Examples

BUS:B1:LABel:FONT:BOLD OFF turns off bold font.

BUS:B1:LABel:FONT:BOLD? might return :BUS:B1:LABEL:FONT:BOLD 1 indicating a bold font.

BUS:B<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified bus label. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:FONT:ITALic {ON|OFF|1|0}
BUS:B<x>:LABel:FONT:ITALic?
```

Arguments

B<x> is the number of the bus.

ON displays the label in italic font.

OFF does not display the label in italic font.

1 displays the label in italic font.

0 does not display the label in italic font.

Examples

BUS:B1LABel:FONT:ITALic OFF turns off italic font.

BUS:B1LABel:FONT:ITALic? might return :BUS:B1:LABEL:FONT:ITALIC 1 indicating the font is italic.

BUS:B<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified bus label. The bus is specified by x.

Group Bus

Syntax
```
BUS:B<x>:LABel:FONT:SIZE <NR1>
BUS:B<x>:LABel:FONT:SIZE?
```

Arguments B<x> is the number of the bus.

<NR1> is the font size.

Examples BUS:B1:LABel:FONT:SIZE 10 sets the font size to 10 points.

BUS:B1:LABel:FONT:SIZE? might return :BUS:B1:LABEL:FONT:SIZE 20 indicating the font size is 20 points.

BUS:B<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified bus label, such as Arial or Times New Roman. The bus is specified by x.

Group Bus

Syntax
```
BUS:B<x>:LABel:FONT:TYPE <QString>
BUS:B<x>:LABel:FONT:TYPE?
```

Arguments B<x> is the number of the bus.

<QString> is the specified font type. Available fonts include: DejaVu Sans, DejaVu Sans Mono, DejaVu Serif, Frutiger LT Std, Monospace, Sans Serif, Serif, Ubuntu, Ubuntu Condensed, and Ubuntu Mono.

Examples BUS:B1:LABel:FONT:TYPE Monospace selects a monospace font.

BUS:B1:LABel:FONT:TYPE? might return :BUS:B1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman" indicating the font type is Frutiger LT Stdn.

BUS:B<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified bus label. The bus is specified by x.

Group Bus

Syntax

```
BUS:B<x>:LABel:FONT:UNDERline {ON|OFF|1|0}
BUS:B<x>:LABel:FONT:UNDERline?
```

Arguments

B<x> is the number of the bus.

ON displays the label in underlined font.

OFF does not display the label in underlined font.

1 displays the label in underlined font.

0 does not display the label in underlined font.

Examples

BUS:B:LABel:FONT:UNDERline ON turns on underline font.

BUS:B:LABel:FONT:UNDERline? might return :BUS:B1:LABEL:FONT:UNDERLINE 0 indicating underline is off.

BUS:B<x>:LABel:name

This command sets or queries the label for the specified bus. The bus is specified by x.

Group

Bus

Syntax

```
BUS:B<x>:LABel:name <QString>
BUS:B<x>:LABel:name?
```

Related Commands

BUS:B<x>:TYPe

Arguments

B<x> is the number of the bus.

<QString> is an alphanumeric string of text enclosed in quotes. The text string is limited to 30 characters. It contains the text label information for the bus.

Examples

BUS:B1:LABEL:NAME "TEST" sets the waveform label for the bus B1 to Test.

BUS:B1:LABEL:NAME? might return :BUS:B1:LABEL:NAME "BUS 1", indicating that the waveform label for the bus B1 is set to "Bus 1".

BUS:B<x>:LABel:XPOS

This command sets or queries the x-position of the specified bus label. The bus is specified by x.

Group Bus

Syntax

```
BUS:B<x>:LABel:XPOS <NR3>
BUS:B<x>:LABel:XPOS?
```

Arguments

B<x> is the number of the bus.

<NR3> is the x-position, in pixels relative to the left edge of the screen of the specified bus label.

Examples

`BUS:B1:LABel:XPOS 90` set the x position to 90.

`BUS:B1:LABel:XPOS?` might return `:BUS:B1:LABEL:XPOS 45.0000` indicating the x position is 45.0 pixels to the right of the left edge of the display.

BUS:B<x>:LABel:YPOS

This command sets or queries the y-position of the specified bus label. The bus is specified by x.

Group Bus

Syntax

```
BUS:B<x>:LABel:YPOS <NR3>
BUS:B<x>:LABel:YPOS?
```

Arguments

B<x> is the number of the bus.

<NR3> is the y-position, in pixels relative to the baseline of the waveform, of the specified bus label.

Examples

`BUS:B1:LABel:YPOS 1.0e1` set the y position to 10.0.

`BUS:B1:LABel:YPOS?` might return `:BUS:B1:LABEL:YPOS 0.0E+0` indicating the y position is 0.0 pixels from the baseline of the waveform.

BUS:B<x>:LIN:BITRate

This command sets or queries the LIN bus bit rate. The bus number is specified by x. If you select Custom, use BUS:B<x>:LIN:BITRate:CUSTom to set the bit rate.

Conditions Requires option SR-AUTO.

Group Bus

Syntax

```
BUS:B<x>:LIN:BITRate
{RATE10K|RATE1K|RATE19K|RATE2K|RATE4K|RATE9K|CUSTom}
BUS:B<x>:LIN:BITRate?
```

Related Commands BUS:B<x>:LIN:BITRate:CUSTom

Arguments `B<x>` is the number of the bus.

Arguments are the available bit rates.

Examples `BUS:B1:LIN:BITRate Rate4k` sets the bit rate to 4 kb.

`BUS:B1:LIN:BITRate?` might return `:BUS:B1:LIN:BITRATE RATE19K`, indicating that the bit rate is set to 19 kb.

BUS:B<x>:LIN:BITRate:CUSTom

This command sets or queries LIN custom bit rate for the specified bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax

```
BUS:B<x>:LIN:BITRate:CUSTom <NR1>
BUS:B<x>:LIN:BITRate:CUSTom?
```

Related Commands BUS:B<x>:LIN:BITRate

Arguments B<x> is the number of the bus.

<NR1> is the LIN custom bit rate for the specified bus.

Examples BUS:B1:LIN:BITRate:CUSTom 9000 sets the bit rate to 9,000.

BUS:B1:LIN:BITRate:CUSTom? might return :BUS:B1:LIN:BITRate:CUSTom 10000 indicating the bit rate is set to 10,000.

BUS:B<x>:LIN:IDFORmat

This command sets or queries LIN bus identifier format for the specified bus. The bus number is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax

```
BUS:B<x>:LIN:IDFORmat {NOPARity|PARity}
BUS:B<x>:LIN:IDFORmat?
```

Arguments B<x> is the number of the bus.

NOPARity specifies an id format that includes parity.

PARity specifies an id format that separates parity.

Examples BUS:B1:LIN:IDFORmat PARITY set the id format to parity.

BUS:B1:LIN:IDFORmat? might return :BUS:B1:LIN:IDFORMAT NOPARITY, indicating the id format is set to no parity.

BUS:B<x>:LIN:POLarity

This command sets or queries the LIN source polarity for the specified bus. The bus number is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax

```
BUS:B<x>:LIN:POLarity {INVerted|NORmal}
BUS:B<x>:LIN:POLarity?
```

Arguments

B<x> is the number of the bus.

INVerted specifies inverted polarity.

NORmal specifies normal polarity.

Examples

`BUS:B1:LIN:POLarity Inverted` sets the polarity to inverted.

`BUS:B1:LIN:POLarity?` might return `:BUS:B1:LIN:POLARITY NORMAL`, indicating the bus polarity is set to normal.

BUS:B<x>:LIN:SAMPLEpoint

Specifies the LIN sample point, for the specified LIN bus. The bus is specified by x.

Conditions

Requires option SR-AUTO.

Group

Bus

Syntax

```
BUS:B<x>:LIN:SAMPLEpoint <NR1>
BUS:B<x>:LIN:SAMPLEpoint?
```

Arguments

B<x> is the number of the bus.

<NR1> is a percentage that represents the point at which to sample during each bit period.

Examples

`BUS:B1:LIN:SAMPLEPOINT 10` sets the sample point to 10% of the bit period

`BUS:B1:LIN:SAMPLEPOINT?` might return `BUS:B1:LIN:SAMPLEPOINT 50` indicating that the sample point is set to 50% of the bit period

BUS:B<x>:LIN:SOUrce

This command sets or queries the LIN data source for the specified bus. The bus number is specified by x.

Conditions

Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:LIN:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:LIN:SOUrce?
```

Arguments B<x> is the number of the bus.

Arguments specify the available sources.

Examples BUS:B1:LIN:SOUrce MATH4 sets the source to MATH 4

BUS:B1:LIN:SOUrce? might return :BUS:B1:LIN:SOURCE CH1, indicating the source is set to channel 1.

BUS:B<x>:LIN:SOUrce:THReshold

This command sets or queries the LIN source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:LIN:SOUrce:THReshold <NR3>
BUS:B<x>:LIN:SOUrce:THReshold?
```

Arguments B<x> is the number of the bus.

<NR3> the LIN source threshold for the specified bus.

Examples BUS:B1:LIN:SOUrce:THReshold 50.0e-3 sets the threshold to 50 mV.

BUS:B1:LIN:SOUrce:THReshold? might return :BUS:B1:LIN:SOURCE:THRESHOLD 0.0E+0 indicating the threshold is 0.0 V.

BUS:B<x>:LIN:STANDard

This command sets or queries the LIN bus standard for the specified bus. The bus number is specified by x.

Conditions Requires option SR-AUTO.

Group Bus

Syntax
```
BUS:B<x>:LIN:STANDard {MIXed|V1X|V2X}
BUS:B<x>:LIN:STANDard?
```

Arguments `B<x>` is the number of the bus.

`MIXed` specifies both versions 1.x and 2.x of the LIN standard.

`V1X` specifies version 1.x of the LIN standard.

`V2X` specifies version 2.x of the LIN standard.

Examples `BUS:B1:LIN:STANDard V1X` sets the standard to version 1.x.

`BUS:B1:LIN:STANDard?` might return `:BUS:B1:LIN:STANDARD V2X`, indicating the standard is set to version 2.x for bus B1.

BUS:B<x>:MANChester:BITORDer

This command sets or queries the Manchester bus Bit Order. The bus is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax
```
BUS:B<x>:MANChester:BITORDer {LSB|MSB}
BUS:B<x>:MANChester:BITORDer?
```

Related Commands BUS:B<x>:MANChester:DISplaymode

Arguments `B<x>` is the number of the bus.

`LSB` arranges the bits in LSB format, least significant bits first.

`MSB` arranges the bits in MSB format, most significant bits first.

Examples `BUS:B1:MANChester:BITORDer LSB` arranges the bits in LSB format for bus 1.

`BUS:B1:MANChester:BITORDer?` might return `:BUS:B1:MANChester:BITORDer MSB`, indicating that bus 1 bits are arranged in MSB format.

BUS:B<x>:MANChester:BITRate

This command sets or queries the Manchester bus bit rate. The bus number is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax

```
BUS:B<x>:MANChester:BITRate <NR1>
BUS:B<x>:MANChester:BITRate?
```

Arguments `B<x>` is the number of the bus.

`<NR1>` sets the bit rate up to 1 Gbps.

Examples `BUS:B1:MANChester:BITRate 125000` sets the bit rate to 125 kb.

`BUS:B1:MANChester:BITRate?` might return `:BUS:B1:MANChester:BITRATE 19000`, indicating that the bit rate is set to 19 kb.

BUS:B<x>:MANChester:DISplaymode

This command sets or queries the Manchester bus Packet View. The bus number is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax

```
BUS:B<x>:MANChester:DISplaymode {BITS|PACKET}
BUS:B<x>:MANChester:DISplaymode?
```

Arguments

B<x> is the number of the bus.

BITS sets the PacketView to off and the data to be seen as single bits formats.

PACKET sets the PacketView to on and the data to be seen in the form of fields.

Examples

BUS:B1:MANChester:DISplaymode PACKET sets the PacketView to off and the data to be seen as single bits formats.

BUS:B1:MANChester:DISplaymode? might return :BUS:B1:MANChester:DISplaymode BITS, indicating the PacketView is on and the data can be seen in the form of fields.

BUS:B<x>:MANChester:HEADer:LENGth

This command sets or queries the Manchester header length in bits. The bus number is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:HEADer:LENGth <NR1>
BUS:B<x>:MANChester:HEADer:LENGth?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode

Arguments

B<x> is the number of the bus.

<NR1> sets the header length in bits.

Examples

BUS:B1:MANChester:HEADer:LENGth 2 sets the header length to 2 bits.

BUS:B1:MANChester:HEADer:LENGth? might return :BUS:B1:MANChester:HEADer:LENGth 2, indicating the header length is 2 bits.

BUS:B<x>:MANChester:IDLE:BITS

This command sets or queries the Manchester idle bit size in bits. The bus number is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax
```
BUS:B<x>:MANChester:IDLE:BITS <NR1>
BUS:B<x>:MANChester:IDLE:BITS?
```

Arguments B<x> is the number of the bus.

<NR1> sets the idle bit size.

Examples BUS:B1:MANChester:IDLE:BITS 2 sets the idle bit size to 2 bits.

BUS:B1:MANChester:IDLE:BITS? might return :BUS:B1:MANChester:IDLE:BITS 2, indicating the idle bit size is 2 bits.

BUS:B<x>:MANChester:parity

This command sets or queries the Manchester bus Parity. The bus number is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax
```
BUS:B<x>:MANChester:parity {ODD|EVEN|NONE}
BUS:B<x>:MANChester:parity?
```

Related Commands BUS:B<x>:MANChester:DISplaymode

Arguments ODD sets the number of 1's to odd.

EVEN sets the number of 1's to even.

NONE specifies that Parity is not considered.

Examples BUS:B1:MANChester:parity ODD sets the number of 1's to odd.

BUS:B1:MANChester:parity? might return :BUS:B1:MANChester:parity EVEN, indicating the number of 1's are even.

BUS:B<x>:MANChester:SOUrce

This command sets or queries the Manchester source for the specified bus. The bus is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax

```
BUS:B<x>:MANChester:SOUrce {CH<x>|Math<x>|REF<x>}
BUS:B<x>:MANChester:SOUrce?
```

Related Commands BUS:B<x>:MANChester:THReshold

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the source.

`Math<x>` specifies a math waveform as the source.

`REF<x>` specifies a reference waveform as the source.

Examples

`BUS:B1:MANChester:SOUrce CH1` sets the source to channel 5.

`BUS:B1:MANChester:SOUrce?` might return `:BUS:B1:MANChester:SOUrce CH5`, indicating the source is set to channel 5.

BUS:B<x>:MANChester:START:INDex

This command sets or queries the Manchester start Index in bits. The bus is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax

```
BUS:B<x>:MANChester:START:INDex <NR1>
BUS:B<x>:MANChester:START:INDex?
```

Arguments B<x> is the number of the bus.

<NR1> sets the start Index in bits.

Examples BUS:B1:MANChester:START:INDex 2 sets the start Index to 2 bits.

BUS:B1:MANChester:START:INDex? might return :BUS:B1:MANChester:START:INDex 2, indicating the start Index is 2 bits.

BUS:B<x>:MANChester:SYNC:SIZe

This command sets or queries the Manchester sync Bit Size in bits. The bus is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax

```
BUS:B<x>:MANChester:SYNC:SIZe <NR1>
BUS:B<x>:MANChester:SYNC:SIZe?
```

Related Commands BUS:B<x>:MANChester:DISplaymode

Arguments B<x> is the number of the bus.

<NR1> sets the Bit Size in bits.

Examples BUS:B1:MANChester:SYNC:SIZe 2 sets the sync Bit Size to 2 bits.

BUS:B1:MANChester:SYNC:SIZe? might return :BUS:B1:MANChester:SYNC:SIZe 2, indicating the sync Bit Size is 2 bits.

BUS:B<x>:MANChester:THReshold

This command sets or queries the Manchester threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax

```
BUS:B<x>:MANChester:THReshold <NR3>
BUS:B<x>:MANChester:THReshold?
```

Related Commands

BUS:B<x>:MANChester:SOUrce

Arguments

B<x> is the number of the bus.

<NR3> sets the Manchester Strobe threshold for the specified bus in Volts. The argument range is -8V to +8V

Examples

`BUS:B4:MANChester:THReshold 1.0` sets the Bus 4 Manchester threshold to 1.0 V.

`BUS:B4:MANChester:THReshold?` might return `:BUS:B4:MANChester:THReshold 225.00000E-3`, indicates the Bus 4 Manchester threshold is set to 225.0 mV.

BUS:B<x>:MANChester:TOLerance

This command sets or queries the Tolerance bus parameter. The bus is specified by x.

Conditions

Requires option SR-MANCH.

Group

Bus

Syntax

```
BUS:B<x>:MANChester:TOLerance <NR3>
BUS:B<x>:MANChester:TOLerance?
```

Arguments

B<x> is the number of the bus.

<NR3> sets the Tolerance bus parameter.

Examples

`BUS:B1:MANChester:TOLerance 20` sets the Tolerance bus parameter to 20 for bus 1.

`BUS:B1:MANChester:TOLerance?` might return `:BUS:B1:MANChester:TOLerance 20`, indicates the Tolerance bus parameter is 20 for bus 1.

BUS:B<x>:MANChester:TRAiler:LENGth

This command sets or queries the Manchester trailer length in bits. The bus is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax

```
BUS:B<x>:MANChester:TRAiler:LENGth <NR1>
BUS:B<x>:MANChester:TRAiler:LENGth?
```

Related Commands BUS:B<x>:MANChester:DISplaymode

Arguments B<x> is the number of the bus.

<NR1> sets the trailer length in bits.

Examples `BUS:B1:MANChester:TRAiler:LENGth 2` sets the trailer length to 2 bits for bus 1.

`BUS:B1:MANChester:TRAiler:LENGth?` might return `:BUS:B1:MANChester:TRAiler:LENGth 2`, indicates the trailer length is 2 bits for bus 1.

BUS:B<x>:MANChester:TRANstion:ZERo

This command sets or queries the Manchester bus for zero falling or rising. Manchester bit are defined by transition in the middle of the bit. Depending on the transition, conventions are defined. The bus is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax

```
BUS:B<x>:MANChester:TRANstion:ZERo {FALLing|RISing}
BUS:B<x>:MANChester:TRANstion:ZERo?
```

Arguments B<x> is the number of the bus.

FALLing sets Falling as 0.

RISing sets Rising as 0.

Examples BUS:B1:MANChester:TRANstion:ZERo FALLing sets Falling as 0 for bus 1.

BUS:B1:MANChester:TRANstion:ZERo? might return :BUS:B1:MANChester:TRANstion:ZERo RISing, indicates that transition of 0 is Rising.

BUS:B<x>:MANChester:WORD:COUNt

This command sets or queries the Manchester word count in bits. The bus is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax
```
BUS:B<x>:MANChester:WORD:COUNt <NR1>
BUS:B<x>:MANChester:WORD:COUNt?
```

Related Commands BUS:B<x>:MANChester:DISplaymode

Arguments B<x> is the number of the bus.

<NR1> sets the word count in bits.

Examples BUS:B1:MANChester:WORD:COUNt 2 sets the word count to 2 bits for bus 1.

BUS:B1:MANChester:WORD:COUNt? might return :BUS:B1:MANChester:WORD:COUNt 2, indicates the word count is 2 bits for bus 1.

BUS:B<x>:MANChester:WORDSIZe

This command sets or queries the Manchester word size in bits. The bus is specified by x.

Conditions Requires option SR-MANCH.

Group Bus

Syntax
```
BUS:B<x>:MANChester:WORDSIZe <NR1>
BUS:B<x>:MANChester:WORDSIZe?
```

Related Commands BUS:B<x>:MANChester:DISplaymode

Arguments B<x> is the number of the bus.

<NR1> sets the word size in bits.

Examples BUS:B1:MANChester:WORDSIZe 12 sets the word size to 2 bits for bus 1.

BUS:B1:MANChester:WORDSIZe? might return :BUS:B1:MANChester:WORDSIZe 12, indicates the word size is 2 bits for bus 1.

BUS:B<x>:MDIO:CLOCk:SOUrce

This command sets or queries the MDIO Clock source for the specified bus. The bus is specified by x.

Conditions Requires option SR-MDIO.

Group Bus

Syntax
```
BUS:B<x>:MDIO:CLOCk:SOUrce {CH<x>|Math<x>|REF<x>}
BUS:B<x>:MDIO:CLOCk:SOUrce?
```

Arguments B<x> is the number of the bus.

CH<x> specifies an analog channel as the Clock source.

Math<x> specifies a math waveform as the Clock source.

REF<x> specifies a reference waveform as the Clock source.

Examples BUS:B1:MDIO:CLOCk:SOUrce CH1 sets the channel to 1 to clock.

BUS:B1:MDIO:CLOCk:SOUrce? might return :BUS:B1:MDIO:CLOCk:SOUrce CH1, indicating that channel 1 is set to clock.

BUS:B<x>:MDIO:CLOCk:THReshold

This command sets or queries the MDIO clock source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-MDIO.

Group Bus

Syntax

```
BUS:B<x>:MDIO:CLOCk:THReshold <NR3>
```

Arguments

B<x> is the number of the bus.

<NR3> is the clock source threshold for the specified bus. The argument range is -8V to +8V.

Examples

`BUS:B1:MDIO:CLOCk:THReshold 0.0` sets the Bus 1 MDIO clock threshold to 0.0V.

`BUS:B1:MDIO:CLOCk:THReshold?` might return `:BUS:B2:MDIO:CLOCk:THReshold -8.0`, indicating the Bus 1 MDIO clock threshold is set to -8.0V.

BUS:B<x>:MDIO:DATA:SOUrce

This command sets or queries the data for the specified bus. The bus is specified by x.

Conditions Requires option SR-MDIO.

Group Bus

Syntax

```
BUS:B<x>:MDIO:DATA:SOUrce {CH<x>|Math<x>|REF<x>}
BUS:B<x>:MDIO:DATA:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the data source.

Math<x> specifies a math waveform as the data source.

REF<x> specifies a reference waveform as the data source.

Examples

`BUS:B1:MDIO:DATA:SOUrce CH2` sets the data source to channel 2.

`BUS:B1:MDIO:DATA:SOUrce?` might return `:BUS:B1:MDIO:DATA:SOUrce CH1`, indicating the data source is set to channel 1.

BUS:B<x>:MDIO:DATA:THReshold

This command sets or queries the MDIO Data source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-MDIO.

Group

Bus

Syntax

`BUS:B<x>:MDIO:DATA:THReshold <NR3>`

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the data source threshold for the specified bus. The argument range is -8V to +8V.

Examples

`BUS:B1:MDIO:DATA:THReshold 0.0` sets the Bus 1 MDIO data threshold to 0.0V.

`BUS:B1:MDIO:DATA:THReshold?` might return `:BUS:B2:MDIO:DATA:THReshold -8.0`, indicating the Bus 1 MDIO data threshold is set to -8.0V.

BUS:B<x>:MIL1553B:LOWTHRESHold

This command sets or queries the MIL-STD-1553 lower threshold for the specified bus. The bus is specified by x.

Conditions

Requires option SR-AERO.

Group

Bus

Syntax

`BUS:B<x>:MIL1553B:LOWTHRESHold <NR3>`

Related Commands

BUS:B<x>:MIL1553B:SOUrce

Arguments B<x> is the number of the bus.

<NR3> is the MIL-STD-1553 lower threshold for the specified bus.

Examples BUS:B3:MIL1553B:LOWTHRESHOLD -200e-3 sets the Bus 3 MIL-STD-1553 lower threshold to -200 mV.

BUS:B2:MIL1553B:LOWTHRESHOLD? might return :BUS:B2:MIL1553B:LOWTHRESHOLD -500.0000E-3 indicating the Bus 2 MIL-STD-1553 lower threshold is set to -500 mV.

BUS:B<x>:MIL1553B:POLarity

This command sets or queries the source polarity for the specified MIL-STD-1553 bus. The bus is specified by x.

Conditions Requires option SR-AERO.

Group Bus

Syntax
```
BUS:B<x>:MIL1553B:POLarity {NORMal|INVERTed}
BUS:B<x>:MIL1553B:POLarity?
```

Arguments B<x> is the number of the bus.

NORMal specifies normal polarity.

INVERTed specifies inverted polarity.

Examples BUS:B3:MIL1553B:POLARITY INVERTED sets Bus 3 MIL-STD-1553 polarity to inverted.

BUS:B2:MIL1553B:POLARITY? might return :BUS:B2:MIL1553B:POLARITY NORMAL, indicating that the Bus 2 MIL-STD-1553 polarity is set to normal.

BUS:B<x>:MIL1553B:RESPonsetime:MAXimum

This command sets or queries the maximum response time to a valid command issued for the specified MIL-STD-1553 bus. The bus is specified by x.

Conditions Requires option SR-AERO.

Group Bus

Syntax
```
BUS:B<x>:MIL1553B:RESPonsetime:MAXimum <NR3>
BUS:B<x>:MIL1553B:RESPonsetime:MAXimum?
```

Related Commands BUS:B<x>:MIL1553B:RESPonsetime:MINimum

Arguments B<x> is the number of the bus.

<NR3> is a floating point number that specifies the maximum response time, in seconds.

Examples BUS:B1:MIL1553B:RESPONSETIME:MAXIMUM 15.0E-6 specifies the maximum response time to a valid command received to be 15.0 microseconds.

BUS:B1:MIL1553B:RESPONSETIME:MAXIMUM? might return :BUS:B1:MIL1553B:RESPONSETIME:MAXIMUM 12.000E-6, indicating a maximum response time of 12 microseconds.

BUS:B<x>:MIL1553B:RESPonsetime:MINimum

This command sets or queries the minimum response time to a valid command issued for the specified MIL-STD-1553 bus. The bus is specified by x.

Conditions Requires option SR-AERO.

Group Bus

Syntax
```
BUS:B<x>:MIL1553B:RESPonsetime:MINimum <NR3>
BUS:B<x>:MIL1553B:RESPonsetime:MINimum?
```

Related Commands BUS:B<x>:MIL1553B:RESPonsetime:MAXimum

Arguments <NR3> is a floating point number that specifies the minimum response time, in seconds.

Examples BUS:B1:MIL1553B:RESPONSETIME:MINIMUM 5.0E-6 specifies the minimum response time to a valid command received to be 5.0 microseconds.

BUS:B1:MIL1553B:RESPONSETIME:MINIMUM? might return :BUS:B1:MIL1553B:RESPONSETIME:MINIMUM 4.000E-6, indicating a minimum response time of 4 microseconds.

BUS:B<x>:MIL1553B:SOUrce

This command sets or queries the source for the specified MIL-STD-1553 bus. The bus is specified by x.

Conditions Requires option SR-AERO.

Group Bus

Syntax

```
BUS:B<x>:MIL1553B:SOUrce {CH<x>|Math<x>|REF<x>}
BUS:B<x>:MIL1553B:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source waveform for the MIL-STD-1553 bus.

Math<x> specifies a math waveform as the source waveform for the MIL-STD-1553 bus.

REF<x> specifies a reference waveform as the source waveform for the MIL-STD-1553 bus.

Examples

BUS:B1:MIL1553B:SOURCE CH1 sets channel 1 as the source for the MIL-STD-1553 bus.

BUS:B1:MIL1553B:SOURCE? might return :BUS:B1:MIL1553B:SOURCE MATH2 indicating that the source is set to MATH2.

BUS:B<x>:MIL1553B:THRESHold

This command sets or queries the MIL-STD-1553 upper threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-AERO.

Group Bus

Syntax

```
BUS:B<x>:MIL1553B:THRESHold <NR3>
BUS:B<x>:MIL1553B:THRESHold
```

Related Commands

BUS:B<x>:MIL1553B:SOUrce

Arguments

B<x> is the number of the bus.

<NR3> is the MIL-STD-1553 upper threshold for the specified bus.

Examples

BUS:B3:MIL1553B:THRESHOLD 2.5 sets the Bus 3 MIL-STD-1553 upper threshold to 2.5 V.

BUS:B2:MIL1553B:THRESHOLD? might return :BUS:B2:MIL1553B:THRESHOLD 500.0000E-3 indicating the Bus 2 MIL-STD-1553 upper threshold is set to 500 mV.

BUS:B<x>:NRZ:BITOrder

This command sets or queries the NRZ bit order for the specified bus. The bus is specified by x.

Conditions

Requires option SRNRZ.

Group

Bus

Syntax

```
BUS:B<x>:NRZ:BITOrder {LSB|MSB}
BUS:B<x>:NRZ:BITOrder?
```

Arguments

B<x> is the number of the bus.

LSB specifies that each bit becomes the recovered value's new LSB, after shifting previously recovered bits one place to the left. The decoding happens right to left.

MSB specifies that each successive bit from the bus's data line becomes the new MSB of the recovered value, shifting any previously recovered bits one place to the right. The decoding happens left to right.

Examples

BUS:B1:NRZ:BITORDER LSB sets each bit order for the NRZ bus B1 to LSB.

BUS:B1:NRZ:BITORDER? might return :BUS:B1:NRZ:BITORDER MSB, indicating that the bit order for the NRZ bus B1 is set to MSB.

BUS:B<x>:NRZ:BITRate

This command sets or queries the NRZ bus bit rate. The bus number is specified by <x>.

Conditions Requires option SRNRZ.

Group Bus

Syntax

```
BUS:B<x>:NRZ:BITRate
BUS:B<x>:NRZ:BITRate?
```

Arguments B<x> is the number of the bus.

Arguments are the available bit rates up to 1 G.

Examples `BUS:B1:NRZ:BITRate 4000` sets the bit rate to 4 kb.

`BUS:B1:NRZ:BITRate?` might return `:BUS:B1:NRZ:BITRATE 19000`, indicating that the bit rate is set to 19 kb.

BUS:B<x>:NRZ:POLarity

This command sets or queries the NRZ source polarity for the specified bus. The bus number is specified by x.

Conditions Requires option SRNRZ.

Group Bus

Syntax

```
BUS:B<x>:NRZ:POLarity {INVerted|NORmal}
BUS:B<x>:NRZ:POLarity?
```

Arguments B<x> is the number of the bus.

INVerted specifies inverted polarity.

NORmal specifies normal polarity.

Examples

`BUS:B1:NRZ:POLarity Inverted` sets the polarity to inverted.

`BUS:B1:NRZ:POLarity?` might return `:BUS:B1:NRZ:POLARITY NORMAL`, indicating the bus polarity is set to normal.

BUS:B<x>:NRZ:SOUrce

This command sets or queries the NRZ source for the specified bus. This command specifies the source channel. The bus is specified by x.

Conditions

Requires option SRNRZ.

Group

Bus

Syntax

```
BUS:B<x>:NRZ:SOUrce {CH<x>|MATH<x>|REF<x>]
BUS:B<x>:NRZ:SOUrce?
```

Related Commands

BUS:B<x>:NRZ:THReshold

Arguments

`B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math waveform as the source.

`REF<x>` specifies a digital reference waveform as the source.

Examples

`BUS:B1:NRZ:SOUrce Ch5` sets the source to channel 5.

`BUS:B1:NRZ:SOUrce?` might return `:BUS:B1:NRZ:SOUrce CH5`, indicating the source is set to channel 5.

BUS:B<x>:NRZ:SPMI:VERsion

This command sets or queries the Version for the specified bus. The bus number is specified by x.

Conditions

Requires option SRPM.

Group

Bus

Syntax

```
BUS:B<x>:NRZ:SPMI:VERsion {v1|v2}
BUS:B<x>:NRZ:SPMI:VERsion?
```

Arguments

B<x> is the number of the bus.

v1 specifies version 1.

v2 specifies version 2.

Examples

BUS:B1:SPMI:VERsion v2 sets to version 2.

BUS:B1:SPMI:VERsion? might return :BUS:B1:SPMI:VERsion v2, indicating version 2 is set.

BUS:B<x>:NRZ:THReshold

This command sets or queries the NRZ threshold for the specified bus. The bus number is specified by x.

Conditions

Requires option SRNRZ.

Group

Bus

Syntax

```
BUS:B<x>:NRZ:THReshold <NR3>
BUS:B<x>:NRZ:THReshold?
```

Related Commands

BUS:B<x>:NRZ:SOUrce

Arguments

B<x> is the number of the bus.

<NR3> is the NRZ Strobe threshold for the specified bus in volts. The valid range is -8 V to +8 V.

Examples

BUS:B4:NRZ:THReshold 1.0 sets the Bus 4 NRZ threshold to 1.0 V.

BUS:B3:NRZ:THReshold? might return :BUS:B4:NRZ:THReshold 225.00000E-3, indicating the Bus 3 NRZ threshold is set to 225.0 mV.

BUS:B<x>:PARallel:ALLTHResholds

This command sets or queries a threshold value for sources for the parallel bus. Use the BUS:B<x>:PARallel:ALLTHResholds:APPly command to set the thresholds to this value. The bus is specified by x.

Group Bus

Syntax `BUS:B<x>:PARallel:ALLTHResholds <NR3>`

Related Commands BUS:B<x>:PARallel:ALLTHResholds:APPly

Arguments `B<x>` is the number of the bus.

`<NR3>` is the source threshold.

Examples `BUS:B4:PARALLEL:ALLTHRESHOLDS` 1.0 sets the threshold of all the sources in parallel Bus 4 to 1.0 V.

`BUS:B3:PARALLEL:ALLTHRESHOLDS?` might return `BUS:B3:PARALLEL:ALLTHRESHOLDS` 500.00000E-3 indicates the threshold for all sources in parallel Bus 3 is currently set to 500.0 mV.

BUS:B<x>:PARallel:ALLTHResholds:APPly (No Query Form)

This command sets all of the data source thresholds to the value set by BUS:B<x>:PARallel:ALLTHResholds for the parallel bus. The bus is specified by x.

Group Bus

Syntax `BUS:B<x>:PARallel:ALLTHResholds:APPly`

Related Commands BUS:B<x>:PARallel:ALLTHResholds

Examples `BUS:B1:PARallel:ALLTHResholds:APPly` sets all data source thresholds to the value set by BUS:B<x>:PARallel:ALLTHResholds.

BUS:B<x>:PARallel:BIT<n>SOUrce

This command sets or queries the specified bit source for specified parallel bus. The bus is specified by x. The bit is specified by n and is an integer in the range of 1 to 64.

Group Bus

Syntax

```
BUS:B<x>:PARallel:BIT<n>SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>|NONE}
```

Related Commands BUS:B<x>:PARallel:BIT<n>SOUrce:THReshold

Arguments

B<x> is the number of the bus.

CH<x> is the specified bit source.

CH<x>_D<x> is the specified bit source.

MATH<x> is the specified bit source.

REF<x> is the specified bit source.

REF<x>_D<x> specifies a digital reference waveform as the bit<x> source waveform for the specified parallel bus.

NONE disables the bit source.

Examples

BUS:B1:PARallel:BIT1SOUrce CH1 sets the bit 1 source to channel 1.

BUS:B1:PARallel:BIT1SOUrce? might return :BUS:B1:PARALLEL:BIT1SOURCE CH1_D0 indicating the bit1 source is CH1_D0.

BUS:B<x>:PARallel:BIT<n>SOUrce:THReshold

This command sets or queries the specified bit source threshold for the specified parallel bus. The bus is specified by x. The bit is specified by n and is an integer in the range of 1 to 64.

Group Bus

Syntax

```
BUS:B<x>:PARallel:BIT<n>SOUrce:THReshold <NR3>
```

Related Commands BUS:B<x>:PARallel:BIT<n>SOUrce

Arguments B<x> is the number of the bus.

<NR3> is the specified bit source threshold for the specified parallel bus.

Examples BUS:B3:PARALLEL:BIT2SOURCE:THRESHOLD 1.0 sets the threshold of bit source 2 of parallel Bus 3 to 1.0 V.

BUS:B1:PARALLEL:BIT4SOURCE:THRESHOLD? might return :BUS:B1:PARALLEL:BIT4SOURCE:THRESHOLD 250.00000E-3 indicates the threshold of bit source 4 of parallel Bus 1 is currently set to 250.0 mV.

BUS:B<x>:PARallel:CLOCk:EDGE

This command sets or queries the clock edge for the parallel bus. The bus is specified by x.

Group Bus

Syntax
```
BUS:B<x>:PARallel:CLOCk:EDGE {FALLING|RISING|EITHER}
BUS:B<x>:PARallel:CLOCk:EDGE?
```

Related Commands BUS:B<x>:PARallel:CLOCkSOUrce

Arguments B<x> is the number of the bus.

FALLING decodes on the falling edge of the clocked parallel bus signal.

RISING decodes on the rising edge of the clocked parallel bus signal.

EITHER decodes on the rising or falling edge of the clocked parallel bus signal.

Examples BUS:B1:PARALLEL:CLOCK:EDGE FALLING sets the decoding to happen on the falling edge of its clocked parallel bus B1.

BUS:B1:PARALLEL:CLOCK:EDGE? might return :BUS:B1:PARALLEL:CLOCK:EDGE RISING, indicating that when bus B1 operates in Clocked Parallel mode, it samples new data values on the rising edge of its clock source signal.

BUS:B<x>:PARallel:CLOCk:ISCLOCKED

This command determines whether the bus operates in a clocked or asynchronous fashion. The bus is specified by x.

Group Bus

Syntax

```
BUS:B<x>:PARallel:CLOCk:ISCLOCKED {OFF|ON|NR1>
BUS:B<x>:PARallel:CLOCk:ISCLOCKED?
```

Arguments

B<x> is the number of the bus.

OFF argument specifies an asynchronous bus.

ON argument specifies a clocked bus.

<NR1> = 0 specifies an asynchronous bus; any other value specifies a clocked bus.

Examples

BUS:B1:PARallel:CLOCk:ISCLOCKED 0 sets the bus to operate asynchronously.

BUS:B1:PARallel:CLOCk:ISCLOCKED? might return :BUS:B1:PARALLEL:CLOCK:ISCLOCKED 1 indicating the bus is clocked.

BUS:B<x>:PARallel:CLOCkSOUrce

This command sets or queries the Parallel clock bit source for the specified bus. The bus is specified by x.

Group Bus

Syntax

```
BUS:B<x>:PARallel:CLOCkSOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>|NONE}
BUS:B<x>:PARallel:CLOCkSOUrce?
```

Related Commands BUS:B<x>:PARallel:CLOCk:ISCLOCKED

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog FlexChannel to use as the bus clock source.

CH<x>_D<x> specifies a digital channel on a specified FlexChannel to use as the bus clock source.

MATH<x> specifies the math channel to use as the bus clock source.

REF<x> specifies the reference channel to use as the bus clock source.

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified parallel bus.

NONE specifies the reference channel to use as the bus clock source.

Examples

BUS:B1:PARALLEL:CLOCK:SOURCE CH3_D6 sets the Parallel clock source for the bus B1 to D6 of FlexChannel 3.

BUS:B1:PARALLEL:CLOCK:SOURCE? might return :BUS:B1:PARALLEL:CLOCK:SOURCE CH8_D9, indicating that the Parallel clock source for the bus B1 is set to D9 of FlexChannel 8.

BUS:B<x>:PARallel:CLOCKSOUrce:THReshold

This command sets or queries the clock source threshold for the parallel bus. The bus is specified by x.

Group

Bus

Syntax

BUS:B<x>:PARallel:CLOCKSOUrce:THReshold <NR3>
BUS:B<x>:PARallel:CLOCKSOUrce:THReshold?

Related Commands

BUS:B<x>:PARallel:CLOCk:ISCLOCKED

Arguments

<NR3> is the clock bit source threshold for the parallel bus.

Examples

BUS:B4:PARALLEL:CLOCKSOUrce:THRESHOLD 1.5 sets the threshold of the clock source of parallel Bus 4 to 1.5 V.

BUS:B3:PARALLEL:CLOCKSOUrce:THRESHOLD? might return :BUS:B3:PARALLEL:CLOCKSOUrce:THRESHOLD 750.00000E-3 indicates the threshold of the clock source of parallel Bus 3 is currently set to 750.0 mV.

BUS:B<x>PSIFIVe:BITPERiod

This command sets or queries the PSI5 Bit period bus parameter.

Conditions Requires option SR-PSI5

Group Bus

Syntax
```
BUS:B<X>PSIFIVe:BITPERiod <NR1>
BUS:B<X>PSIFIVe:BITPERiod?
```

Arguments B<X> is the number of the bus.

<NR1> specifies the bit period. The default bit period is 60 Micro seconds.

Examples BUS:B1:PSIFIVe:BITPERiod 0.000060 sets the bit period to 60 µs.

BUS:B1:PSIFIVe:BITPERiod? might return :BUS:B1:PSIFIVe:BITPERiod 0.000120, indicating that the bit rate is set to 120 µs.

BUS:B<x>:PSIFIVe:BITRate

This command sets or queries the PSI5 bitrate.

Conditions Requires option SR-PSI5

Group Bus

Syntax
```
BUS:B<X>:PSIFIVe:BITRate {RATE125K|RATE189K|RATE83K}
BUS:B<X>:PSIFIVe:BITRate?
```

Arguments B<X> is the number of the bus.

Arguments are the available bit rates.

Examples BUS:B1:PSIFIVe:BITRate RATE125k sets the bit rate to 125 kb.

BUS:B1:PSIFIVe:BITRate? might return :BUS:B1:PSIFIVe:BITRATE RATE189K, indicating that the bit rate is set to 189 kb.

BUS:B<x>PSIFIVe:COMM:DIRection

This command sets or queries the PSI5 bus communication direction. Communication direction by default is set to Sensor to ECU.

Conditions Requires option SR-PSI5

Group Bus

Syntax
```
BUS:B<x>PSIFIVe:COMM:DIRection {SENSORECU|ECUSENSor}
BUS:B<x>PSIFIVe:COMM:DIRection?
```

Arguments B<x> is the number of the bus.

SENSORECU specifies the communication direction to Sensor to ECU. This is the default value.

ECUSENSor specifies the communication direction to ECU to Sensor.

Examples BUS:B1:PSIFIVe:COMM:DIRection SENSORECU sets the communication direction to Sensor to ECU.

BUS:B1:PSIFIVe:COMM:DIRection? might return :BUS:B1:PSIFIVe:COMM:DIRection ECUSENSor, indicating that the communication direction is ECU to Sensor .

BUS:B<x>:PSIFIVe:DATAA

This command sets or queries the PSI5 frame mandatory data region A.

Conditions Requires option SR-PSI5

Group Bus

Syntax
```
BUS:B<x>:PSIFIVe:DATAA <NR1>
BUS:B<x>:PSIFIVe:DATAA?
```

Arguments B<x> is the number of the bus.

<NR1> specifies the dataA value in bits. The default dataA value is 10 bits, otherwise it ranges between 10 to 24 bits.

Examples BUS:B1:PSIFIVe:DATAA 17 sets the dataA as 17 bits.

BUS:B1:PSIFIVe:DATAA? might return :BUS:B1:PSIFIVe:DATAA 10, indicating that the dataA is 10 bits.

BUS:B<x>:PSIFIVe:DATAB

This command sets or queries the PSI5 frame optional bits of data region B.

Conditions Requires option SR-PSI5

Group Bus

Syntax

```
BUS:B<x>:PSIFIVe:DATAB <NR1>
BUS:B<x>:PSIFIVe:DATAB?
```

Arguments

`B<x>` is the number of the bus.

`<NR1>` specifies the dataA value in bits. The default dataB value is 0 bits, otherwise it ranges between 0 to 12 bits.

Examples

`BUS:B1:PSIFIVe:DATAB 7` sets the dataB as 7 bits.

`BUS:B1:PSIFIVe:DATAB?` might return `:BUS:B1:PSIFIVe:DATAB 0`, indicating that the dataB is 0 bits.

BUS:B<x>:PSIFIVe:DATAFORMat

This command sets or queries the data format in PSI5 Frame2 packet.

Conditions Requires option SR-PSI5

Group Bus

Syntax

```
BUS:B<x>:PSIFIVe:DATAFORMat {NIBBLe|BYTe}
BUS:B<x>:PSIFIVe:DATAFORMat?
```

Arguments

`B<x>` is the number of the bus.

`NIBBLe` specifies the data format as Nibble.

`BYTe` specifies the data format as Byte.

Examples `BUS:B1:PSIFIVe:DATAFORMat NIBBLe` sets the data format to Nibble.

`BUS:B1:PSIFIVe:DATAFORMat?` might return `:BUS:B1:PSIFIVe:DATAFORMat BYTe`, indicating that the data format is Byte.

BUS:B<x>:PSIFIVe:ECUSOURce

This command sets or queries the ECU to sensor source channel for the specified Bus.

Conditions Requires option SR-PSI5

Group Bus

Syntax
```
BUS:B<x>:PSIFIVe:ECUSOURce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:PSIFIVe:ECUSOURce?
```

Arguments `B<x>` is the number of the bus.

`CH<x>` specifies an analog channel as the source.

`MATH<x>` specifies a math waveform as the source.

`REF<x>` specifies a digital reference waveform as the source.

Examples `BUS:B1:PSIFIVe:ECUSOURce CH5` sets the source to channel 5.

`BUS:B1:PSIFIVe:ECUSOURce?` might return `:BUS:B1:PSIFIVe:ECUSOURce CH5`, indicating the source is set to channel 5.

BUS:B<x>:PSIFIVe:FRAMECONTrol

This command sets or queries the PSI5 frame optional control bits.

Conditions Requires option SR-PSI5

Group Bus

Syntax
```
BUS:B<x>:PSIFIVe:FRAMECONTrol <NR1>
BUS:B<x>:PSIFIVe:FRAMECONTrol?
```

Arguments B<x> is the number of the bus.

<NR1> specifies the frame control value in bits. The default frame control value is 0 bits, otherwise it ranges between 0 to 4 bits.

Examples BUS:B1:PSIFIVe:FRAMECONTrol 2 sets the frame control as 2 bits.

BUS:B1:PSIFIVe:FRAMECONTrol? might return :BUS:B1:PSIFIVe:FRAMECONTrol 3, indicating that the frame control is 3 bits.

BUS:B<x>:PSIFIVe:MESSaging

This command sets or queries the PSI5 optional messaging bits.

Conditions Requires option SR-PSI5

Group Bus

Syntax
```
BUS:B<x>:PSIFIVe:MESSaging {OFF|ON}
BUS:B<x>:PSIFIVe:MESSaging?
```

Arguments B<x> is the number of the bus.

OFF specifies the messaging bits as the default value of 0.

ON specifies the messaging bits as 2.

Examples BUS:B1:PSIFIVe:MESSaging OFF sets the messaging bits as 0.

BUS:B1:PSIFIVe:MESSaging? might return :BUS:B1:PSIFIVe:MESSaging ON, indicating that the messaging bits are 2.

BUS:B<x>:PSIFIVe:SOUrce

This command sets or queries serial channel on or off.

Conditions Requires option SR-PSI5

Group Bus

Syntax

```
BUS:B<x>:PSIFIVe:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:PSIFIVe:SOUrce?
```

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a digital reference waveform as the source.

Examples

BUS:B1:PSIFIVe:SOUrce CH5 sets the source to channel 5.

BUS:B1:PSIFIVe:SOUrce? might return :BUS:B1:PSIFIVe:SOUrce CH5, indicating the source is set to channel 5.

BUS:B<x>:PSIFIVe:STATus

This command sets or queries the optional status bits of PSI5.

Conditions

Requires option SR-PSI5

Group

Bus

Syntax

```
BUS:B<x>:PSIFIVe:STATus <NR1>
BUS:B<x>:PSIFIVe:STATus?
```

Arguments

B<x> is the number of the bus.

<NR1> specifies the status value in bits. The default status value is 0 bits, otherwise it ranges between 0 to 2 bits.

Examples

BUS:B1:PSIFIVe:STATus 2 sets the status as 2 bits.

BUS:B1:PSIFIVe:STATus? might return :BUS:B1:PSIFIVe:STATus 0, indicating that the status is 0 bits.

BUS:B<x>:PSIFIVe:SYNCMODe

This command sets or queries the PSI5 Sync Mode.

Conditions

Requires option SR-PSI5

Group Bus

Syntax
```
BUS:B<x>:PSIFIVe:SYNCMODe {PULSEWIDTh|TOOTHGAP}
BUS:B<x>:PSIFIVe:SYNCMODe?
```

Arguments `B<x>` is the number of the bus.

`PULSEWIDTh` specifies the Sync Mode as Pulse Width.

`TOOTHGAP` specifies the Sync Mode as Tooth Gap.

Examples `BUS:B1:PSIFIVe:SYNCMODe PULSEWIDTh` sets the Sync Mode to Pulse Width.

`BUS:B1:PSIFIVe:SYNCMODe?` might return `:BUS:B1:PSIFIVe:SYNCMODe TOOTHGAP`, indicating that the Sync Mode is Tooth Gap.

BUS:B<x>:PSIFIVe:SYNCTHRESHold

This command sets or queries the PSI5 threshold for the ECU To Sensor specified bus.

Conditions Requires option SR-PSI5

Group Bus

Syntax
```
BUS:B<x>:PSIFIVe:SYNCTHRESHold <NR3>
BUS:B<x>:PSIFIVe:SYNCTHRESHold?
```

Arguments `B<x>` is the number of the bus.

`<NR3>` specifies the PSI5 Strobe threshold for the specified bus. The threshold range is -8 V to +8 V.

Examples `BUS:B1:PSIFIVe:SYNCTHRESHold 1.0` sets the bus 4 PSI5 threshold to 1.0 V.

`BUS:B1:PSIFIVe:SYNCTHRESHold?` might return `:BUS:B1:PSIFIVe:SYNCTHRESHold 225.00000E-3`, indicating the bus 3 PSIFIVe threshold is set to 225.0 mV.

BUS:B<x>:PSIFIVe:THRESHold

This command sets or queries the PSI5 threshold for the Sensor To ECU specified bus.

Conditions Requires option SR-PSI5

Group Bus

Syntax

```
BUS:B<x>:PSIFIVe:THRESHold <NR3>
BUS:B<x>:PSIFIVe:THRESHold?
```

Arguments

B<x> is the number of the bus.

<NR3> specifies the PSI5 Strobe threshold for the specified bus. The threshold range is -8 V to +8 V.

Examples

`BUS:B1:PSIFIVe:THRESHold 1.0` sets the bus 4 PSI5 threshold to 1.0 V.

`BUS:B1:PSIFIVe:THRESHold?` might return `:BUS:B1:PSIFIVe:THRESHold 225.00000E-3`, indicating the bus 4 PSI5 threshold is set to 225.0 mV.

BUS:B<x>:RS232C:BITRate

This command sets or queries the RS-232C bit rate for bus<x>, where the bus number is specified by x. If you select Custom, use BUS:B<x>:RS232C:BITRate:CUSTom to set the bit rate.

Conditions Requires option SR-COMP.

Group Bus

Syntax

```
BUS:B<x>:RS232C:BITRate
{CUSTOM|RATE300|RATE1K|RATE2K|RATE9K|
RATE19K|RATE38K|RATE115K|RATE921K}
BUS:B<x>:RS232C:BITRate?
```

Arguments

B<x> is the number of the bus.

Arguments are the available bit rates.

Examples `BUS:B1:RS232C:BITRATE RATE9K` sets the bit rate for the RS-232C bus B1 to 9000 bits-per-second.

`BUS:B1:RS232C:BITRATE?` might return `:BUS:B1:RS232C:BITRATE 9000`, indicating that the bit rate for the RS-232C bus B1 is set to 9000 bits-per-second.

BUS:B<x>:RS232C:BITRate:CUSTom

This command sets or queries the RS-232C custom bit rate for the specified bus. The bus is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax

```
BUS:B<x>:RS232C:BITRate:CUSTom <NR1>
BUS:B<x>:RS232C:BITRate:CUSTom?
```

Related Commands BUS:B<x>:RS232C:BITRate

Arguments `B<x>` is the number of the bus.

`<NR1>` is the custom bit rate.

Examples `BUS:B1:RS232C:BITRATE:CUSTOM 9500` sets the bit rate for the RS-232C bus B1 to 9500 bits-per-second.

`BUS:B1:RS232C:BITRATE:CUSTOM?` might return `:BUS:B1:RS232C:BITRATE:CUSTOM 9500`, indicating that the bit rate for the RS-232C bus B1 is set to 9500 bits-per-second.

BUS:B<x>:RS232C:DATABits

This command sets or queries the RS-232C data width for bus<x>, where the bus number is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax

```
BUS:B<x>:RS232C:DATABits {7|8|9}
BUS:B<x>:RS232C:DATABits?
```

Arguments

B<x> is the number of the bus.

<NR1> specifies the number of bits in the RS-232C data frame.

Examples

BUS:B1:RS232C:DATABITS 8 sets the data bits to 8 for the RS-232C bus B1.

BUS:B5:RS232C:DATABITS? might return :BUS:B1:RS232C:DATABITS 7, indicating that the data bits is set to 7 for the RS-232C bus B5.

BUS:B<x>:RS232C:DELIMiter

This command sets or queries the RS-232C string delimiter on bus <x>, where the bus number is specified by x. This command only applies when Packet view is turned On.

Conditions

Requires option SR-COMP.

Group

Bus

Syntax

```
BUS:B<x>:RS232C:DELIMiter {NULl|CR|LF|SPace|XFF}
BUS:B<x>:RS232C:DELIMiter?
```

Related Commands

BUS:B<x>:RS232C:DISplaymode

Arguments

B<x> is the number of the bus.

NULl specifies NULL (0x00) delimiting value for a packet.

CR specifies CR (0x0D) delimiting value for a packet.

LF specifies LF (0x0A) delimiting value for a packet.

XFF specifies XFF (0xFF) delimiting value for a packet.

SPace specifies SPace delimiting value for a packet.

Examples

BUS:B1:RS232C:DELIMITER LF sets the delimiter to 0x0A for the RS-232C bus B1.

BUS:B1:RS232C:DELIMITER? might return :BUS:B1:RS232C:DELIMITER NULL, indicating that the delimiter to 0x00 for the RS-232C bus B1.

BUS:B<x>:RS232C:DISplaymode

This command sets or queries the RS-232C display mode for the specified bus. The bus is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax

```
BUS:B<x>:RS232C:DISplaymode {FRame|PACKET}
BUS:B<x>:RS232C:DISplaymode?
```

Related Commands BUS:B<x>:RS232C:DELIMiter

Arguments

`B<x>` is the number of the bus.

`FRame` displays each frame as a single entity.

`PACKET` displays a group of frames terminated with a single frame defined by the `BUS:B<x>:RS232C:DELImiter` command.

Examples

`BUS:B1:RS232C:DISPLAYMODE FRAME` sets the display mode for the RS-232C bus B1 to Frame.

`BUS:B1:RS232C:DISPLAYMODE?` might return `:BUS:B1:RS232C:DISPLAYMODE PACKET`, indicating that the display mode for the RS-232C bus B1 is set to Packet.

BUS:B<x>:RS232C:PARity

This command sets or queries the RS-232C parity for bus <x>, where the bus number is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax

```
BUS:B<x>:RS232C:PARity {NONe|EVEN|ODD}
BUS:B<x>:RS232C:PARity?
```

Arguments B<x> is the number of the bus.

NONe specifies no parity.

EVEN specifies even parity.

ODD specifies odd parity.

Examples BUS:B1:RS232C:PARITY ODD sets the parity for the RS-232C bus B1 to odd.

BUS:B1:RS232C:PARITY? might return :BUS:B1:RS232C:PARITY NONE, indicating that the display mode for the RS-232C bus B1 is set to none.

BUS:B<x>:RS232C:POLarity

This command sets or queries the RS-232C source polarity for bus <x>, where the bus number is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax
```
BUS:B<x>:RS232C:POLarity {NORmal|INVERTed}
BUS:B<x>:RS232C:POLarity?
```

Arguments B<x> is the number of the bus.

NORmal sets the RS-232C bus polarity to positive.

INVERTed sets the RS-232C bus polarity to negative.

Examples BUS:B1:RS232C:POLARITY INVERTed sets the polarity for the RS-232C bus B1 to Inverted.

BUS:B1:RS232C:POLARITY? might return :BUS:B1:RS232C:POLARITY NORMAL, indicating that the polarity for the RS-232C bus B1 is set to Normal.

BUS:B<x>:RS232C:SOUrce

This command sets or queries the RS-232C source for bus <x>, where the bus number is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax
```
BUS:B<x>:RS232C:SOUrce
{CH<x>|CH<x>_D<x>|REF<x>|MATH<x>|REF<x>_D<x>}
BUS:B<x>:RS232C:SOUrce?
```

Arguments B<x> is the number of the bus.

CH<x> specifies an analog channel to use as the RS-232C source.

CH<x>_D<x> specifies a digital channel of a specified FlexChannel to use for the RS-232C source.

MATH<x> specifies a math channel to use for the RS-232C source.

REF<x> specifies a reference channel to use for the RS-232C source.

REF<x>_D<x> specifies a digital reference waveform as the source waveform for the specified RS-232C bus.

Examples BUS:B1:RS232C:SOURCE CH1_D0 sets the source for the RS-232C bus B1 to D0 of FlexChannel 1.

BUS:B1:RS232C:SOURCE? might return :BUS:B1:RS232C:SOURCE CH1, indicating that the source for the RS-232C bus B1 is CH1.

BUS:B<x>:RS232C:SOUrce:THReshold

This command sets or queries the RS-232C source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-COMP.

Group Bus

Syntax
```
BUS:B<x>:RS232C:SOUrce:THReshold <NR3>
BUS:B<x>:RS232C:SOUrce:THReshold?
```

Arguments B<x> is the number of the bus.

<NR3> is the RS-232C source threshold for the specified bus.

Examples `BUS:B1:RS232C:SOURCE:THRESHOLD 50.0e-3` sets the threshold to 50 mV.

`BUS:B1:RS232C:SOURCE:THRESHOLD?` might return `:BUS:B1:RS232C:SOURCE:THRESHOLD 0.0E+0` indicating the threshold is set to 0.0 V.

BUS:B<x>:S8B10B:BITRate

This command sets or queries the 8B10b bit rate for the specified bus, where the bus is specified by x.

Conditions Requires option SR8B10b.

Group Bus

Syntax

```
BUS:B<x>:S8B10B:BITRate
BUS:B<x>:S8B10B:BITRate?
```

Arguments `B<x>` is the number of the bus.

Arguments are the available bit rates up to 1 Tbps.

Examples `BUS:B1:S8B10B:BITRate 4000` sets the bit rate to 4 kb.

`BUS:B1:S8B10B:BITRate?` might return `:BUS:B1:S8B10B:BITRATE 19000`, which indicates that the bit rate is set to 19 kb.

BUS:B<x>:S8B10B:SOUrce

This command sets or queries the 8B10b source for the specified bus, where the bus is specified by x. This command specifies the source channel.

Conditions Requires option SR8B10b.

Group Bus

Syntax

```
BUS:B<x>:S8B10B:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:S8B10B:SOUrce?
```

Related Commands BUS:B<x>:S8B10B:THReshold

Arguments

B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a digital reference waveform as the source.

Examples

BUS:B1:S8B10B:SOUrce Ch5 sets the source to channel 5.

BUS:B1:S8B10B:SOUrce? might return :BUS:B1:S8B10B:SOUrce CH5, indicating the source is set to channel 5.

BUS:B<x>:S8B10B:THReshold

This command sets or queries the 8B10b threshold for the specified bus, where the bus is specified by x.

Conditions

Requires option SR8B10b.

Group

Bus

Syntax

```
BUS:B<x>:S8B10B:THReshold <NR3>
BUS:B<x>:S8B10B:THReshold?
```

Related Commands

BUS:B<x>:S8B10B:SOUrce

Arguments

B<x> is the number of the bus.

<NR3> is the 8B10b Strobe threshold for the specified bus in volts. The valid range is -8V to +8V.

Examples

BUS:B4:S8B10B:THReshold 1.0 sets Bus 4 8B10b threshold to 1.0 V.

BUS:B3:S8B10B:THReshold? might return :BUS:B3:S8B10B:THReshold 225.00000E-3 indicates the Bus 3 8B10b threshold is set to 225.0 mV.

BUS:B<x>:SDLC:BITRate

This command sets or queries the bit rate for the specified SDLC bus. The bus is specified by x.

Conditions Requires option SR-SDLC.

Group Bus

Syntax
```
BUS:B<x>:SDLC:BITRate <NR1>
BUS:B<x>:SDLC:BITRate?
```

Arguments B<x> is the number of the bus.

<NR1> specifies the bit rate. The default bit rate is 10 kbs and varies 300 ~ 1000000000.

Examples BUS:B1:SDLC:BITRate 80000 sets the bit rate as 80 Kbps.

BUS:B1:SDLC:BITRate? might return :BUS:B1:SDLC:BITRate 80000, indicating that the bit rate is 80 Kbps.

BUS:B<x>:SDLC:DATA:SOUrce

This command sets or queries the source for the specified bus. The bus is specified by x.

Conditions Requires option SR-SDLC.

Group Bus

Syntax
```
BUS:B<x>:SDLC:DATA:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SDLC:DATA:SOUrce?
```

Arguments B<x> is the number of the bus.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math waveform as the source.

REF<x> specifies a digital reference waveform as the source.

Examples BUS:B1:SDLC:DATA:SOUrce Ch5 sets the source to channel 5.

BUS:B1:SDLC:DATA:SOUrce? might return :BUS:B1:SDLC:DATA:SOUrce CH5, indicating the source is set to channel 5.

BUS:B<x>:SDLC:DATA:THReshold

This command sets or queries the SDLC data source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-SDLC.

Group Bus

Syntax

```
BUS:B<x>:SDLC:DATA:THReshold <NR3>
BUS:B<x>:SDLC:DATA:THReshold?
```

Arguments

`B<x>` is the number of the bus.

`<NR3>` is the SDLC Strobe threshold for the specified bus. The valid range is -8V to +8V.

Examples

`BUS:B1:SDLC:DATA:THReshold 1.0` sets the bus 1 SDLC threshold to 1.0 V.

`BUS:B1:SDLC:DATA:THReshold?` might return `:BUS:B1:SDLC:DATA:THReshold 225.00000E-3` indicates the bus 1 SDLC threshold is set to 225.0 mV.

BUS:B<x>:SDLC:ENCoding

This command sets or queries the SDLC Bus Encoding. The bus is specified by x.

Conditions Requires option SR-SDLC.

Group Bus

Syntax

```
BUS:B<x>:SDLC:ENCoding {DISCrete|INVert}
BUS:B<x>:SDLC:ENCoding?
```

Arguments

`B<x>` is the number of the bus waveform.

`DISCrete` specifies the encoding mechanism is Discrete Transmission (NRZ).

`INVert` specifies that encoding mechanism is Invert On Zero i.e. NRZI.

Examples

`BUS:B1:SDLC:ENCoding DISCrete` sets the encoding method to discrete transmission.

`BUS:B1:SDLC:ENCoding?` might return `:BUS:B1:SDLC:ENCoding DISCrete`, indicating that the encoding type is Discrete Transmission.

BUS:B<x>:SDLC:MODulo

This command sets or queries the SDLC Bus Modulo. The bus is specified by x.

Conditions

Requires option SR-SDLC.

Group

Bus

Syntax

```
BUS:B<x>:SDLC:MODulo {8|128}
BUS:B<x>:SDLC:MODulo?
```

Arguments

`B<x>` is the number of the bus waveform.

`8` specifies the C-Field size is 8 bit in SDLC frame.

`128` specifies the C-Field size is 16 bit in SDLC frame.

Examples

`BUS:B1:SDLC:MODulo 128` sets the modulo to 128.

`BUS:B1:SDLC:MODulo?` might return `:BUS:B1:SDLC:MODulo 128`, indicating that the modulo is 128.

BUS:B<x>:SENT:CHANWidth

This command sets or queries SENT fast channel bit widths for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

```
BUS:B<x>:SENT:CHANWidth
{TWELVEtwelve|FOURTEENten|SIXTEENeight}
BUS:B<x>:SENT:CHANWidth?
```

Related Commands

BUS:B<x>:SENT:NUMCHANnel

Arguments

B<x> is the number of the bus waveform.

TWELVEtwelve sets both Fast Channel 1 and Fast Channel 2 to12 bits wide.

FOURTEENten sets Fast Channel 1 to 14 bits and Fast Channel 2 to 10 bits.

SIXTEENeight sets Fast Channel 1 to 16 bits and Fast Channel 2 to eight bits.

Examples

BUS:B1:SENT:CHANWIDTH TWELVETWELVE sets the channel width of both Fast Channel 1 and Fast Channel 2 to twelve bits.

BUS:B1:SENT:CHANWIDTH? might return :BUS:B1:SENT:CHANWIDTH SIXTEENEIGHT, which indicates that the Fast Channel 1 width is 16 and the Fast Channel 2 width is eight.

BUS:B<x>:SENT:NIBBLECount

This command sets or queries SENT data nibbles for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group

Bus

Syntax

```
BUS:B<x>:SENT:NIBBLECount {THREE|FOUR|SIX}
BUS:B<x>:SENT:NIBBLECount?
```

Arguments

B<x> is the number of the bus waveform.

THREE specifies regular data with 3 nibbles

FOUR specifies regular data with 4nibbles

SIX specifies regular data with 6nibbles

Examples

BUS:B1:SENT:NIBBLECOUNT THREE sets the data nibble count to three.

BUS:B1:SENT:NIBBLECOUNT? might return :BUS:B1:SENT:NIBBLECOUNT SIX, which indicates the nibble count is set to six.

BUS:B<x>:SENT:NUMCHANnel

This command sets or queries SENT fast data channels for the specified bus.

Conditions

Requires option SRAUTOSEN.

Group Bus

Syntax
```
BUS:B<x>:SENT:NUMCHANnel {ONE|TWO}
BUS:B<x>:SENT:NUMCHANnel?
```

Arguments B<x> is the number of the bus waveform.

ONE specifies a SENT bus with one fast channel.

TWO specifies a SENT bus with two fast channels.

Examples BUS:B1:SENT:NUMCHANnel TWO sets the number of SENT channels to two.

BUS:B1:SENT:NUMCHANNEL? might return :BUS:B1:SENT:NUMCHANNEL ONE to indicate that the number of channels is set to one.

BUS:B<x>:SENT:PAUSEPULSe

This command sets or queries SENT pause pulse for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax
```
BUS:B<x>:SENT:PAUSEPULSe {NO|YES}
BUS:B<x>:SENT:PAUSEPULSe?
```

Arguments B<x> is the number of the bus waveform.

NO specifies no pause pulse.

YES specifies a pause pulse is used.

Examples BUS:B1:SENT:PAUSEPULSE NO sets the pause pulse value to indicate no pause pulse.

BUS:B1:SENT:PAUSEPULSE? might return :BUS:B1:SENT:PAUSEPULSE YES, which indicates the SENT bus is set to use pause pulses.

BUS:B<x>:SENT:POLARITY

This command sets or queries SENT Idle State signal polarity for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax
```
BUS:B<x>:SENT:POLARITY {INVerted|NORmal}
BUS:B<x>:SENT:POLARITY?
```

Arguments `B<x>` is the number of the bus waveform.

`INVerted` specifies inverted polarity.

`NORmal` specifies normal polarity.

Examples `BUS:B1:SENT:POLARITY INVERTED` sets the idle state of the bus to inverted polarity.

`BUS:B1:SENT:POLARITY?` might return `:BUS:B1:SENT:POLARITY NORMAL` to indicate the bus idle state polarity is set to normal

BUS:B<x>:SENT:SLOW

This command sets or queries the SENT slow channel configuration for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax
```
BUS:B<x>:SENT:SLOW {NOne|ENHANCED4|ENHANCED8|SHOrt}
BUS:B<x>:SENT:SLOW?
```

Arguments `B<x>` is the number of the bus waveform.

`NOne` specifies no slow channel configured.

`ENHANCED4` specifies Enhanced 4 slow channel configuration.

`ENHANCED8` specifies Enhanced 8 slow channel configuration.

`SHOrt` specifies short slow channel configuration.

Examples `BUS:B1:SENT:SLOW SHORT` sets the SENT bus to use the short slow channel configuration.

BUS:B1:SENT:SLOW? might return :BUS:B1:SENT:SLOW NONE to indicate that the SENT bus is set to have no slow channel.

BUS:B<x>:SENT:SOUrce

This command sets or queries the SENT DATA source for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax

```
BUS:B<x>:SENT:SOUrce
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:SENT:SOUrce?
```

Arguments B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the clock source waveform for the audio bus.

CH<x>_D<x> specifies a digital channel as the clock source waveform for the specified audio bus.

Math<x> specifies a math waveform as the clock source waveform for the audio bus.

REF<x> specifies a reference waveform as the clock source waveform for the audio bus

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified audio bus.

Examples BUS:B1:SENT:SOURCE CH1_D1 sets D1 of channel 1 as the source for the SENT bus.

BUS:B1:SENT:SOURCE? might return :BUS:B1:SENT:SOURCE CH1 indicating that the source is set to channel 1

BUS:B<x>:SENT:THRESHold

This command sets or queries the SENT DATA source threshold for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax
```
BUS:B<x>:SENT:THRESHold <NR3>
BUS:B<x>:SENT:THRESHold?
```

Arguments B<x> is the number of the bus waveform.

<NR3> sets the data source threshold value in volts.

Examples BUS:B1:SENT:THRESHOLD 500.0E-3 sets the Bus 1 source threshold to 500.0 mV.

BUS:B1:SENT:THRESHOLD? might return :BUS:B1:SENT:THRESHOLD 2.5000 to indicate the Bus 1 source threshold is 2.5 V.

BUS:B<x>:SENT:TICKTIME

This command sets or queries the SENT bus Clock Tick parameter for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax
```
BUS:B<x>:SENT:TICKTIME <NR3>
BUS:B<x>:SENT:TICKTIME?
```

Arguments B<x> is the number of the bus waveform.

<NR3> is the SENT clock tick time, in seconds.

Examples BUS:B1:SENT:TICKTIME 3.0E-6 sets the SENT bus to clock tick to 3 microseconds.

BUS:B1:SENT:TICKTIME? might return :BUS:B1:SENT:TICKTIME 10.0000E-6 to indicate the clock tick is set to 10 microseconds.

BUS:B<x>:SENT:TICKTOLerance

This command sets or queries the SENT bus Tick Tolerance percent parameter for the specified bus.

Conditions Requires option SRAUTOSEN.

Group Bus

Syntax
```
BUS:B<x>:SENT:TICKTOLerance <NR3>
BUS:B<x>:SENT:TICKTOLerance?
```

Arguments B<x> is the number of the bus waveform.

<NR3> is the tick tolerance percentage.

Examples `BUS:B1:SENT:TICKTOLERANCE 20.0` sets the tick tolerance to 20%.

`US:B1:SENT:TICKTOLERANCE?` might return `:BUS:B1:SENT:TICKTOLERANCE 22.5` to indicate the tolerance is set to 22.5%.

BUS:B<x>:SPACEWIRe:BITRate

This command sets or queries the SpaceWire bit rate.

Conditions Requires option SR-SPACEWIRE

Group Bus

Syntax
```
BUS:B<x>:SPACEWIRe:BITRate <NR3>
BUS:B<x>:SPACEWIRe:BITRate?
```

Arguments B<x> is the bus number.

NR3 specifies the SpaceWire bit rate for the specified bus. The valid bit rate range is 2 Mbps to 200 Mbps. The default value is 10 Mbps.

Examples `BUS:B1:SPACEWIRe:BITRate 10000000` sets the bit rate to 10 Mbps.

`BUS:B1:SPACEWIRe:BITRate?` might return `:BUS:B1:SPACEWIRe:BITRate RATE189K`, indicating that the bit rate is set to 10 Mbps.

BUS:B<x>:SPACEWIRe:DATa:SOUrce

This command sets or queries the source of the SpaceWire Data signal for the specified bus.

Conditions Requires option SR-SPACEWIRE

Group Bus

Syntax

```
BUS:B<x>:SPACEWIRe:DATa:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPACEWIRe:DATa:SOUrce?
```

Related Commands BUS:B<x>:SPACEWIRe:DATa:THReshold

Arguments `B<x>` is the bus number.

`CH<x>` specifies an analog channel as the source for the data signal, where <x> is the channel number.

`MATH<x>` specifies a math channel as the source for the data signal, where <x> is the math waveform number.

`REF<x>` specifies a reference waveform as the source for the data signal, where <x> is the reference waveform number.

Examples `BUS:B1:SPACEWIRe:DATa:SOUrce Ch5` sets the Bus 1 DATa source to channel 5.

`BUS:B3:SPACEWIRe:DATa:SOUrce?` might return `BUS:B3:SPACEWIRe:DATa:SOUrce MATH2`, indicating the DATa source for Bus 3 is set to Math waveform 2.

BUS:B<x>:SPACEWIRe:DATa:THReshold

This command sets or queries the threshold of the SpaceWire Data signal for the specified bus.

Conditions Requires option SR-SPACEWIRE

Group Bus

Syntax

```
BUS:B<x>:SPACEWIRe:DATa:THReshold <NR3>
BUS:B<x>:SPACEWIRe:DATa:THReshold?
```

Related Commands

BUS:B<x>:SPACEWIRe:DATa:SOUrce

Arguments

B<x> is the bus number.

NR3 specifies the SpaceWire Data threshold level for the specified bus, in volts.

Examples

BUS:B1:SPACEWIRe:DATa:THReshold 2.0 sets the Bus 1 DATa source threshold to 2.0 V.

BUS:B3:SPACEWIRe:DATa:THReshold? might return BUS:B3:SPACEWIRe:DATa:THReshold 978.00000E-3, indicating the DATa source threshold level for Bus 3 is set to 978.0 mV.

BUS:B<x>:SPACEWIRe:DECode:TYPe

This command sets or queries the decode type for SpaceWire bus decode.

Conditions

Requires option SR-SPACEWIRE

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:DECode:TYPe {STRObe|DATARate}
BUS:B<x>:SPACEWIRe:DECode:TYPe?
```

Arguments

B<x> is the bus number.

STRObe specifies the decode type as strobe.

DATARate specifies the decode type as data rate.

Examples

BUS:B1:SPACEWIRe:DECode:TYPe STRObe sets the decode type as strobe.

BUS:B1:SPACEWIRe:DECode:TYPe? might return :BUS:B1:SPACEWIRe:DECode:TYPe Strobe, indicating the decode type is strobe.

BUS:B<x>:SPACEWIRe:STRobe:SOUrce

This command sets or queries the source of the SpaceWire Strobe signal for the specified bus.

Conditions Requires option SR-SPACEWIRE

Group Bus

Syntax

```
BUS:B<x>:SPACEWIRe:STRobe:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPACEWIRe:STRobe:SOUrce?
```

Related Commands BUS:B<x>:SPACEWIRe:STRobe:THReshold

Arguments `B<x>` is the bus number.

`CH<x>` specifies an analog channel as the source for the strobe signal, where <x> is the channel number.

`MATH<x>` specifies a math channel as the source for the strobe signal, where <x> is the math waveform number.

`REF<x>` specifies a reference waveform as the source for the strobe signal, where <x> is the reference waveform number.

Examples `BUS:B1:SPACEWIRe:STRobe:SOUrce Ch5` sets the Bus 1 Strobe source to channel 5.

`BUS:B3:SPACEWIRe:STRobe:SOUrce?` might return `BUS:B3:SPACEWIRe:STRobe:SOUrce MATH2`, indicating the Strobe source for Bus 3 is set to Math waveform 2.

BUS:B<x>:SPACEWIRe:STRobe:THReshold

This command sets or queries the threshold level of the SpaceWire Strobe signal for the specified bus.

Conditions Requires option SR-SPACEWIRE

Group Bus

Syntax

```
BUS:B<x>:SPACEWIRe:STRobe:THReshold <NR3>
BUS:B<x>:SPACEWIRe:STRobe:THReshold?
```

Related Commands

BUS:B<x>:SPACEWIRe:STRobe:SOUrce

Arguments

B<x> is the bus number.

NR3 specifies the SpaceWire Strobe signal threshold level for the specified bus, in volts.

Examples

BUS:B4:SPACEWIRe:STRobe:THReshold 1.2 sets the Bus 4 Strobe source threshold to 1.2 V.

BUS:B1:SPACEWIRe:STRobe:THReshold? might return BUS:B1:SPACEWIRe:STRobe:THReshold 950.00000E-3, indicating the Strobe source threshold level for bus 1 is set to 950.0 mV.

BUS:B<x>:SPACEWIRe:SYNC

This command sets or queries sync for SpaceWire decoding.

Conditions

Requires option SR-SPACEWIRE

Group

Bus

Syntax

```
BUS:B<x>:SPACEWIRe:SYNC {DATA|NULL|AUTO|CUSTom}
BUS:B<x>:SPACEWIRe:SYNC?
```

Arguments

B<x> is the bus number.

DATA specifies sync as data.

NULL specifies sync as null.

AUTO specifies sync as auto.

CUSTom specifies sync as custom.

Examples

BUS:B1:SPACEWIRe:SYNC NULL sets the sync to null.

BUS:B1:SPACEWIRe:SYNC? might return :BUS:B1:SPACEWIRe:SYNC NULL, indicating the sync is set to null.

BUS:B<x>:SPACEWIRe:SYNC:COUnt

This command sets or queries the length of the data string in bytes to be used for a SpaceWire sync bytes.

Conditions Requires option SR-SPACEWIRE

Group Bus

Syntax

```
BUS:B<x>:SPACEWIRe:SYNC:COUnt <NR1>
BUS:B<x>:SPACEWIRe:SYNC:COUnt?
```

Arguments B<x> is the bus number.

NR1 specifies the length of the data string in bytes. The mininum is 2 and the maximum is 10. The default is 2 bytes.

Examples BUS:B1:SPACEWIRe:SYNC:COUnt 3 sets the count to 3.

BUS:B1:SPACEWIRe:SYNC:COUnt? might return :BUS:B1:SPACEWIRe:SYNC:COUnt 3, indicating the count is 3.

BUS:B<x>:SPACEWIRe:SYNC:PATTern

This command sets or queries sync pattern for SpaceWire decoding.

Conditions Requires option SR-SPACEWIRE

Group Bus

Syntax

```
BUS:B<x>:SPACEWIRe:SYNC:PATTern <NR3>
BUS:B<x>:SPACEWIRe:SYNC:PATTern?
```

Arguments B<x> is the bus number.

NR3 specifies the sync pattern.

Examples `BUS:B1:SPACEWIRe:SYNC:PATTern N N` sets the sync pattern to N N.

`BUS:B1:SPACEWIRe:SYNC:PATTern?` might return `:BUS:B1:SPACEWIRe:SYNC:PATTern N N`, indicating the Bus 1 SpaceWire pattern to be N N.

BUS:B<x>:SPACEWIRe:SYNC:VALUe

This command sets or queries sync value for sync option data.

Conditions Requires option SR-SPACEWIRE

Group Bus

Syntax
```
BUS:B<x>:SPACEWIRe:SYNC:VALUe <NR3>
BUS:B<x>:SPACEWIRe:SYNC:VALUe?
```

Arguments `B<x>` is the bus number.

`NR3` specifies the sync value for sync option data.

Examples `BUS:B1:SPACEWIRe:SYNC:VALUe D-00 D-00` sets the sync value to D-00 D-00.

`BUS:B1:SPACEWIRe:SYNC:VALUe?` might return `:BUS:B1:SPACEWIRe:SYNC:VALUe D-00 D-00`, indicating the Bus 1 SpaceWire sync value is D-00 D-00.

BUS:B<x>:SPI:BITOrder

This command sets or queries the SPI bit order for the specified bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax
```
BUS:B<x>:SPI:BITOrder {LSB|MSB}
BUS:B<x>:SPI:BITOrder?
```

Related Commands BUS:B<x>:SPI:DATa:SIZe

Arguments B<x> is the number of the bus waveform.

LSB specifies that each bit becomes the recovered value's new LSB, after shifting previously recovered bits one place to the left. The decoding happens right to left.

MSB specifies that each successive bit from the bus's data line becomes the new MSB of the recovered value, shifting any previously recovered bits one place to the right. The decoding happens left to right.

Examples BUS:B1:SPI:BITORDER LSB sets each bit order for the SPI Bus 1 to LSB.

BUS:B1:SPI:BITORDER? might return BUS:B1:SPI:BITORDER MSB, indicating that the bit order for the SPI Bus 1 is set to MSB.

BUS:B<x>:SPI:CLOCk:POLarity

This command sets or queries the SPI clock (SCLK) source polarity for the specified bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax
```
BUS:B<x>:SPI:CLOCk:POLarity {FALL|RISE}
BUS:B<x>:SPI:CLOCk:POLarity?
```

Related Commands BUS:B<x>:SPI:CLOCk:SOUrce

Arguments B<x> is the number of the bus waveform.

FALL sets the clock to the falling edge of the signal.

RISE sets the clock to the rising edge of the signal.

Examples BUS:B1:SPI:CLOCK:POLARITY FALL sets the SPI clock polarity to the falling edge for the Bus 1.

BUS:B1:SPI:CLOCK:POLARITY? might return BUS:B1:SPI:CLOCK:POLARITY RISE, indicating that the SPI clock polarity for the Bus 1 is set to the rising edge.

BUS:B<x>:SPI:CLOCk:SOUrce

This command sets or queries the SPI clock (SCLK) source for the specified bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax

```
BUS:B<x>:SPI:CLOCk:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:SPI:CLOCk:SOUrce?
```

Related Commands BUS:B<x>:SPI:CLOCk:POLarity

Arguments

`B<x>` is the number of the bus waveform.

`CH<x>` designates an analog channel as the bus SPI clock source.

`CH<x>_D<x>` designates an digital channel as the bus SPI clock source.

`MATH<x>` designates a math waveform as the clock source.

`REF<x>` designates a reference waveform as the clock source.

`REF<x>_D<x>` specifies a digital reference waveform as the clock source waveform for the specified SPI bus.

Examples

`BUS:B1:SPI:CLOCK:SOURCE CH5` sets the SPI clock source for the Bus 1 to CH5.

`BUS:B1:SPI:CLOCK:SOURCE?` might return `BUS:B1:SPI:CLOCK:SOURCE MATH1`, indicating that the SPI clock source for the Bus 1 is set to MATH1.

BUS:B<x>:SPI:CLOCk:THReshold

This command sets or queries the SPI Clock (SCLK) source threshold for the specified bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax

```
BUS:B<x>:SPI:CLOCk:THReshold <NR3>
BUS:B<x>:SPI:CLOCk:THReshold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the SPI Clock (SCLK) source threshold for the specified bus.

Examples

BUS:B1:SPI:CLOCk:THReshold 50.0e-3 sets the threshold to 50.0 mV.

BUS:B1:SPI:CLOCk:THReshold? might return BUS:B1:SPI:CLOCK:THRESHOLD 0.0E+0, indicating the threshold is set to 0.0 V.

BUS:B<x>:SPI:DATa:POLarity

This command sets or queries the SPI Data (Data) source polarity for the bus number specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:DATa:POLarity {HIGH|LOW}
BUS:B<x>:SPI:DATa:POLarity?
```

Arguments

B<x> is the number of the bus waveform.

HIGH sets the SPI data polarity to active high.

LOW sets the SPI data polarity to active low.

Examples

BUS:B4:SPI:DATA:POLARITY HIGH sets the SPI data polarity for the Bus 4 to active high.

BUS:B1:SPI:DATA:POLARITY? might return BUS:B1:SPI:DATA:POLARITY LOW, indicating that the SPI SDA polarity for the Bus 1 is set to active low.

BUS:B<x>:SPI:DATa:SIZe

This command sets or queries the number of bits per word for the specified SPI bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax
```
BUS:B<x>:SPI:DATa:SIZe <NR1>
BUS:B<x>:SPI:DATa:SIZe?
```

Arguments B<x> is the number of the bus waveform.

<NR1> is the data size for the specified bus. The minimum value is 2 and maximum is 32.

Examples BUS:B1:SPI:DATA:SIZE 7 sets the data size for SPI Bus 1 to seven bits per word.

BUS:B1:SPI:DATA:SIZE? might return BUS:B1:SPI:DATA:SIZE 8, indicating that the data size for SPI Bus 1 is set to eight bits per word.

BUS:B<x>:SPI:DATa:SOUrce

This command sets or queries the SPI Data (Data) source for the bus number specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax
```
BUS:B<x>:SPI:DATa:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:SPI:DATa:SOUrce?
```

Arguments B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the data source for the specified SPI bus.

CH<x>_D<x> designates an digital channel as the bus SPI clock source.

MATH<x> designates a math waveform as the data source.

REF<x> designates a reference waveform as the data source.

REF<x>_D<x> specifies a digital reference waveform as the data source waveform for the specified SPI bus.

Examples `BUS:B1:SPI:DATA:SOURCE CH2` sets the SPI data source for the Bus 1 to CH2.

`BUS:B3:SPI:DATA:SOURCE?` might return `BUS:B3:SPI:DATA:SOURCE CH3`, indicating that the SPI data source for the Bus 3 is set to CH3.

BUS:B<x>:SPI:DATa:THReshold

This command sets or queries the SPI Data (Data) source threshold for the specified bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax
```
BUS:B<x>:SPI:DATa:THReshold <NR3>
BUS:B<x>:SPI:DATa:THReshold?
```

Arguments `B<x>` is the number of the bus waveform.

`<NR3>` is the SPI Data (SDA) source threshold for the specified bus.

Examples `BUS:B1:SPI:DATa:THReshold 50.0e-3` sets the threshold to 50.0 mV.

`BUS:B1:SPI:DATa:THReshold?` might return `BUS:B1:SPI:DATA:THRESHOLD 0.0E+0` indicating the threshold is set to 0.0 V.

BUS:B<x>:SPI:FRAMING

This command sets or queries the SPI framing setting for the specified bus. The bus number is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax
```
BUS:B<x>:SPI:FRAMING {IDLE|SS}
BUS:B<x>:SPI:FRAMING?
```

Arguments `B<x>` is the number of the bus waveform.

`IDLE` specifies IDLE SPI framing.

`SS` specifies SS SPI framing.

Examples `BUS:B1:SPI:FRAMING IDLE` sets the SPI framing to IDLE.

`BUS:B1:SPI:FRAMING?` might return `BUS:B1:SPI:FRAMING SS`, indicating that the SPI framing is set to SS.

BUS:B<x>:SPI:IDLETime

This command sets or queries the SPI idle time for the specified bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax

```
BUS:B<x>:SPI:IDLETime <NR3>
BUS:B<x>:SPI:IDLETime?
```

Arguments `B<x>` is the number of the bus waveform.

`<NR3>` specifies the SPI idle time.

Examples `BUS:B1:SPI:IDLETime 0.000004` sets the idle time to 4 µs.

`BUS:B1:SPI:IDLETime?` might return `BUS:B1:SPI:IDLETIME 5.0000E-6`, indicating that the idle time is set to 5 µs.

BUS:B<x>:SPI:MISo:DATa:POLarity

This command sets or queries the SPI MISo Data source polarity for the specified bus. The bus is specified by x.

Conditions Requires option SR-SPI.

Group Bus

Syntax

```
BUS:B<x>:SPI:MISo:DATa:POLarity {HIGH|LOW}
BUS:B<x>:SPI:MISo:DATa:POLarity?
```

Arguments

B<x> is the number of the bus waveform.

HIGH sets the polarity to high.

LOW sets the polarity to low.

Examples

BUS:B1:SPI:MISo:DATa:POLarity HIGH sets the SPI MISo data polarity to high.

BUS:B1:SPI:MISo:DATa:POLarity? might return BUS:B1:SPI:MISo:DATa:POLarity HIGH, indicating that the SPI MISo data polarity is high.

BUS:B<x>:SPI:MISo:INPut

This command sets or queries the SPI MISo source for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SPI.

Group

Bus

Syntax

```
BUS:B<x>:SPI:MISo:INPut {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPI:MISo:INPut?
```

Arguments

B<x> is the number of the bus waveform.

CH<x> designates an analog channel as the source.

MATH<x> designates a math waveform as the source.

REF<x> designates a reference waveform as the source.

Examples

BUS:B1:SPI:MISo:INPut Ch5 sets the source to channel 5.

BUS:B1:SPI:MISo:INPut? might return BUS:B1:SPI:MISo:INPut HIGH, indicating the source is set to channel 5.

BUS:B<x>:SPI:MISo:THReshold

This command sets or queries the SPI MISo Data source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-SPI.

Group Bus

Syntax

```
BUS:B<x>:SPI:MISo:THReshold <NR3>
BUS:B<x>:SPI:MISo:THReshold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the SPI MISo Data source threshold for the specified bus.

Examples

`BUS:B1:SPI:MISo:THReshold 1.0` sets the Bus 1 SPI MISo threshold to 1.0 V.

`BUS:B1:SPI:MISo:THReshold?` might return `:BUS:B1:SPI:MISo:THReshold 225.00000E-3`, indicates the Bus 1 SPI MISo threshold is set to 225.0 mV.

BUS:B<x>:SPI:MOSi:DATa:POLarity

This command sets or queries the SPI MOSI source polarity for the specified bus. The bus is specified by x.

Conditions Requires option SR-SPI.

Group Bus

Syntax

```
BUS:B<x>:SPI:MOSi:DATa:POLarity {HIGH|LOW}
BUS:B<x>:SPI:MOSi:DATa:POLarity?
```

Arguments

`B<x>` is the number of the bus waveform.

`HIGH` sets the polarity to high.

`LOW` sets the polarity to low.

Examples `BUS:B1:SPI:MOSi:DATa:POLarity HIGH` sets the SPI MOSI data polarity to high.

`BUS:B1:SPI:MOSi:DATa:POLarity?` might return `:BUS:B1:SPI:MOSi:DATa:POLarity HIGH`, indicating that the SPI MOSI data polarity is high.

BUS:B<x>:SPI:MOSi:INPut

This command sets or queries the SPI MOSI source for the specified bus. The bus is specified by x.

Conditions Requires option SR-SPI.

Group Bus

Syntax
```
BUS:B<x>:SPI:MOSi:INPut {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SPI:MOSi:INPut?
```

Arguments `B<x>` is the number of the bus waveform.

`CH<x>` designates an analog channel as the source.

`MATH<x>` designates a math waveform as the source.

`REF<x>` designates a reference waveform as the source.

Examples `BUS:B1:SPI:MOSi:INPut Ch5` sets the source to channel 5.

`BUS:B1:SPI:MOSi:INPut?` might return `:BUS:B1:SPI:MOSi:INPut HIGH`, indicating the source is set to channel 5.

BUS:B<x>:SPI:MOSi:THReshold

This command sets or queries the SPI MOSI source thresold for the specified bus. The bus is specified by x.

Conditions Requires option SR-SPI.

Group Bus

Syntax

```
BUS:B<x>:SPI:MOSi:THReshold <NR3>
BUS:B<x>:SPI:MOSi:THReshold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the SPI MOSI Data source threshold for the specified bus.

Examples

BUS:B1:SPI:MOSi:THReshold 1.0 sets the Bus 1 SPI MOSI threshold to 1.0 V.

BUS:B1:SPI:MOSi:THReshold? might return :BUS:B1:SPI:MOSi:THReshold 225.00000E-3, indicates the Bus 1 SPI MOSI threshold is set to 225.0 mV.

BUS:B<x>:SPI:NUMBer:INputs

This command sets or queries the number of inputs for the specified bus. The bus is specified by x.

Conditions

Requires option SR-SPI.

Group

Bus

Syntax

```
BUS:B<x>:SPI:NUMBer:INputs {ONE|TWO}
BUS:B<x>:SPI:NUMBer:INputs?
```

Arguments

B<x> is the number of the bus waveform.

ONE sets the Data Inputs as one.

TWO sets the Data Inputs as two.

Examples

BUS:B1:SPI:NUMBer:INputs TWO sets the Data Inputs as two.

BUS:B1:SPI:NUMBer:INputs? might return :BUS:B1:SPI:NUMBer:INputs TWO, indicating that the Data Inputs is two.

BUS:B<x>:SPI:SELect:POLarity

This command sets or queries the SPI Slave Select (SS) polarity for the specified bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax
```
BUS:B<x>:SPI:SELect:POLarity {LOW|HIGH}
BUS:B<x>:SPI:SELect:POLarity?
```

Related Commands BUS:B<x>:SPI:SELect:SOUrce

Arguments `B<x>` is the number of the bus waveform.

`LOW` sets an active low polarity.

`HIGH` sets an active high polarity.

Examples `BUS:B1:SPI:SELECT:POLARITY HIGH` sets the SPI Slave Select polarity for the bus B1 to active high.

`BUS:B1:SPI:SELECT:POLARITY?` might return `BUS:B1:SPI:SELECT:POLARITY HIGH`, indicating that the SPI Slave Select polarity for the bus B1 is set to active high.

BUS:B<x>:SPI:SELect:SOUrce

This command sets or queries the SPI Slave Select (SS) source for the specified bus. The bus is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD.

Group Bus

Syntax
```
BUS:B<x>:SPI:SELect:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:SPI:SELect:SOUrce?
```

Related Commands BUS:B<x>:SPI:SELect:POLarity

Arguments `B<x>` is the number of the bus waveform.

`CH<x>` designates an analog channel as the buses' SPI Slave Select source.

CH<x>_D<x> designates a digital channel as the buses' SPI Slave Select source.

MATH<x> designates a math waveform as the Slave Select source.

REF<x> designates a reference waveform as the Slave Select source.

REF<x> _D<x> specifies a digital reference waveform as the clock source waveform for the specified SPI bus.

Examples

BUS:B1:SPI:SELECT:SOURCE MATH3 sets the SPI Slave Select source for the bus B1 to MATH3.

BUS:B1:SPI:SELECT:SOURCE? might return BUS:B1:SPI:SELECT:SOURCE CH3, indicating that the SPI Slave Select source for the bus B1 is set to CH3.

BUS:B<x>:SPI:SELect:THReshold

This command sets or queries the SPI Select (SS) source threshold for the specified bus. The bus is specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD.

Group

Bus

Syntax

```
BUS:B<x>:SPI:SELect:THReshold <NR3>
BUS:B<x>:SPI:SELect:THReshold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the SPI Select (SS) source threshold for the specified bus.

Examples

BUS:B1:SPI:SELect:THReshold 50.0e-3 sets the threshold to 50.0 mV.

BUS:B1:SPI:SELect:THReshold? might return BUS:B1:SPI:SELECT:THRESHOLD 0.0E+0, indicating the threshold is set to 0.0 V.

BUS:B<x>:SPMI:SCLk:SOUrce

This command sets or queries the SPMI Clock (SCLK) source for the specified bus.

Conditions

Requires option SRPM

Group Bus

Syntax
```
BUS:B<x>:SPMI:SCLk:SOUrce {CH<x>|CH<x>_Dx>|
Math<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:SPMI:SCLk:SOUrce
```

Related Commands BUS:B<x>:SPMI:SDATa:SOUrce

Arguments `B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the clock source waveform for the SPMI bus.

`CH<x>_Dx>` specifies a digital channel and bit as the clock source waveform for the specified SPMI bus.

`Math<x>` specifies a math waveform as the clock source waveform for the specified SPMI bus.

`REF<x>` specifies a reference waveform as the clock source waveform for the specified SPMI bus.

`REF<x>_D<x>` specifies a digital reference waveform and bit as the clock source waveform for the specified SPMI bus.

Examples `BUS:B1:SPMI:SCLk:SOUrce CH1_D1` sets D1 of channel 1 as the clock source for the SPMI bus B1.

`BUS:B1:SPMI:SCLk:SOUrce?` might return `BUS:B3:SPMI:SCLk:SOUrce CH3`, indicating that the data source for bus B3 is set to channel 3.

BUS:B<x>:SPMI:SCLk:THReshold

This command sets or queries the SPMI Clock (SCLK) source threshold for the specified bus.

Conditions Requires option SRPM

Group Bus

Syntax
```
BUS:B<x>:SPMI:SCLk:THReshold <NR3>
BUS:B<x>:SPMI:SCLk:THReshold?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

Arguments

B<x> is the number of the bus waveform.

<NR3> is the clock (SCLK) source threshold value for the specified SPMI bus.

Examples

BUS:B3:SPMI:SCLK:THRESHOLD 500.0E-3 sets the Bus3 clock source threshold to 500.0 mV.

BUS:B1:SPMI:SCLK:THRESHOLD? might return BUS:B1:SPMI:SCLK:THRESHOLD 250.0E-3 to indicate Bus1 clock threshold is set to 250.0 mV.

BUS:B<x>:SPMI:SDATa:SOUrce

This command sets or queries the SPMI Data (SDATA) source for the specified bus.

Conditions

Requires option SRPM

Group

Bus

Syntax

```
BUS:B<x>:SPMI:SDATa:SOUrce {CH<x>|CH<x>_Dx>|
Math<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:SPMI:SDATa:SOUrce
```

Related Commands

BUS:B<x>:SPMI:SCLk:SOUrce

Arguments

B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the data source waveform for the SPMI bus.

CH<x>_Dx> specifies a digital channel and bit as the data source waveform for the specified SPMI bus.

Math<x> specifies a math waveform as the data source waveform for the specified SPMI bus.

REF<x> specifies a reference waveform as the data source waveform for the specified SPMI bus.

REF<x>_D<x> specifies a digital reference waveform and bit as the data source waveform for the specified SPMI bus.

Examples

BUS:B4:SPMI:SDATa:SOUrce CH1_D2 sets bit D2 of digital channel 1 as the data source for the SPMI bus B4.

BUS:B1:SPMI:SCLk:SOUrce? might return BUS:B3:SPMI:SCLk:SOUrce CH5 indicating that the data source is set to channel 5 for bus B3.

BUS:B<x>:SPMI:SDATa:THReshold

This command sets or queries the SPMI Data (SDATA) source threshold for the specified bus.

Conditions Requires option SRPM

Group Bus

Syntax

```
BUS:B<x>:SPMI:SDATa:THReshold <NR3>
BUS:B<x>:SPMI:SDATa:THReshold?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

Arguments B<x> is the number of the bus waveform.

<NR3> is the data (SDATA) source threshold value for the specified SPMI bus.

Examples BUS:B2:SPMI:SDATA:THRESHOLD 400.0E-3 sets Bus2 data source threshold to 400.0 mV.

BUS:B1:SPMI:SDATA:THRESHOLD? might return BUS:B1:SPMI:SDATA:THRESHOLD 220.0E-3 to indicate that Bus1 data source threshold is set to 220.0 mV

BUS:B<x>:SVID:ALERT:SOUrce

This command sets or queries alert source channel for the specified SVID bus. The bus is specified by x.

Conditions Requires option SR-SVID

Group Bus

Syntax

```
BUS:B<x>:SVID:ALERT:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SVID:ALERT:SOUrce?
```

Arguments B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the data source.

MATH<x> specifies a math channel as the data source.

REF<x> specifies a digital reference waveform as the data source.

Examples BUS:B1:SVID:ALERT:SOUrce CH5 sets the source to channel 5.

BUS:B1:SVID:ALERT:SOUrce? might return :BUS:B1:SVID:ALERT:SOUrce CH5, indicating the source is set to channel 5.

BUS:B<x>:SVID:ALERT:THReshold

This command sets or queries the alert threshold for the specified SVID bus. The bus is specified by x.

Conditions Requires option SR-SVID

Group Bus

Syntax
```
BUS:B<x>:SVID:ALERT:THReshold <NR3>
BUS:B<x>:SVID:ALERT:THReshold?
```

Arguments B<x> is the number of the bus waveform.

<NR3> is the SVID Strobe threshold for the specified bus. The argument range is -8V to +8V

Examples BUS:B1:SVID:ALERT:THReshold 0.6 sets the Bus 1 SVID threshold to 600 mV.

BUS:B1:SVID:ALERT:THReshold? might return :BUS:B1:SVID:ALERT:THReshold 600.00000E-3, indicates the Bus 1 SVID threshold is set to 600 mV.

BUS:B<x>:SVID:CLOCk:SOUrce

This command sets or queries clock source channel for the specified SVID bus. The bus is specified by x.

Conditions Requires option SR-SVID

Group Bus

Syntax
```
BUS:B<x>:SVID:CLOCk:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SVID:CLOCk:SOUrce?
```

Arguments B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math channel as the source.

REF<x> specifies a digital reference waveform as the source.

Examples BUS:B1:SVID:CLOCk:SOUrce CH3 sets the source to channel 3.

BUS:B1:SVID:CLOCk:SOUrce? might return :BUS:B1:SVID:CLOCk:SOUrce CH3, indicating the source is set to channel 3.

BUS:B<x>:SVID:CLOCk:THReshold

This command sets or queries the clock threshold for the specified SVID bus. The bus is specified by x.

Conditions Requires option SR-SVID

Group Bus

Syntax
```
BUS:B<x>:SVID:CLOCk:THReshold <NR3>
BUS:B<x>:SVID:CLOCk:THReshold?
```

Arguments B<x> is the number of the bus waveform.

<NR3> is the SVID Strobe threshold for the specified bus. The argument range is -8V to +8V

Examples BUS:B1:SVID:CLOCk:THReshold 1.0 sets the Bus 1 SVID threshold to 1.0 V.

BUS:B1:SVID:CLOCk:THReshold? might return :BUS:B1:SVID:CLOCk:THReshold 1.0000, indicates the Bus 1 SVID threshold is set to 1V.

BUS:B<x>:SVID:DATA:SOUrce

This command sets or queries data source channel for the specified SVID bus. The bus is specified by x.

Conditions Requires option SR-SVID

Group Bus

Syntax

```
BUS:B<x>:SVID:DATA:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:SVID:DATA:SOUrce?
```

Arguments

B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the source.

MATH<x> specifies a math channel as the source.

REF<x> specifies a digital reference waveform as the source.

Examples

BUS:B1:SVID:DATA:SOUrce CH4 sets the source to channel 4.

BUS:B1:SVID:DATA:SOUrce? might return :BUS:B1:SVID:DATA:SOUrce CH4, indicating the source is set to channel 4.

BUS:B<x>:SVID:DATA:THReshold

This command sets or queries the data threshold for the specified SVID bus. The bus is specified by x.

Conditions Requires option SR-SVID

Group Bus

Syntax

```
BUS:B<x>:SVID:DATA:THReshold <NR3>
BUS:B<x>:SVID:DATA:THReshold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the SVID Strobe threshold for the specified bus. The argument range is -8V to +8V

Examples

`BUS:B1:SVID:DATA:THReshold 0.25` sets the Bus 1 SVID threshold to 250 mV.

`BUS:B1:SVID:DATA:THReshold?` might return `:BUS:B1:SVID:DATA:THReshold 250.00000E-3`, indicates the Bus 1 SVID threshold is set to 250 mV.

BUS:B<x>:TYPe

This command sets or queries the bus type or standard for the specified bus. The bus is specified by x. Arguments for a bus type are only available then the required serial bus option is installed.

Group

Bus

Syntax

```
BUS:B<x>:TYPe {ARINC429| AUDio| CAN| ETHernet| EUSB|
FLEXRAY| I2C| I3C| LIN| MDIO| MIL1553B| PARallel| RS232C|
SENT| SPI| SPMI| SVID| USB}
BUS:B<x>:TYPe?
```

Arguments

`B<x>` is the number of the bus waveform.

`ARINC429` specifies the ARINC 429 avionics serial bus.

`AUDio` specifies an audio bus.

`CAN` specifies a Controller Area Network bus.

`ETHernet` specifies the Ethernet bus.

`EUSB` specifies a eUSB bus. Requires option SR-EUSB2.

`FLEXRAY` specifies a FlexRay bus.

`I2C` specifies the Inter-IC bus.

`I3C` specifies the MIPI Improved Inter Integrated Circuit (I3C) bus.

`LIN` specifies a Local Interconnect Network bus.

`MDIO` specifies a MDIO bus.

`MIL1553B` specifies the MIL-STD-1553 avionics serial bus.

`PARallel` specifies a parallel bus.

`RS232C` specifies the RS-232 Serial bus.

`SENT` specifies the Single Edge Nibble Transmission (SENT) automotive serial bus.

SPI specifies the Serial Peripheral Interface bus.

SPMI Specifies a System Power Management Interface bus

SVID Specifies a Serial VID bus

USB specifies the Universal Serial bus.

Examples

BUS:B1:TYPE I2C sets the bus B1 type to I2C.

BUS:B1:TYPE? might return BUS:B1:TYPE SPI, indicating that the bus B1 type is set to SPI and that the bus behavior is governed by the BUS:B1:SPI:XXXX commands.

BUS:B<x>:USB:BITRate

This command sets or queries the USB data rate for bus <x>, where the bus number is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:BITRate {FULL|HIGH|LOW}
BUS:B<x>:USB:BITRate?
```

Arguments

B<x> is the number of the bus waveform.

FULL indicates the bit rate is 12 Mbps.

HIGH indicates the bit rate is 480 Mbps.

LOW indicates the bit rate is 1.5 Mbps.

Examples

BUS:B1:USB:BITRATE FULL sets the bit rate to 12 Mbps.

BUS:B1:USB:BITRATE? might return BUS:B1:USB:BITRATE LOW, indicating that the bit rate is 1.5 Mbps.

BUS:B<x>:USB:DATAMINUSTHRESHold

This command sets or queries the USB D- source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-USB2.

Group Bus

Syntax
```
BUS:B<x>:USB:DATAMINUSTHRESHold <NR3>
BUS:B<x>:USB:DATAMINUSTHRESHold?
```

Arguments B<x> is the number of the bus waveform.

<NR3> is the Minus threshold.

Examples `BUS:B1:USB:DATAMINUSTHRESHold 50.0e-3` sets the threshold to 50.0 mV.

`BUS:B1:USB:DATAMINUSTHRESHold?` might return `BUS:B1:USB:DATAMINUSTHRESHOLD 0.0E+0` indicating the threshold is set to 0.0 V.

BUS:B<x>:USB:DATAPLUSTHRESHold

This command sets or queries the USB D+ source threshold for the specified bus. The bus is specified by x.

Conditions Requires option SR-USB2.

Group Bus

Syntax
```
BUS:B<x>:USB:DATAPLUSTHRESHold <NR3>
BUS:B<x>:USB:DATAPLUSTHRESHold?
```

Arguments B<x> is the number of the bus waveform.

<NR3> is the Plus threshold.

Examples `BUS:B1:USB:DATAPLUSTHRESHold 50.0e-3` sets the threshold to 50.0 mV.

`BUS:B1:USB:DATAPLUSTHRESHold?` might return `BUS:B1:USB:DATAPLUSTHRESHOLD 0.0E+0` indicating the threshold is set to 0.0 V.

BUS:B<x>:USB:LOWTHRESHold

This command sets or queries the USB data source threshold for the specified bus when the signal type is differential. The bus is specified by x.

Conditions Requires option SR-USB2.

Group Bus

Syntax

```
BUS:B<x>:USB:LOWTHRESHold <NR3>
BUS:B<x>:USB:LOWTHRESHold?
```

Arguments

`B<x>` is the number of the bus waveform.

`<NR3>` is the Low threshold.

Examples

`BUS:B1:USB:LOWTHRESHold 50.0e-3` sets the threshold to 50.0 mV.

`BUS:B1:USB:LOWTHRESHold?` might return `BUS:B1:USB:LOWTHRESHOLD 0.0E+0` indicating the threshold is set to 0.0 V.

BUS:B<x>:USB:SIGNALTYpe

This command sets or queries the USB signal type for the specified bus. The bus is specified by x.

Conditions Requires option SR-USB2.

Group Bus

Syntax

```
BUS:B<x>:USB:SIGNALTYpe {SINGLE|DIFF}
BUS:B<x>:USB:SIGNALTYpe?
```

Arguments

`B<x>` is the number of the bus waveform.

`SINGLE` specifies single-ended signals.

`DIFF` specifies differential signals.

Examples `BUS:B1:USB:SIGNALTYpe SINGLE` specifies single-ended signals.

`BUS:B1:USB:SIGNALTYpe?` might return `BUS:B1:USB:SIGNALTYPE DIFF` indicating the signal type is differential.

BUS:B<x>:USB:SOUrce

This command sets or queries the USB data source when the signal type is differential for bus <x>. The bus number is specified by x.

Conditions Requires option SR-USB2.

Group Bus

Syntax
```
BUS:B<x>:USB:SOUrce {CH<x>|MATH<x>|REF<x>}
BUS:B<x>:USB:SOUrce?
```

Related Commands BUS:B<x>:USB:SOUrce:DMINus

BUS:B<x>:USB:SOUrce:DPLUs

Arguments `B<x>` is the number of the bus waveform.

`CH<x>` specifies an analog channel as the data source for the specified USB bus.

`MATH<x>` specifies a math channel as the data source for the specified USB bus.

`REF<x>` specifies a reference waveform as the data source.

Examples `BUS:B1:USB:SOURCE D2` sets the USB data source for bus B1 to D2.

`BUS:B1:USB:SOURCE?` might return `BUS:B1:USB:SOURCE D3`, indicating that the USB data source for bus B1 is D3.

BUS:B<x>:USB:SOUrce:DMINus

This command sets or queries the USB D- (SDATAMINUS) source for bus <x> when the signal type is single ended. The bus number is specified by x.

Conditions Requires option SR-USB2.

Group Bus

Syntax

```
BUS:B<x>:USB:SOUrce:DMINus
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:USB:SOUrce:DMINus?
```

Related Commands

BUS:B<x>:USB:SOUrce

BUS:B<x>:USB:SOUrce:DPLUs

BUS:B<x>:USB:SIGNALTYpe

Arguments

B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the D- source for the specified USB bus.

CH<x>_D<x> specifies a digital channel as the D- source for the specified USB bus.

MATH<x> specifies a math channel as the D- source for the specified USB bus.

REF<x> specifies a reference waveform as the source

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified USB bus.

Examples

`BUS:B1:USB:SOURCE:DMINUS CH2` sets the USB Data Source for D- input to CH2.

`BUS:B1:USB:SOURCE:DMINUS?` might return `BUS:B1:USB:SOURCE:DMINUS CH4`, indicating that CH4 is set to be the D- input for USB data.

BUS:B<x>:USB:SOUrce:DPLUs

This command sets or queries the USB dataPlus (SDATAPLUS) source for the specified bus when the signal type is single ended. The bus is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:SOUrce:DPLUs
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
BUS:B<x>:USB:SOUrce:DPLUs?
```

Related Commands

BUS:B<x>:USB:SOUrce

BUS:B<x>:USB:SOUrce:DMINus

BUS:B<x>:USB:SIGNALTYpe

Arguments

B<x> is the number of the bus waveform.

CH<x> specifies an analog channel as the D+ source for the specified USB bus

CH<x>_D<x> specifies a digital channel as the D+ source for the specified USB bus

MATH<x> specifies a math channel as the D+ source for the specified USB bus.

REF<x> specifies a reference waveform as the source.

REF<x>_D<x> specifies a digital reference waveform as the clock source waveform for the specified USB bus.

Examples

BUS:B1:USB:SOURCE:DPLUS CH2 sets the USB Data Source for D+ input to CH2.

BUS:B1:USB:SOURCE:DPLUS? might return BUS:B1:USB:SOURCE:DPLUS CH3, indicating that CH3 is set to be the D+ input for USB data.

BUS:B<x>:USB:THRESHold

This command sets or queries the USB DATA source High threshold for the specified bus when the signal source is differential. The bus is specified by x.

Conditions

Requires option SR-USB2.

Group

Bus

Syntax

```
BUS:B<x>:USB:THRESHold <NR3>
BUS:B<x>:USB:THRESHold?
```

Arguments

B<x> is the number of the bus waveform.

<NR3> is the USB DATA source High threshold for the specified bus.

Examples

BUS:B1:USB:THRESHold 50.0e-3 sets the threshold to 50.0 mV.

BUS:B1:USB:THRESHold? might return BUS:B1:USB:THRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

BUS:DELete (No Query Form)

This command deletes the specified bus.

Group Bus

Syntax `BUS:DELete <QString>`

Arguments `<QString>` specifies the bus to delete and is of the form "B<NR1>", where <NR1> is ≥1.

Examples `BUS:DELETE B1` deletes bus B1.

BUS:LIST? (Query Only)

This query returns a comma separated list of all currently defined buses.

Group Bus

Syntax `BUS:LIST?`

Returns Returns all currently defined buses.

Examples `BUS:LIST?` might return `BUS:LIST B1,B4` indicating the bus 1 and bus 4 are defined.

BUSTABle:ADDNew (No Query Form)

Adds the specified bus table. Argument is of the form `"TABLE<NR1>"`, where `<NR1>` is ≥1).

Group Bus

Syntax `BUSTABle:ADDNew <QString>`

Arguments `<QString>` is a quoted string that is the name of the new bus table.

Examples `BUSTABLE:ADDNEW "Table1"` adds bus table Table1.

BUSTABle:DELete (No Query Form)

Deletes the specified bus table. Argument is of the form `"TABLE<NR1>"`, where `<NR1>` is ≥1).

Group Bus

Syntax `BUSTABle:DELete <QString>`

Arguments `<QString>` is a quoted string that is the name of the bus table to delete.

Examples `BUSTABLE:DELETE "Table1"` deletes bus table Table1.

BUSTABle:LIST? (Query Only)

This query lists all currently defined bus tables.

Group Bus

Syntax `BUSTABle:LIST?`

Returns Returns a list of all currently defined bus tables.

Examples `BUSTABLE:LIST?` might return `BUSTABLE:LIST TABLE1` indicating TABLE1 is currently the only defined bus table.

BUSY? (Query Only)

This query-only command returns the status of the instrument. This command allows you to synchronize the operation of the instrument with your application program.

Group Status and Error

Syntax BUSY?

Related Commands *OPC

*WAI

Returns <NR1> = 0 means that the instrument is not busy processing a command whose execution time is extensive.

<NR1> = 1 means that the instrument is busy processing Commands that Generate an OPC Message (See Table 2-51.).

Examples BUSY? might return BUSY 1, indicating that the instrument is currently busy.

*CAL? (Query Only)

This query-only command starts signal path calibration (SPC) and returns the status upon completion.

NOTE. *When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 40 minutes on an 8-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.*

Group Calibration

Syntax *CAL?

Returns 0 indicates SPC passed.

-1 indicates SPC failed or did not complete.

Examples *CAL? starts the signal path calibration and returns the status upon completion.

CALibrate? (Query Only)

This query returns the status of signal path calibration.

NOTE. When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes, which can take several minutes.

Group Calibration

Syntax `CALibrate?`

Examples `CALIBRATE?` might return `CALIBRATE:INTERNAL:STATUS PASS`, indicating the calibration status.

CALibrate:INTERNal (No Query Form)

This command (no query form) starts the signal path calibration (SPC) of the instrument. You can use the CALibrate:INTERNal:STATus? query to return the current status of the signal path calibration of the instrument.

NOTE. When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 40 minutes on an 8-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.

Group Calibration

Syntax `CALibrate:INTERNal`

Related Commands CALibrate:INTERNal:STATus?

Examples `CALIBRATE:INTERNAL` starts the signal path calibration of the instrument.

CALibrate:INTERNal:STARt (No Query Form)

This command (no query form) starts the signal path calibration (SPC) of the analog channels. This command is the same as the CALibrate:INTERNal command. You can use the CALibrate:INTERNal:STATus? query to return the current status of the signal path calibration of the instrument.

NOTE. *When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 40 minutes on an 8-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.*

Group Calibration

Syntax `CALibrate:INTERNal:STARt`

Related Commands CALibrate:INTERNal:STATus?

Examples `CALIBRATE:INTERNAL:START` starts the signal path calibration.

CALibrate:INTERNal:STATus? (Query Only)

This query-only command returns the current status of the signal path calibration.

NOTE. *When running SPC through the remote interface, calibration status cannot be obtained until after the SPC completes. SPC takes approximately 5 minutes per channel which means a total of 40 minutes on an 8-channel model. Any remote command that performs an action on the instrument is also disabled until the SPC is complete.*

Group Calibration

Syntax `CALibrate:INTERNal:STATus?`

Related Commands *CAL?

Returns This query will return one of the following:

- `INIT` indicates the instrument has not had signal path calibration run. The instrument may need to be readjusted at the Tektronix service center.
- `PASS` indicates that the signal path calibration completed successfully.
- `FAIL` indicates that the signal path calibration did not complete successfully.

Examples `CALIBRATE:INTERNAL:STATUS?` might return `PASS`, indicating that the current status of the signal path calibration is that the signal path calibration completed successfully.

CALibrate:PWRUpstatus? (Query Only)

This query-only command returns the current status of the power-up calibration.

Group Calibration

Syntax `CALibrate:PWRUpstatus?`

Returns This query will return one of the following:

- 0 Indicating the power-up calibration failed.
- 1 Indicating the power-up calibration passed.

Examples `CALIBRATE:PWRUPSTATUS?` might return `0`, indicating that the power-up calibration failed.

CALLOUTS:CALLOUT<x> (No Query Form)

This command creates a new note. A Note is the default callout value. The callout number is specified by x.

Group Callouts

Syntax `CALLOUTS:CALLOUT<x>`

Examples `CALLOUTS:CALLOUT1` creates a new Note callout specified as callout1.

CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE

This command sets or queries the source of the Bookmark callout type.

Group Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE {CH1|CH2|CH3|CH4}
CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE?
```

Arguments

CH1 specifies the bookmark callout source as Ch1.

CH2 specifies the bookmark callout source as Ch2.

CH3 specifies the bookmark callout source as Ch3.

CH4 specifies the bookmark callout source as Ch4.

Examples

CALLOUTS:CALLOUT1:BOOKMark:SOURCE CH2 sets the bookmark callout source as Ch2 for callout1.

CALLOUTS:CALLOUT1:BOOKMark:SOURCE?, might return "CH1", indicating that the bookmark callout source is Ch1.

CALLOUTS:CALLOUT<x>:BOOKMark:XPOS

This command sets or queries the X-Position of the Bookmark callout type.

Group Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:BOOKMark:XPOS <NR1>
CALLOUTS:CALLOUT<x>:BOOKMark:XPOS?
```

Arguments

<NR1> specifies the location of the bookmark linked to the source waveform in X-axis.

Examples

CALLOUTS:CALLOUT1:BOOKMark:XPOS 9.28µs sets the bookmark callout X-Position to 9.28 µs for callout1.

CALLOUTS:CALLOUT1:BOOKMark:XPOS?, might return "9.28 µs", indicating the bookmark callout X-Position is 9.28 µs.

CALLOUTS:CALLOUT<x>:COLOR

This command sets or queries the text color of the callout.

Group Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:COLOR <QString>
CALLOUTS:CALLOUT<x>:COLOR?
```

Arguments

<QString> specifies the callout text color using hexadecimal color values.

Examples

CALLOUTS:CALLOUT1:COLOR "#FF0000" sets the text color of callout 1 to red.

CALLOUTS:CALLOUT1:COLOR?, might return "#FF0000", indicating the text color is red.

CALLOUTS:CALLOUT<x>:FONT:BOLD

This command sets or queries the bold state of the callout text.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:FONT:BOLD {1|0}
CALLOUTS:CALLOUT<x>:FONT:BOLD?
```

Arguments

1 specifies the callout font weight as bold.

0 specifies the callout font weight as normal.

Examples

CALLOUTS:CALLOUT1:FONT:BOLD 1 sets the font weight of callout1 to bold.

CALLOUTS:CALLOUT1:FONT:BOLD?, might return "0", indicating the font weight is set to normal.

CALLOUTS:CALLOUT<x>:FONT:ITALIC

This command sets or queries the italic state of the callout text.

Group

Callouts

Syntax

```
CALLOUTS:CALLOUT<x>:FONT:ITALIC {1|0}
CALLOUTS:CALLOUT<x>:FONT:ITALIC?
```

Arguments

1 specifies the callout font style as italic.

0 does not specify the font style as italic.

Examples `CALLOUTS:CALLOUT1:FONT:ITALIC 1` sets the font style of callout1 to italic.

`CALLOUTS:CALLOUT1:FONT:ITALIC?`, might return `"0"`, indicating the font style is not italic.

CALLOUTS:CALLOUT<x>:FONT:SIZE

This command sets or queries the font size of the callout text.

Group Callouts

Syntax
```
CALLOUTS:CALLOUT<x>:FONT:SIZE <NR1>
CALLOUTS:CALLOUT<x>:FONT:SIZE?
```

Arguments `<NR1>` specifies the font size in points.

Examples `CALLOUTS:CALLOUT1:FONT:SIZE 11` sets the font size of callout1 to 11 points.

`CALLOUTS:CALLOUT1:FONT:SIZE?`, might return `"14"`, indicating the font size is 14 points.

CALLOUTS:CALLOUT<x>:FONT:TYPE

This command sets or queries type of the callout.

Group Callouts

Syntax
```
CALLOUTS:CALLOUT<x>:FONT:TYPE <QString>
CALLOUTS:CALLOUT<x>:FONT:TYPE?
```

Arguments `<QString>` specifies the type of font for the callout text. The available font types include: DejaVu Sans, DejaVu Sans Mono, DejaVu Serif, Frutiger LT Std, Monospace, Sans Serif, Serif, Ubuntu, Ubuntu Condensed, and Ubuntu Mono.

Examples `CALLOUTS:CALLOUT1:FONT:TYPE "Serif"` sets the font type of callout1 to serif.

`CALLOUTS:CALLOUT1:FONT:TYPE?`, might return `"Monospace"`, indicating the font type is monospace.

CALLOUTS:CALLOUT<x>:FONT:UNDERLine

This command sets or queries the underline state of the callout text.

Group Callouts

Syntax
```
CALLOUTS:CALLOUT<x>:FONT:UNDERLine {1|0}
CALLOUTS:CALLOUT<x>:FONT:UNDERLine?
```

Arguments 1 underlines the callout text.

0 does not underline the callout text.

Examples CALLOUTS:CALLOUT1:FONT:UNDERLine 1 displays the text of callout 1 as underlined.

CALLOUTS:CALLOUT1:FONT:UNDERLine?, might return "0", indicating the callout text is not underlined.

CALLOUTS:CALLOUT<x>:TEXT

This command sets or queries the callout text.

Group Callouts

Syntax
```
CALLOUTS:CALLOUT<x>:TEXT <QString>
CALLOUTS:CALLOUT<x>:TEXT?
```

Arguments <QString> specifies the callout text.

Examples CALLOUTS:CALLOUT1:TEXT "Review measurements" sets the text for callout 1 as "Review measurements".

CALLOUTS:CALLOUT1:TEXT?, might return "Review measurements", indicating the text for callout 1 is "Review measurements".

CALLOUTS:CALLOUT<x>:TYPE

This command sets or queries type of the callout.

Group Callouts

Syntax
```
CALLOUTS:CALLOUT<x>:TYPE {NOTE|ARROW|RECTANGLE|BOOKMARK}
CALLOUTS:CALLOUT<x>:TYPE?
```

Arguments NOTE specifies callout type as note.

ARROW specifies callout type as arrow.

RECTANGLE specifies callout type as rectangle.

BOOKMARK specifies callout type as bookmark.

Examples CALLOUTS:CALLOUT1:TYPE NOTE sets the callout type of callout1 to note.

CALLOUTS:CALLOUT1:TYPE?, might return "ARROW", indicating the callout type is arrow.

CH<x>? (Query Only)

This query-only command returns the vertical parameters for the specified channel. The channel is specified by x.

Group Vertical

Syntax CH<x>?

Examples CH1? might return the following vertical parameters for channel 1:
```
CH1:BANDWIDTH 1.0000E+09;COUPLING DC;DESKEW0.0000E+00;OFFSET
0.0000E+00;POSITION 0.0000E+00;SCALE 5.0000E-01;PROBCAL
INIT;PROBE:GAIN 1.0000E+00;RESISTANCE 1.0000E+06;UNITS
"V";ID:TYPE "1X";SERNUMBER "N/A";:CH1:PROBEFUNC:EXTATTEN
1.0000E+00;EXTUNITS "None";:CH1:LABEL:NAME "";XPOS 5;YPOS 5.
```

CH<x>:BANdwidth

This command sets or queries the selectable low-pass bandwidth limit filter of the specified channel. The channel is specified by x.

The query form of this command always returns the approximate realized bandwidth of the channel.

Available arguments depend upon the instrument and the attached accessories.

Group Vertical

Syntax
```
CH<x>:BANdwidth {<NR3>|FULl}
CH<x>:BANdwidth?
```

Arguments CH<x> is the channel number.

<NR3> is the desired bandwidth. The instrument rounds this value to an available bandwidth using geometric rounding and then uses this value to set the upper bandwidth.

FULl disables any optional bandwidth limiting. The specified channel operates at its maximum bandwidth.

Examples CH1:BANDWIDTH 20 sets the bandwidth of Channel 1 to 20 MHz.

CH2:BANDWIDTH?, might return :CH2:BANDWIDTH 500.0000E+06, indicating that there is bandwidth limiting on Channel 2.

CH<x>:BANdwidth:FILTer:OPTIMIZation

This command sets or queries the channel filter shape. The channel is specified by x.

Conditions Only available on 6 Series MSO instruments.

Group Vertical

Syntax
```
CH<x>:BANdwidth:FILTer:OPTIMIZation {STEPRESPONSE|FLATNESS}
CH<x>:BANdwidth:FILTer:OPTIMIZation?
```

Arguments CH<x> is the channel number.

STEPRESPONSE sets a Bessel-Thompson filter that minimizes overshoot with a gradual rollof.

FLATNESS sets selects a brick-wall filter optimized for flatness within band with a sharp rolloff. Flatness filtering is not compatible with Peak Detect and Envelope acquisition modes..

Examples CH1:BANdwidth:FILTer:OPTIMIZation FLATNESS sets channel 1 filter shape to Flatness.

CH1:BANdwidth:FILTer:OPTIMIZation?, might return :CH1:BANdwidth:FILTer:OPTIMIZation FLATNESS, indicating that the filter shape is Flatness for channel 2.

CH<x>:CLIPping? (Query Only)

Queries whether the specified channel's input signal is clipping (exceeding) the channel A/D converter range. The channel is specified by x.

Group Vertical

Syntax CH<x>:CLIPping?

Related Commands CH<x>:SCAle

CH<x>:POSition

Returns This query will return one of the following:

0 indicates the channel is not clipping.

1 indicates the channel is clipping.

Examples CH4:CLIPping? might return CH4:CLIPping 1, indicating that the input signal range is exceeding (clipping) the channel 4 A/D converter range.

CH<x>:COUPling

This command sets or queries the input coupling setting for the specified analog channel. The channel is specified by x.

NOTE. *The available arguments depend on the attached accessories.*

Group Vertical

Syntax
```
CH<x>:COUPling {AC|DC|DCREJ}
CH<x>:COUPling?
```

Arguments CH<x> is the channel number.

AC sets the specified channel to AC coupling.

DC sets the specified channel to DC coupling.

DCREJect sets DC Reject coupling when probes are attached that support that feature.

Examples CH2:COUPLING AC sets Channel 2 coupling to AC.

CH3:COUPLING? might return :CH3:COUPLING DC, indicating that Channel 3 is set to DC coupling.

CH<x>:DESKew

This command sets or queries the horizontal deskew time for the specified channel. The channel is specified by x.

Group Vertical

Syntax
```
CH<x>:DESKew <NR3>
CH<x>:DESKew?
```

Arguments CH<x> is the channel number.

<NR3> is the deskew time for this channel, ranging from -125 ns to +125 ns with a resolution of 40 ps. Out-of-range values are clipped.

Examples CH4:DESKEW 5.0E-9 sets the deskew time for Channel 4 to 5 ns.

CH2:DESKEW? might return CH2:DESKEW 2.0000E-09, indicating that the deskew time for Channel 2 is set to 2 ns.

CH<x>:LABel:COLor

This command sets or queries the color of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:COLor <QString>

Arguments CH<x> is the channel number.

<QString> is the label color. To return the color to the default color, send an empty string as in this example: CH5:LABEL:COLOR "".

Examples CH2:LABel:COLor "#FFFF00" sets the label color to yellow.

CH2:LABel:COLor? might return CH2:LABEL:COLOR "#FF0000" indicating the color is red.

CH<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:FONT:BOLD {ON|OFF|<NR1>}

Arguments CH<x> is the channel number.

OFF argument turns off bold font.

ON argument turns on bold font.

<NR1> = 0 turns off bold font; any other value turns on bold font.

Examples CH2:LABel:FONT:BOLD OFF turns off the bold font.

CH2:LABel:FONT:BOLD? might return CH2:LABEL:FONT:BOLD 1 indicating a bold font.

CH<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:FONT:ITALic {ON|OFF|<NR1>}

Arguments CH<x> is the channel number.

OFF argument turns off italic font.

ON argument turns on italic font.

<NR1> = 0 turns off italic font; any other value turns on italic font.

Examples CH2:LABel:FONT:ITALic ON set the font to italic.

CH2:LABel:FONT:ITALic? might return CH2:LABEL:FONT:ITALIC 0 indicating the font is not italic.

CH<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:FONT:SIZE <NR1>

Arguments CH<x> is the channel number.

<NR1> is the font size.

Examples CH2:LABel:FONT:SIZE 16 sets the font size to 16 point size.

CH2:LABel:FONT:SIZE? might return CH2:LABEL:FONT:SIZE 20 indicating a 20 point font size.

CH<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified channel label, such as Arial or Times New Roman. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:FONT:TYPE <QString>

Arguments CH<x> is the channel number.

<QString> is the specified font type.

Examples CH2:LABel:FONT:TYPE "Monospace" sets the font to a mono space font.

CH2:LABel:FONT:TYPE? might return CH2:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman".

CH<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified channel label. The channel is specified by x.

Group Vertical

Syntax CH<x>:LABel:FONT:UNDERline {ON|OFF|<NR1>}

Arguments CH<x> is the channel number.

OFF argument turns off underlined font.

ON argument turns on underlined font.

<NR1> = 0 turns off underlined font; any other value turns on underlined font.

Examples CH2:LABel:FONT:UNDERline ON sets the font to underlined.

CH2:LABel:FONT:UNDERline? might return CH2:LABEL:FONT:UNDERLINE 0 indicating the font is not underlined.

CH<x>:LABel:NAMe

This command sets or queries the label attached to the displayed waveform for the specified channel. The channel is specified by x.

Group Vertical

Syntax
```
CH<x>:LABel:NAMe <QString>
CH<x>:LABel:NAMe?
```

Arguments CH<x> is the channel number.

<QString> is an alphanumeric character string, ranging from 1 through 32 characters in length.

Examples CH2:LABEL:NAME "Pressure" changes the waveform label for the Channel 2 waveform to “Pressure”.

CH3:LABEL:NAME? might return CH3:LABEL:NAME "Force", indicating that the waveform label for the Channel 3 waveform is “Force”.

CH<x>:LABel:XPOS

This command sets or queries the X-position of the specified channel label. The channel is specified by x.

Group Vertical

Syntax
```
CH<x>:LABel:XPOS <NR3>
CH<x>:LABel:XPOS?
```

Arguments CH<x> is the channel number.

<NR3> is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the left edge of the screen.

Examples CH3:LABEL:XPOS 5 moves the waveform label for Channel 3 so that it begins 5 pixels to the right of the left edge of the screen.

CH2:LABEL:XPOS? might return CH2:LABEL:XPOS 5, indicating that the waveform label for the Channel 2 currently 5 pixels to the right of the left edge of the screen.

CH<x>:LABel:YPOS

This command sets or queries the Y-position of the specified channel label. The channel is specified by x.

Group Vertical

Syntax
```
CH<x>:LABel:YPOS <NR3>
CH<x>:LABel:YPOS?
```

Arguments CH<x> is the channel number.

<NR3> is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the baseline of the waveform. Positive values are above the baseline and negative values are below.

Examples CH3:LABel:YPOS -20 moves the waveform label for the Channel 3 20 pixels below the baseline of the waveform.

CH2:LABel:YPOS? might return CH2:LABEL:YPOS 0, indicating that the waveform label for the Channel 2 is currently located at the baseline of the waveform.

CH<x>:OFFSet

This command sets or queries the vertical offset for the specified analog channel.

Group Vertical

Syntax
```
CH<x>:OFFSet <NR3>
CH<x>:OFFSet?
```

Arguments CH<x> is the channel number.

<NR3> is the offset value for the specified channel.

Examples CH3:OFFSET 2.0E-3 sets the offset for Channel 3 to 2 mV.

CH4:OFFSET? might return :CH4:OFFSET 1.0000E-03, indicating that the offset for Channel 4 is set to 1 mV.

CH<x>:POSition

This command sets or queries the vertical position for the specified analog channel.

Group Vertical

Syntax CH<x>:POSition <NR1>

Arguments CH<x> is the channel number.

<NR1> is the vertical position for the specified analog channel.

Examples CH2:POSition -2.0 sets the position to -2 divisions.

CH2:POSition? might return CH2:POSITION -2.2400 indicating the position is -2.24 divisions.

CH<x>:PRObe? (Query Only)

This query-only command returns all information concerning the probe that is attached to the specified channel. The channel is specified by x.

Group Vertical

Syntax CH<x>:PRObe?

Related Commands CH<x>:PROBECal?

Examples CH2:PROBE? might return CH2:PROBE:GAIN 1.0000E-01; RESISTANCE 1.0000E+07;UNITS "V";ID:TYPE "10X"'SERNUMBER "N/A" for a 10X probe, indicating that (among other parameters) the attenuation factor for the probe attached to Channel 2 is 100.0 mV (assuming that probe units are set to volts).

CH<x>:PRObe:AUTOZero (No Query Form)

This command executes the attached probe's Auto Zero function, for probes that support this feature. See your probe documentation for more details. The channel is specified by x.

Group Vertical

Syntax CH<x>:PRObe:AUTOZero EXECute

Arguments CH<x> is the channel number.

EXECute sets the probe attached to the specified channel to autozero.

Examples `CH1:PROBE:AUTOZERO EXECUTE` sets the probe attached to the Channel 1 to autozero.

CH<x>:PRObe:COMPensate (No Query Form)

This command starts the probe compensation procedure for passive probes. The channel is specified by x.

Group Vertical

Syntax `CH<x>:PRObe:COMPensate`

Examples `CH1:PRObe:COMPensate` starts probe compensation for passive probes (TPP).

CH<x>:PRObe:DEGAUSS (No Query Form)

This command starts a degauss cycle of the TekVPI probe attached to the specified channel. The channel is specified by x.

Group Vertical

Syntax `CH<x>:PRObe:DEGAUSS EXECute`

Arguments `CH<x>` is the channel number.

`EXECute` starts the degauss cycle.

Examples `CH1:PROBE:DEGAUSS EXECUTE` causes the probe attached to Channel 1 to degauss.

CH<x>:PRObe:DEGAUSS:STATE? (Query Only)

This command queries whether the probe attached to the specified channel requires a degauss operation. The channel is specified by x.

Group Vertical

Syntax `CH<x>:PRObe:DEGAUSS:STATE?`

Returns

`Required` indicates the probe should be degaussed before taking measurements.

`Recommended` indicates the measurement accuracy might be improved by degaussing the probe.

`Passed` indicates the probe is degaussed.

`Failed` indicates the degauss operation failed.

`Inprocess` indicates the probe degauss operation is currently in progress.

Examples

`CH2:PROBE:DEGAUSS:STATE?` might return `CH2:PROBE:DEGAUSS:STATE PASSED`, indicating that the probe attached to the Channel 2 is degaussed.

CH<x>:PRObe:FORCEDRange

This command sets the attached TekVPI probe to the specified range, or it queries the range of the probe attached to the specified channel. The channel is specified by x.

Group

Vertical

Syntax

```
CH<x>:PRObe:FORCEDRange <NR3>
CH<x>:PRObe:FORCEDRange?
```

Arguments

`CH<x>` is the channel number.

`<NR3>` specifies the probe dynamic range.

Examples

If a TCP0030 current probe is attached to the Channel 1 input, `CH1:PROBE:FORCEDRANGE 5.0` sets the attached probe to its 5 Ampere range.

`CH3:PROBE:FORCEDRANGE?` might return `CH1:PROBE:FORCEDRANGE 30.0000`, indicating that the range of the probe attached to the Channel 3 is set to 30 Amperes.

CH<x>:PRObe:GAIN? (Query Only)

This query-only command returns the gain factor of the probe that is attached to the specified channel. The channel is specified by x. The gain of a probe is the output divided by the input transfer ratio. For example, a common 10x probe has a gain of 0.1.

Group

Vertical

Syntax CH<x>:PRObe:GAIN?

Examples CH2:PROBE:GAIN? might return CH2:PROBE:GAIN 100.0000E-3, indicating that the attached 10X probe delivers 0.1 V to the Channel 2 BNC for every 1.0 V applied to the probe input.

CH<x>:PRObe:ID? (Query Only)

This query-only command returns the type and serial number of the probe that is attached to the specified channel. The channel is specified by x.

Group Vertical

Syntax CH<x>:PRObe:ID?

Examples CH2:PROBE:ID? might return CH2:PROBE:ID:TYPE "10X";SERNUMBER "N/A", indicating that a passive 10X probe of unknown serial number is attached to Channel 2.

CH<x>:PRObe:ID:SERnumber? (Query Only)

This query-only command returns the serial number of the probe that is attached to the specified channel. The channel is specified by x.

NOTE. *For Level 0 and 1 probes, the serial number will be "N/A".*

Group Vertical

Syntax CH<x>:PRObe:ID:SERnumber?

Examples CH1:PROBE:ID:SERNUMBER? might return CH1:PROBE:ID:SERNUMBER "B010289", indicating that the serial number of the probe attached to Channel 1 is B010289.

CH<x>:PRObe:ID:TYPe? (Query Only)

This query-only command returns the type of probe that is attached to the specified channel. The channel is specified by x.

Group Vertical

Syntax `CH<x>:PRObe:ID:TYPe?`

Examples `CH1:PROBE:ID:TYPE?` might return `CH1:PROBE:ID:TYPE "TCP0030"`, indicating that a TCP0030 current probe is attached to Channel 1.

CH<x>:PRObe:INPUTMode

Sets or queries the input mode of the probe that is attached to the specified channel.

Conditions Requires a probe with dual inputs.

Group Vertical

Syntax
```
CH<x>:PRObe:INPUTMode {A|B|C|D}
CH<x>:PRObe:INPUTMode?
```

Arguments `CH<x>` is the channel number.

`A` sets the probe to send single-ended A signals to the instrument.

`B` sets the probe to send single-ended B signals to the instrument.

`C` sets the probe to send common-mode signals to the instrument.

`D` sets the probe to send differential signals to the instrument.

Examples `CH8:PROBE:INPUTMODE C` sets the probe input type to send common mode signals to channel 8.

`CH6:PROBE:INPUTMODE?` might return CH6:PROBE:INPUTMODE A, indicating that the input mode of the probe that is attached to the Channel 6 is set to single ended A signals.

CH<x>:PRObe:INPUTMode:AOFFSet

Sets or queries the A mode offset control of the probe that is attached to the specified channel.

Conditions Requires a probe with dual inputs.

Group Vertical

Syntax
```
CH<x>:PRObe:INPUTMode:AOFFSet <NR3>
CH<x>:PRObe:INPUTMode:AOFFSet?
```

Arguments CH<x> is the channel number.

<NR3> sets the A mode offset value, in vertical units (V or A).

Examples CH2:PROBE:INPUTMODE:AOFFSET 5.0 sets the A mode offset control value of the probe that is attached to the Channel 2 to 5.

CH1:PROBE:INPUTMODE:AOFFSET? might return CH1:PROBE:INPUTMODE:AOFFSET? 0.0000, indicating that the A mode offset value of the probe that is attached to the channel 1 is set to 0.0.

CH<x>:PRObe:INPUTMode:BOFFSet

Sets or queries the B mode offset control of the probe that is attached to the specified channel.

Conditions Requires a probe with dual inputs.

Group Vertical

Syntax
```
CH<x>:PRObe:INPUTMode:BOFFSet <NR3>
CH<x>:PRObe:INPUTMode:BOFFSet?
```

Arguments CH<x> is the channel number.

<NR3> sets the B mode offset value, in vertical units (V or A).

Examples CH2:PROBE:INPUTMODE:BOFFSET 5.0 sets the B mode offset value of the probe that is attached to the Channel 2 to 5.

CH5:PROBE:INPUTMODE:BOFFSET? might return CH5:PROBE:INPUTMODE:BOFFSET? 1.2500, indicating that the B mode offset value of the probe that is attached to the channel 5 is set to 1.25.

CH<x>:PRObe:INPUTMode:COFFSet

Sets or queries the common mode offset value of the probe that is attached to the specified channel.

Conditions Requires a probe with dual inputs.

Group Vertical

Syntax

```
CH<x>:PRObe:INPUTMode:COFFSet <NR3>
CH<x>:PRObe:INPUTMode:COFFSet?
```

Arguments CH<x> is the channel number.

<NR3> sets the C (common) mode offset value, in vertical units (V or A).

Examples CH7:PROBE:INPUTMODE:COFFSET 3.2 sets the C mode offset value of the probe that is attached to the Channel 7 to 3.2.

CH5:PROBE:INPUTMODE:COFFSET? might return CH5:PROBE:INPUTMODE:COFFSET? 1.0250, indicating that the C mode offset value of the probe that is attached to the channel 5 is set to 1.025.

CH<x>:PRObe:INPUTMode:DOFFSet

Sets or queries the differential offset value of the probe that is attached to the specified channel.

Conditions Requires a probe with dual inputs.

Group Vertical

Syntax

```
CH<x>:PRObe:INPUTMode:DOFFSet <NR3>
CH<x>:PRObe:INPUTMode:DOFFSet?
```

Arguments CH<x> is the channel number.

<NR3> sets the D (differential) mode offset value, in vertical units (V or A).

Examples `CH7:PROBE:INPUTMODE:DOFFSET 3.2` sets the D mode offset value of the probe that is attached to the Channel 7 to 3.2.

`CH5:PROBE:INPUTMODE:DOFFSET?` might return `CH5:PROBE:INPUTMODE:DOFFSET? 1.0250`, indicating that the D mode offset value of the probe that is attached to the channel 5 is set to 1.025.

CH<x>:PRObe:RESistance? (Query Only)

This query-only command returns the resistance of the probe that is attached to the specified channel. The channel is specified by x.

Group Vertical

Syntax `CH<x>:PRObe:RESistance?`

Examples `CH2:PROBE:RESISTANCE?` might return `:CH2:PROBE:RESISTANCE 1.0000E+06`, indicating that the input resistance of the probe attached to Channel 2 is 1 MΩ.

CH<x>:PRObe:SET

This command sets or queries aspects of probe accessory user interfaces, for example probe attenuation factors or probe audible over range. The available arguments for this command will vary depending on the accessory you attach to the instrument. The channel is specified by x.

Group Vertical

Syntax
```
CH<x>:PRObe:SET <QString>
CH<x>:PRObe:SET?
```

Arguments `CH<x>` is the channel number.

`<QString>` is a quoted string representing a settable aspect of the attached accessory.

Examples `CH6:PRObe:SET "ATTENUATION 10X"` sets the probe to 10X attenuation.

`CH6:PRObe:SET?` might return `CH6:PROBE:SET "ATTENUATION 25X"` indicating that the probe is set to the selection for 25X attenuation.

CH<x>:PRObe:STATus? (Query Only)

Queries the probe unsigned integer error value.

Conditions Requires a probe that supports the relevant error messages.

Group Vertical

Syntax `CH<x>:PRObe:STATus?x`

Returns Returns an integer number that represents the sum total of binary error bits B0 – B15. The error bits are not displayed; they are concatenated into the integer value. The following is a list of the error for each bit.

- B0 – Probe disabled
- B1 – Jaws open
- B2 – Over range
- B3 – Probe temperature out of limits
- B4 – Degauss needed
- B5 – Probe tip missing
- B6 – Probe tip failed
- B7 – Probe tip not supported
- B8 through B15 – Reserved

Examples `CH4:PROBE:STATus?` might return `CH4:PROBE:STATus 0`, indicating that the probe is not reporting any errors.

`CH4:PROBE:STATus?` might return `CH4:PROBE:STATus 20`, indicating that the probe is reporting an over range error and a Degauss needed error.

`CH4:PROBE:STATus?` might return `CH4:PROBE:STATus 2`, indicating that the probe is reporting an open jaws error.

CH<x>:PRObe:UNIts? (Query Only)

This query-only command returns a string describing the units of measure for the probe attached to the specified channel. The channel is specified by x.

Group Vertical

Syntax `CH<x>:PROBe:UNIts?`

Related Commands CH<x>:PROBEFunc:EXTUnits

Examples `CH4:PROBE:UNITS?` might return `:CH4:PROBE:UNITS "V"`, indicating that the units of measure for the probe attached to Channel 4 are volts.

CH<x>:PROBECal? (Query Only)

This query-only command returns the probe calibration state for the specified channel. The channel is specified by x.

Group Vertical

Syntax `CH<x>:PROBECal?`

Returns This query will return one of the following:

- `Failed` signifies that the probe calibration has failed for the selected channel.
- `Default` signifies that the probe calibration has not yet been run for the selected channel.
- `Passed` signifies the probe calibration has passed for the selected channel.
- `Running` signifies the probe calibration is running.

Examples `CH2:PROBECAL?` might return `CH2:PROBECAL PASSED` indicating that the probe calibration has passed for Channel 2.

CH<x>:PROBECOntrol

This command sets or queries multirange probe range-control policy preference of the probe that is attached to CH<x>. The channel number is specified by x.

Group Vertical

Syntax
```
CH<x>:PROBECOntrol {AUTO|MANual}
CH<x>:PROBECOntrol?
```

Arguments

CH<x> is the channel number.

AUTO sets the values. The probe range is automatically calculated.

MANual allows you to select various valid values for the probe connected to a particular channel.

Examples

CH2:PROBECONTROL AUTO sets the values and the probe range is automatically calculated.

CH2:PROBECONTROL? might return CH2:PROBECONTROL MANUAL indicating that you can select various valid values for the probe connected to channel 2.

CH<x>:PROBEFunc:EXTAtten

This command is used to specify the attenuation value as a multiplier to the given scale factor on the specified channel. The channel is specified by x.

The query form of this command returns the user-specified attenuation.

Group

Vertical

Syntax

```
CH<x>:PROBEFunc:EXTAtten <NR3>
CH<x>:PROBEFunc:EXTAtten?
```

Related Commands

CH<x>:PROBEFunc:EXTDBatten

Arguments

CH<x> is the channel number.

<NR3> is the attenuation value, which is specified as a multiplier in the range from 1.00E-10 to 1.00E+10.

Examples

CH1:PROBEFUNC:EXTATTEN 167.00E-3 specifies an external attenuation, which is connected between the your input signal and the input of the probe attached to Channel 1.

CH2:PROBEFUNC:EXTATTEN? might return CH2:PROBEFUNC:EXTATTEN 1.0000E+00, indicating that the probe attached to Channel 2 is connected directly to the user's signal.

CH<x>:PROBEFunc:EXTDBatten

This command sets or queries the input-output ratio (expressed in decibel units) of external attenuation or gain between the signal and the instrument input channels. The channel is specified by x.

The query form of this command returns the user-specified attenuation in decibels.

Group Vertical

Syntax

```
CH<x>:PROBEFunc:EXTDBatten <NR3>
CH<x>:PROBEFunc:EXTDBatten?
```

Related Commands CH<x>:PROBEFunc:EXTAtten

Arguments CH<x> is the channel number.

<NR3> is the attenuation value, which is specified in the range from -200.00 dB to 200.00 dB.

Examples CH3:PROBEFUNC:EXTDBATTEN 2.5 specifies an external 2.5 dB attenuator on Channel 3.

CH1:PROBEFUNC:EXTDBATTEN? might return CH1:PROBEFUNC:EXTDBATTEEN 2.5000E+00, indicating that the attenuation for Channel 1 is 2.5 dB.

CH<x>:PROBEFunc:EXTUnits

This command sets the unit of measurement for the external attenuator of the specified channel. The channel is specified by x. The alternate units are used if they are enabled. Use the CH<x>:PROBEFunc:EXTUnits:STATE command to enable or disable the alternate units.

Group Vertical

Syntax

```
CH<x>:PROBEFunc:EXTUnits <QString>
CH<x>:PROBEFunc:EXTUnits?
```

Related Commands CH<x>:PRObe:UNIts?

Arguments CH<x> is the channel number.

<QString> indicates the attenuation unit of measurement for the specified channel.

Examples CH4:PROBEFUNC:EXTUNITS "Pascals" sets the unit of measurement for the Channel 4 external attenuator.

CH2:PROBEFUNC:EXTUNITS? might return :CH2:PROBEFUNC:EXTUNITS "Pascals", indicating that the Channel 2 external attenuator units of measurement are Pascals.

CH<x>:PROBEFunc:EXTUnits:STATE

This command sets or queries the custom units enable state for the specified channel. The channel is specified by x.

Group Vertical

Syntax CH<x>:PROBEFunc:EXTUnits:STATE {ON|OFF|<NR1>}

Arguments CH<x> is the channel number.

OFF argument turns off external units.

ON argument turns on external units.

<NR1> = 0 turns off external units; any other value turns on external units.

Examples CH2:PROBEFunc:EXTUnits:STATE ON turns on external units.

CH2:PROBEFunc:EXTUnits:STATE? might return CH2:PROBEFUNC:EXTUNITS:STATE 0 indicating that external units are off for the specified channel.

CH<x>:SCAle

This command sets or returns the vertical scale for the specified analog channel. The channel is specified by x.

Group Vertical

Syntax `CH<x>:SCAle <NR3>`

Arguments `CH<x>` is the channel number.

`<NR3>` is the vertical scale for the specified analog channel.

Examples `CH2:SCAle 200E-3` sets the scale to 200 mV per division.

`CH2:SCAle?` might return `CH2:SCALE 500.0000E-3` indicating the vertical scale for the specified channel is 500 mV per division.

CH<x>:SCALERATio

This command sets or returns the scale ration for the specified analog channel.

Group Horizontal

Syntax
```
CH<x>:SCALERATio <NR2>
CH<x>:SCALERATio?
```

Arguments `CH<x>` is the channel number.

`<NR2>` is the scale ration for the specified analog channel.

Examples `CH2:SCALERATio 2.0` sets the scale ratio to 2.0.

`CH2:SCALERATio?` might return `CH2:SCALERATIO 1.0000` indicating the scale ratio is 1.0.

CH<x>:SV:CENTERFrequency

This command sets or queries the center frequency of the specified spectrum trace channel for Spectrum View.

Group Spectrum View

Syntax
```
CH<x>:SV:CENTERFrequency <NR3>
CH<x>:SV:CENTERFrequency?
```

Related Commands SV:LOCKCenter

Arguments CH<x> specifies the spectrum trace channel source.

<NR3> specifies the spectrum trace center frequency for the specified channel, in hertz. The range of values is 0.0 to the maximum licensed bandwidth of the instrument.

Examples CH4:SV:CENTERFrequency 3.0E+06 sets the center frequency of the channel 4 spectrum trace to 3 MHz.

CH2:SV:CENTERFrequency? might return 1.0E+07, indicating that the center frequency of the channel 2 spectrum trace is set to 10 MHz.

CH<x>:SV:POSition

This command sets or queries the Spectrum Time setting of the specified spectrum trace channel in the Spectrum View.

Group Spectrum View

Syntax
```
CH<x>:SV:POSition <NR3>
CH<x>:SV:POSition?
```

Related Commands SV:LOCKSpectrum

Arguments CH<x> specifies the spectrum trace channel source.

<NR3> specifies the spectrum time location, as a percentage of the record length. The range of values is 0.0% to 100%.

Examples CH6:SV:POSition 25 sets the channel 6 spectrum trace spectrum time value to 25%.

CH1:SV:POSition? might return 74.0, indicating that the spectrum time value of the channel 1 spectrum trace is set to 74%.

CH<x>:SV:SPANABovebw? (Query Only)

This command queries whether the stop frequency for the specified spectrum trace channel is above the scope input bandwidth.

Group Spectrum View

Syntax `CH<x>:SV:SPANABovebw?`

Arguments `CH<x>` specifies the spectrum trace channel source.

Returns `0` specifies the stop frequency is not above the scope bandwidth.

`1` specifies the stop frequency is above the scope bandwidth.

Examples `CH3:SV:SPANABovebw?` might return `0`, indicating the stop frequency is not above the scope bandwidth.

CH<x>:SV:SPANBELowdc? (Query Only)

This command queries whether the start frequency for the specified spectrum trace channel is below 0 Hz.

Group Spectrum View

Syntax `CH<x>:SV:SPANBELowdc?`

Arguments `CH<x>` specifies the spectrum trace channel source.

Returns `0` specifies the start frequency is not below 0 Hz.

`1` specifies the start frequency is below 0 Hz.

Examples `CH<x>:SV:SPANBELowdc?` might return `0`, indicating the start frequency is not below 0 Hz.

CH<x>:SV:STARTFrequency? (Query Only)

This command queries the start frequency of the specified spectrum trace channel in the Spectrum View window

Group Spectrum View

Syntax `CH<x>:SV:STARTFrequency?`

Arguments CH<x> specifies the spectrum trace channel source.

Returns <NR3> specifies the start frequency value in hertz.

Examples CH3:SV:STARTFrequency? might return 35.7E+6, indicating the start frequency for the channel 3 spectrum trace is 35.7 MHz.

CH<x>:SV:STATE

This command sets or queries the on/off setting of data acquisition for the specified spectrum trace channel.

Group Spectrum View

Syntax
```
CH<x>:SV:STATE {ON|OFF}
CH<x>:SV:STATE?
```

Arguments CH<x> specifies the spectrum trace channel source.

ON enables spectrum data acquisition for the specified spectrum trace channel source.

OFF disables spectrum data acquisition for the specified spectrum trace channel source.

Examples SV:CH1:STATE ON turns on Spectrum View data acquisition for the channel 1 spectrum trace.

SV:CH3:STATE? might return OFF, indicating that Spectrum View data acquisition is turned off for the channel 3 spectrum trace.

CH<x>:SV:STOPFrequency? (Query Only)

This command queries the stop frequency of the specified spectrum trace channel in the Spectrum View window

Group Spectrum View

Syntax CH<x>:SV:STOPFrequency?

Arguments CH<x> specifies the spectrum trace channel source.

Returns <NR3> specifies the stop frequency value in hertz.

Examples CH2:SV:STOPFrequency? might return 95.1E+6, indicating the stop frequency for the channel 2 spectrum trace is 95.1 MHz.

CH<x>:TERmination

This command sets or queries the vertical termination for the specified analog channel. The channel is specified by x.

NOTE. *The available arguments depend on the instrument model and the attached accessories.*

Group Vertical

Syntax
```
CH<x>:TERmination <NR3>
CH<x>:TERmination?
```

Arguments CH<x> is the channel number.

<NR3> specifies the channel input resistance, which can be specified as 50 Ω or 1,000,000 Ω.

Examples CH4:TERMINATION 50.0E+0 establishes 50 Ω impedance on Channel 1.

CH2:TERMINATION? might return CH2:TERMINATION 50.0E+0, indicating that Channel 2 is set to 50 Ω impedance.

CH<x>:VTERm:BIAS

Sets or queries the termination bias voltage for the specified channel (if control is available).

Conditions Requires a probe with dual inputs.

Group Vertical

Syntax

```
CH<x>:VTERm:BIAS <NR3>
CH<x>:VTERm:BIAS?
```

Arguments

CH<x> is the channel number.

<NR> is the termination voltage.

Examples

CH1:VTERm:BIAS 1.5 sets the termination bias voltage on channel 1 to 1.5 Volts

CH2:VTERm:BIAS? might return CH2:VTERm:BIAS 0.0000, indicating that the termination bias voltage for channel 2 is set to 0 volts.

CH<x>_DALL:LABel:COLor

This command sets or queries the color of the specified digital group label. The channel is specified by x.

Group

Digital

Syntax

```
CH<x>_DALL:LABel:COLor <QString>
```

Arguments

CH<x> is the channel number.

<QString> is the color of the digital group label. To return the color to the default color, send an empty string as in this example: :CH5_DALL:LABEL:COLOR "".

Examples

CH1_DALL:LABel:COLor "#FF0000" sets the font color to red.

CH1_DALL:LABel:COLor? might return CH1_DALL:LABEL:COLOR "#FFFF00" indicating the font color is yellow.

CH<x>_DALL:LABel:FONT:BOLD

This command sets or queries the bold state of the specified digital group. The channel is specified by x.

Group

Digital

Syntax

```
CH<x>_DALL:LABel:FONT:BOLD {ON|OFF|<NR1>}
```

Arguments CH<x> is the channel number.

OFF argument turns off bold font.

ON argument turns on bold font.

<NR1> = 0 turns off bold font; any other value turns on bold font.

Examples CH1_DALL:LABel:FONT:BOLD ON sets the font to bold.

CH1_DALL:LABel:FONT:BOLD? might return CH1_DALL:LABEL:FONT:BOLD 0 indicating the font is not bold.

CH<x>_DALL:LABel:FONT:ITALic

This command sets or queries the italic state of the specified digital group. The channel is specified by x.

Group Digital

Syntax CH<x>_DALL:LABel:FONT:ITALic {ON|OFF|<NR1>}

Arguments CH<x> is the channel number.

OFF argument turns off italic font.

ON argument turns on italic font.

<NR1> = 0 turns off italic font; any other value turns on italic font.

Examples CH1_DALL:LABel:FONT:ITALic 1 turns on italic font.

CH1_DALL:LABel:FONT:ITALic? might return CH1_DALL:LABEL:FONT:ITALIC 0 indicating the font is not italic.

CH<x>_DALL:LABel:FONT:SIZE

This command sets or queries the font size of the specified digital group. The channel is specified by x.

Group Digital

Syntax CH<x>_DALL:LABel:FONT:SIZE <NR1>

Arguments CH<x> is the channel number.

<NR1> is the font size.

Examples CH1_DALL:LABel:FONT:SIZE 16 sets the font size to 16 points.

CH1_DALL:LABel:FONT:SIZE? might return CH1_DALL:LABEL:FONT:SIZE 20 indicating the font size is 20 points.

CH<x>_DALL:LABel:FONT:TYPE

This command sets or queries the font type of the specified digital group, such as Arial or Times New Roman. The channel is specified by x.

Group Digital

Syntax CH<x>_DALL:LABel:FONT:TYPE <QString>

Arguments CH<x> is the channel number.

<QString> is the font type.

Examples CH1_DALL:LABel:FONT:TYPE "Monosapce" sets the font to a monospace font.

CH1_DALL:LABel:FONT:TYPE? might return CH1_DALL:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman".

CH<x>_DALL:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified digital group. The channel is specified by x.

Group Digital

Syntax CH<x>_DALL:LABel:FONT:UNDERline {ON|OFF|<NR1>}

Arguments CH<x> is the channel number.

OFF argument turns off underline font.

ON argument turns on underline font.

<NR1> = 0 turns off underline font; any other value turns on underline font.

Examples CH1_DALL:LABel:FONT:UNDERline ON specifies an underlined font.

CH1_DALL:LABel:FONT:UNDERline? might return CH1_DALL:LABEL:FONT:UNDERLINE 0 indicating underline is off.

CH<x>_DALL:LABel:NAMe

This command sets or queries the label of the specified digital group. The channel is specified by x.

Group Digital

Syntax CH<x>_DALL:LABel:NAMe <QString>

Arguments CH<x> is the channel number.

<QString> is the name of the group.

Examples CH1_DALL:LABel:NAMe "Clock Out" sets the label name to Clock Out.

CH1_DALL:LABel:NAMe? might return CH1_DALL:LABEL:NAME "This is the digital name".

CH<x>_D<x>:LABel:COLor

This command sets or queries the color of the label of the specified digital bit. The channel is specified by x.

Group Digital

Syntax CH<x>_D<x>:LABel:COLor <QString>

Arguments CH<x> is the channel number.

<QString> is the label color. To return the color to the default color, send an empty string as in this example: :CH5_D1:LABEL:COLOR "".

Examples `CH1_D1:LABel:COLor "#FF0000"` sets the color to red.

`CH1_D1:LABel:COLor?` might return `CH1_D1:LABEL:COLOR "#FFFF00"` indicating the color is yellow.

CH<x>_D<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the label of the specified digital bit. The channel is specified by x.

Group Digital

Syntax `CH<x>_D<x>:LABel:FONT:BOLD {ON|OFF|<NR1>}`

Arguments `CH<x>` is the channel number.

`OFF` argument turns off bold font.

`ON` argument turns on bold font.

`<NR1>` = 0 turns off bold font; any other value turns on bold font.

Examples `CH1_D1:LABel:FONT:BOLD ON` sets the font to bold.

`CH1_D1:LABel:FONT:BOLD?` might return `CH1_D1:LABEL:FONT:BOLD 0` indicating the font is not bold.

CH<x>_D<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the label of the specified digital bit. The channel is specified by x.

Group Digital

Syntax `CH<x>_D<x>:LABel:FONT:ITALic {ON|OFF|<NR1>}`

Arguments `CH<x>` is the channel number.

`OFF` argument turns off italic font.

`ON` argument turns on italic font.

`<NR1>` = 0 turns off italic font; any other value turns on italic font.

Examples CH1_D1:LABel:FONT:ITALic OFF turns off italic font.

CH1_D1:LABel:FONT:ITALic? might return CH1_D1:LABEL:FONT:ITALIC 1 indicating the font is italic.

CH<x>_D<x>:LABel:FONT:SIZE

This command sets or queries the font size of the label of the specified digital bit. The channel is specified by x.

Group Digital

Syntax CH<x>_D<x>:LABel:FONT:SIZE <NR1>

Arguments CH<x> is the channel number.

<NR1> is the font size.

Examples CH1_D1:LABel:FONT:SIZE 16 sets the font size to 16 points.

CH1_D1:LABel:FONT:SIZE? might return CH1_D1:LABEL:FONT:SIZE 20 indicating the font size is 20 points.

CH<x>_D<x>:LABel:FONT:TYPE

This command sets or queries the font type of the label of the specified digital bit, such as Arial or Times New Roman. The channel is specified by x.

Group Digital

Syntax CH<x>_D<x>:LABel:FONT:TYPE <QString>

Arguments CH<x> is the channel number.

<QString> is the font type of the label.

Examples CH1_D1:LABel:FONT:TYPE "Monospace" sets the font to Monospace.

CH1_D1:LABel:FONT:TYPE? might return CH1_D1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman".

CH<x>_D<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the label of the specified digital bit. The channel is specified by x.

Group Digital

Syntax `CH<x>_D<x>:LABel:FONT:UNDERline {ON|OFF|<NR1>}`

Arguments `CH<x>` is the channel number.

`OFF` argument turns off underline font.

`ON` argument turns on underline font.

`<NR1>` = 0 turns off underline font; any other value turns on underline font.

Examples `CH1_D1:LABel:FONT:UNDERline ON` turns on underline font.

`CH1_D1:LABel:FONT:UNDERline?` might return `CH1_D1:LABEL:FONT:UNDERLINE 0` indicating the underline font is off.

CH<x>_D<x>:LABel:NAMe

Sets or queries the label of the specified digital bit. The channel is specified by x.

Group Digital

Syntax `CH<x>_D<x>:LABel:NAMe <QString>`

Arguments `CH<x>` is the channel number.

`<QString>` is the label.

Examples `CH1_D1:LABel:NAMe "Clock in"` sets the name to Clock in.

`CH1_D1:LABel:NAMe?` might return `CH1_D1:LABEL:NAME "Digital 1"`.

CLEAR (No Query Form)

This command clears acquisitions, measurements, and waveforms.

Group Miscellaneous

Syntax CLEAR

Examples CLEAR clears all acquisitions, measurements, and waveforms.

*CLS (No Query Form)

This command (no query form) clears the following:

- Event Queue
- Standard Event Status Register
- Status Byte Register (except the MAV bit)

If the *CLS command immediately follows an <EOI>, the Output Queue and MAV bit (Status Byte Register bit 4) are also cleared. MAV indicates that information is in the output queue. The device clear (DCL) control message will clear the output queue and thus MAV. *CLS does not clear the output queue or MAV.

*CLS can suppress a Service Request that is to be generated by an *OPC. This will happen if a single sequence acquisition operation is still being processed when the *CLS command is executed.

Group Status and Error

Syntax *CLS

Related Commands DESE

*ESE

*ESR?

EVENT?

EVMsg?

*SRE

*STB?

Examples *CLS clears the instrument status data structures.

CONFIGuration:ANALOg:BANDWidth? (Query Only)

This command queries the maximum licensed bandwidth of the instrument.

Group Vertical

Syntax `CONFIGuration:ANALOg:BANDWidth?`

Returns The maximum licensed bandwidth of the instrument is returned.

Examples `CONFIGuration:ANALOg:BANDWidth?` might return `CONFIGURATION:ANALOG:BANDWIDTH 2.0000E+9` indicating the bandwidth is 2.0 GHz.

CONNected:REQUested:STATus (No Query Form)

This command sets the status of the Connected Scope Preference feature. After issuing the status command, the `CONNected:SAVEPREferences` command should be set immediately to make the feature enabled. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group Miscellaneous

Syntax `CONNected:REQUested:STATus <NR1>`

Arguments `<NR1>` enables or disables the Connected Scope Preference feature. The argument can be either 1 or 0. Setting 1 will enable the feature.

Examples `CONNected:REQUested:STATus 1 CONNected:SAVEPREferences` enables and saves Connected Scope Preference feature.

CONNected:STATus? (Query Only)

This query command returns the status of the Connected Scope Preference feature. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group Miscellaneous

Syntax CONNected:STATus?

Returns This command returns either 1 or 0. A return value of 1 means the feature is enabled and 0 means the feature is disabled.

Examples CONNected:STATus? might return :CONNected:STATus 1, indicating the Connected Scope Preference feature is enabled.

CONNected:USAGe:TRack:REQUested:STATus (No Query Form)

This command sets the tracking usage status of the Connected Scope Preference feature. After issuing the status command, the CONNected:SAVEPREferences command should be set immediately to make the feature enabled. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group Miscellaneous

Syntax CONNected:USAGe:TRack:REQUested:STATus<NR1>

Arguments <NR1> enables or disables the Connected Scope Preference feature. The argument can be either 1 or 0. Setting 1 will enable the tracking usage status of the Connected Scope Preference feature.

Examples CONNected:USAGe:TRack:REQUested:STATus 1
CONNected:SAVEPREferences enables and saves the tracking usage status of the Connected Scope Preference feature.

CONNected:USAGe:TRack:STATus? (Query Only)

This query command returns the tracking usage status of the Connected Scope Preference feature. On the instrument, the feature is found in Utility > User Preferences > Other > Connected Scope Preferences.

Group Miscellaneous

Syntax CONNected:USAGe:TRack:STATus?

Returns This command returns either 1 or 0. A return value of 1 means the feature is enabled and 0 means the feature is disabled.

Examples `CONNected:USAGe:TRack:STATus?` might return `:CONNected:USAGe:TRack:STATus 1`, indicating the tracking usage status of the Connected Scope Preference feature is enabled.

CURVe

This command transfers waveform data to and from the instrument. Each waveform that is transferred has an associated waveform preamble that contains information such as data format and scale.

The `CURVe?` query transfers data from the instrument. The data source is specified by the DATa:SOUrce command. The first and last data points are specified by the DATa:STARt and DATa:STOP commands.

The `CURVe` command transfers waveform data to the instrument. Only one waveform can be transferred at a time. The waveform will only be displayed if the reference is displayed.

For digital sources, CH<x>_D<n> or CH<x> _DALL, when the :DATa:WIDth is 1, the returned data is state only. When the :DATa:WIDth is 2, the returned data is transition data with 2 bits per digital channel representing the transition information as follows:

- 0 0 low
- 0 1 high
- 1 1 multiple transitions in interval ending with high
- 1 0 multiple transitions in interval ending with low

For individual digital channels (such as CH<x>_D<n>), :DATa:WIDth 2 provides the 2-bit transition data with the upper 14 bits zero. :DATa:WIDth 1 provides only the state in the LSB with the upper 7 bits all zero.

For CH<x>_DAll sources, :DATa:WIDth 2 provides the 2-bit transition data for each of the 8 constituent channels with the D7 bit represented in the 2 most significant bits, D6 in the next 2, and so on. :DATa:WIDth 1 provides the states of each of the 8 constituent channels with D7 represented in the most significant bit and D0 in the least significant bit.

Depending on the sample rate, multi-transition data may not be available and :CURVe? queries for digital channels with :DATa:WIDth 2 may result in a warning event "Execution warning: Multi-transition data not available". In this case, the transition data returned will be 0 0 or 0 1.

For MATH sources, only 8-byte double precision floating point data is returned in :CURVe? queries.

Group Waveform Transfer

Syntax

```
CURVe {<Block>|<asc curve>}
CURVe?
```

Related Commands DATa:SOUrce

DATa:STARt

DATa:STOP

SAVe:WAVEform

SAVEON:WAVEform:FILEFormat

WFMOutpre?

Arguments <Block> is the waveform data in binary format. The waveform is formatted as: #<x><yyy><data><newline>, where:

<x> is the number of y bytes. For example, if <yyy>=500, then <x>=3)

NOTE. *<x> is hexadecimal format. The letters A-F denote several y bytes between 10 and 15 digits.*

<yyy> is the number of bytes to transfer. If width is 1, then all bytes on the bus are single data points. If width is 2, then all bytes on the bus are 2-byte pairs. If width is 4, then all bytes on the bus are 4-byte pairs.

<data> is the curve data.

<newline> is a single byte new line character at the end of the data.

<asc curve> is the waveform data in ASCII format. The format for ASCII data is <NR1>[,<NR1>...], where each <NR1> represents a data point.

Returns CH<x>_SV_NORMal, CH<x>_SV_AVErage, CH<x>_SV_MAXHold, CH<x>_SV_MINHold: When DATa:SOUrce is set to one of these traces, the data points represent the amplitude of the Spectrum View frequency domain traces in the current units, and are floating point values. The frequency domain trace data is returned as 8-byte floating point values.

CH<x>_SV_AMPlitude, CH<x>_SV_FREQuency, CH<x>_SV_PHASe: When DATa:SOUrce is set to one of these traces, the data points are the Magnitude vs. Time, Frequency vs. Time, or Phase vs. Time representations of the RF input signal (Spectrum View) time domain traces. The RF time domain traces are returned as floating point values.

CH<x>_FREQ_vs_time is the frequency in Hz.

CH<x>_PHASE_vs_time is the phase in degrees.

CH<x>_MAG_vs_time is the magnitude in Volts.

CH<x>_SV_BASEband_IQ is the Spectrum View RF baseband IQ traces. When DATa:SOUrce is set to one of these traces, the data points represent the digitizing levels of the IQ data. The IQ data is formatted as pairs of integer values (i.e. <I>,<Q>,<I>,<Q>,...)

Examples

CURVE <Block> sets the format of the waveform data, transferred to and from the instrument, to binary format.

CURVE? with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return :CURVE 61,62,61,60,60,-59,-59,-58,-58,-59

NOTE. *Curve data is transferred from the instrument asynchronously and, depending upon the length of the curve record, such transfers can require several seconds to complete. During this time, the instrument will not respond to user controls. You can interrupt these asynchronous data transfers by sending a device clear message to the instrument or by interrupting the query with another command or query. Verify that curve data is completely transferred.*

*It is recommended that you follow such queries with an *ESR? query and verify that the error bit returned and, if set, check the event queue to ascertain the reason for the error. If the error was caused by an interrupted query, then the asynchronous data transfer had not completed when the *ESR? query was sent. In this case, you might need increase your program's time-out value to ensure that all data is transferred and read.*

CUSTOMTABle:ADDNew (No Query Form)

This command adds new custom results table.

Group Measurement, IMDA

Syntax CUSTOMTABle:ADDNew <Qstring>

Arguments <Qstring> specifies the custom results table name.

Examples CUSTOMTABLE:ADDNEW"TABLE1" adds a new custom table named TABLE1.

CUSTOMTABle:DELete (No Query Form)

This command deletes the custom result(s) table that was added.

Group Measurement, IMDA

Syntax `CUSTOMTABle:DELete <Qstring>`

Arguments `<Qstring>` specifies the custom results table name.

Examples `CUSTOMTABLE:DELETE"TABLE1"` deletes the new custom table named TABLE1.

CUSTOMTABle:LIST? (Query Only)

This command queries the list of custom result tables has been added.

Group Measurement, IMDA

Syntax `CUSTOMTABle:LIST? <Qstring>`

Arguments `<Qstring>` specifies the custom results table name.

Examples `CUSTOMTABLE:LIST?` might return `:CUSTOMTABle:LIST TABLE1, TABLE2`, indicating there are two custom result tables.

DATa

This command sets or queries the format and location of the waveform data that is transferred with the CURVe command.

Group Waveform Transfer

Syntax
```
DATa {INIT|SNAp}
DATa?
```

Related Commands

CURVe

DATa:STARt

DATa:STOP

DATa:ENCdg

SAVEON:WAVEform:FILEFormat

WFMOutpre:NR_Pt?

Arguments

`INIT` initializes the waveform data parameters to their factory defaults except for DATa:STOP, which isset to the current acquisition record length.

`SNAp` Sets DATa:STARt and DATa:STOP to match the current waveform cursor positions of WAVEVIEW1 CURSOR1 if these waveform cursors are currently on. If these waveform cursors are not on when the DATa SNAp command is sent, it is silently ignored and DATa:STARt and :STOP remain unchanged.

If either of the cursors is outside the record, DATa:STARt and :STOP are set to nearest values within the record. This will cause DATa:STARt and/or :STOP to snap to the beginning or end of the record whichever is nearest, and the following event is set: 500, "Execution warning", "One or both cursors outside of waveform record".

If there is no acquired waveform for the cursor source, then DATa:STARt and :STOP remain unchanged and the following event is set: 500, "Execution warning", "No acquired waveform record".

If DATa:STARt is greater than :STOP, the values of DATa:STARt and STOp are swapped and the following event is set: 530, "Data start > stop, Values swapped internally".

Examples

`DATA INIT` initializes the waveform data parameters to their factory defaults.

`DATA?` might return `DATA:ENCDG RIBINARY;SOURCE CH1;START 1;STOP 1000;FRAMESTART 1;FRAMESTOP 1000`.

DATa:ENCdg

This command sets or queries the format of outgoing waveform data. This command is equivalent to setting WFMOutpre:ENCdg, WFMOutpre:BN_Fmt, and WFMOutpre:BYT_Or. Setting the DATa:ENGdg value causes the corresponding WFMOutpre values to be updated and vice versa.

NOTE. *Values are constrained (for outbound data) to the format of the data specified by DATa:SOUrce.*

Group Waveform Transfer

Syntax

```
DATa:ENCdg {ASCIi|RIBinary|RPBinary|FPBinary|SRIbinary|
SRPbinary|SFPbinary}
DATa:ENCdg?
```

Related Commands WFMOutpre:ENCdg

WFMOutpre:BN_Fmt

WFMOutpre:BYT_Or

Arguments

- `ASCIi` specifies the ASCII representation of signed INT, FLOAT. If ASCII is the value, then :BN_Fmt and :BYT_Or are ignored.
- `RIBinary` specifies signed integer data point representation with the most significant byte transferred first.

  When :BYT_Nr is 1, the range is from -128 through 127. When :BYT_Nr is 2, the range is from -32,768 through 32,767. When :BYT_Nr is 8, then the waveform being queried is set to Fast Acquisition mode. Center screen is 0 (zero). The upper limit is the top of the screen and the lower limit is the bottom of the screen. This is the default argument.

- RPBinary specifies the positive integer data-point representation, with the most significant byte transferred first.

  When :BYT_Nr is 1, the range from 0 through 255. When :BYT_Nr is 2, the range is from 0 to 65,535. When :BYT_Nr is 8, then the waveform being queried is set to Fast Acquisition mode. The center of the screen is 127. The upper limit is the top of the screen and the lower limit is the bottom of the screen.

- FPBinary specifies the floating point (width = 4) data.

  The range is from -3.4×10^{38} to 3.4×10^{38}. The center of the screen is 0. The upper limit is the top of the screen and the lower limit is the bottom of the screen.

  The FPBinary argument is only applicable to math waveforms or ref waveforms saved from math waveforms.

- SRIbinary is the same as RIBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to IBM compatible PCs.

- SRPbinary is the same as RPBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.

- SFPbinary specifies floating point data in IBM PC format. The SFPbinary argument only works on math waveforms or ref waveforms saved from math waveforms.

Table 2-50: DATa and WFMOutpre Parameter Settings

	WFMOutpre Settings			
DATa:ENCdg Setting	**:ENCdg**	**:BN_Fmt**	**:BYT_Or**	**:BYT_NR**
ASCii	ASC	N/A	N/A	1,2,4,8
RIBinary	BIN	RI	MSB	1,2,8
RPBinary	BIN	RP	MSB	1,2,8
FPBinary	BIN	FP	MSB	4
SRIbinary	BIN	RI	LSB	1,2,8
SRPbinary	BIN	RP	LSB	1,2,8
SFPbinary	BIN	FP	LSB	4,8

Examples

DATA:ENCDG RPBinary sets the data encoding format to be a positive integer where the most significant byte is transferred first.

DATA:ENCDG? might return DATa:ENCDG SRPBINARY for the format of the outgoing waveform data.

DATa:RESample

This command sets or queries the resampling of outgoing waveform data. This command is equivalent to setting `WFMOutpre:RESample`. Setting the `DATa:RESample` value causes the corresponding WFMOutpre value to be updated and vice versa.

Group Waveform Transfer

Syntax

```
DATa:RESample <NR1>
DATa:RESample?
```

Related Commands WFMOutpre:RESample

Arguments `<NR1>` is the resampling rate. The default value is 1, which means every sample is returned. A value of 2 returns every other sample, while a value of 3 returns every third sample, and so on.

Examples `DATA:RESAMPLE 1` sets the resampling to return all waveform data samples.

`DATA:RESAMPLE?` might return `:DATa:RESAMPLE 4`, indicating that every forth sample of the waveform data is returned.

DATa:SOUrce

This command sets or queries the location of waveform data that is transferred from the instrument by the CURVe? Query.

Group Waveform Transfer

Syntax

```
DATa:SOUrce <wfm>[<,><wfm>]
DATa:SOUrce?
```

Related Commands CURVe

DATa

Arguments `<wfm>` is the location of the waveform data that will be transferred from the instrument to the controller. It can consist of CH<x>, MATH<x>, REF<x>, DIGITALALL. Note that digital data is transferred as 16-bit data, with the least-significant bit representing D0, and the most-significant bit representing D15.

<wfm> can consist of the following:

CH<x> selects the specified analog channel as the source.

MATH<x> selects the specified reference waveform as the source. The reference number is specified by x, which ranges from 1 through 4.

REF<x> selects the specified reference waveform as the source. The reference number is specified by x, which ranges from 1 through 8.

CH<x>_D<x> selects the specified digital channel.

CH<x>_DAll selects the specified channel group of digital channels.

DIGITALALL selects digital waveforms as the source. The Digital data is transferred as 16-bit data, with the least-significant bit representing D0, and the most-significant bit representing D15. The LSB always contains D0-D7 and MSB always contains D8-D15 data.

CH<x>_SV_NORMal, CH<x>_SV_AVErage, CH<x>_SV_MAXHold, CH<x>_SV_MINHold selects the specified Spectrum View waveform.

CH<x>_MAG_VS_TIME, CH<x>_FREQ_VS_TIME, CH<x>_PHASE_VS_TIME selects the specified RF vs. Time waveform.

CH<x>_SV_BASEBAND_IQ selects the specified RF baseband IQ data.

Examples

DATA:SOURCE CH1 specifies that the CH1 waveforms will be transferred in the next CURVe? query.

DATA:SOURCE? might return DATA:SOURCE REF3, indicating that the source for the waveform data which is transferred using a CURVe? query is reference 3.

DATa:SOUrce:AVAILable? (Query Only)

This query returns a list of enumerations representing the source waveforms that are currently available for :CURVe? queries. This means that the waveforms have been acquired. If there are none, NONE is returned.

Group

Waveform Transfer

Syntax

DATa:SOUrce:AVAILable?

Related Commands

CURVe

Returns

Returns a list of source waveforms that are currently available for :CURVe? queries.

Examples

DATA:SOURCE:AVAILABLE? might return DATA:SOURCE:AVAILABLE CH2,CH3,CH6,CH7,MATH1,REF1 indicating that CH2, CH3, CH6, CH7, MATH1, and REF1 are available.

DATa:STARt

This command sets or queries the starting data point for waveform transfer. This command allows for the transfer of partial waveforms to and from the instrument.

Group

Waveform Transfer

Syntax

```
DATa:STARt <NR1>
DATa:STARt?
```

Related Commands

CURVe

DATa

DATa:STOP

WFMOutpre:NR_Pt?

Arguments

<NR1> is the first data point that will be transferred, which ranges from 1 to the record length. Data will be transferred from <NR1> to DATa:STOP or the record length, whichever is less. If <NR1> is greater than the record length, the last data point in the record is transferred.

DATa:STARt and DATa:STOP are order independent. When DATa:STOP is greater than DATa:STARt, the values will be swapped internally for the CURVE? query.

Examples

DATA:START 10 specifies that the waveform transfer will begin with data point 10.

DATA:START? might return DATA:START 214, indicating that data point 214 is the first waveform data point that will be transferred.

DATa:STOP

This command sets or queries the last data point that will be transferred when using the CURVe? query. This command allows for the transfer of partial waveforms to the controller.

Changes to the record length value are not automatically reflected in the data:stop value. As record length is varied, the DATa:STOP value must be explicitly changed to ensure the entire record is transmitted. In other words, curve results will not

automatically and correctly reflect increases in record length if the distance from DATa:STARt to DATa:STOP stays smaller than the increased record length.

NOTE. *When using the CURVe command, DATa:STOP is ignored.*

Group Waveform Transfer

Syntax

```
DATa:STOP <NR1>
DATa:STOP?
```

Related Commands CURVe

DATa

DATa:STARt

WFMOutpre:NR_Pt?

Arguments <NR1> is the last data point that will be transferred, which ranges from 1 to the record length. If <NR1> is greater than the record length, then data will be transferred up to the record length. If both DATa:STARt and DATa:STOP are greater than the record length, the last data point in the record is returned.

DATa:STARt and DATa:STOP are order independent. When DATa:STOP is less than DATa:STARt, the values will be swapped internally for the CURVE? query.

If you always want to transfer complete waveforms, set DATa:STARt to 1 and DATa:STOP to the maximum record length, or larger.

Examples DATA:STOP 15000 specifies that the waveform transfer will stop at data point 15000.

DATA:STOP? might return DATA:STOP 14900, indicating that 14900 is the last waveform data point that will be transferred.

DATa:WIDth

This command specifies the width, in bytes per point, for waveform data transferred from the instrument via the CURVe? query. (This command is synonymous with WFMOutpre:BYT_Nr.)

Group Waveform Transfer

Syntax

```
DATa:WIDth <NR1>
DATa:WIDth?
```

Related Commands

WFMOutpre:BYT_Nr

Arguments

<NR1> is an integer that indicates the number of bytes per point for the outgoing waveform data when queried using the CURVe? command.

- For analog channels, NR1 can be 1 or 2.
- For digital channels, NR1 can be 1 or 2.
- For the digital collection, NR1 can be 4 or 8.
- For CH<x>_SV_NORMal, CH<x>_SV_AVErage, CH<x>_SV_MAXHold, CH<x>_SV_MINHold, CH<x>_MAG_VS_TIME, CH<x>_FREQ_VS_TIME, and CH<x>_PHASE_VS_TIME, NR1 can only be 8.
- For CH<x>_SV_BASEBAND_IQ, NR1 can be 1 or 2.

Examples

DATA:WIDTH 1 sets the data width to 1 byte.

DATA:WIDTH? might return DATA:WIDTH 1 indicating the width, in bytes per point, for waveform data transferred bya the CURVE? query is 1 byte.

DATE? (Query Only)

This command queries the date that the instrument displays.

Group

Miscellaneous

Syntax

```
DATE?
```

Related Commands

TIMe

Returns

<QString> is a date in the form “yyyy-mm-dd” where yyyy refers to a four-digit year number, mm refers to a two-digit month number from 01 to 12, and dd refers to a two-digit day number in the month.

Examples

DATE? might return DATE 2000-01-24, indicating the current date is set to January 24, 2000.

*DDT

This command allows you to specify a command or a list of commands that are executed when the instrument receives a *TRG command. Define Device Trigger (*DDT) is a special alias that the *TRG command uses.

Group Miscellaneous

Syntax

```
*DDT {<Block>|<QString>}
*DDT?
```

Related Commands ALIas

*TRG

Arguments <Block> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters. The format of this argument is always returned as a query.

<QString> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters.

Examples *DDT #217ACQUIRE:STATE RUN specifies that the acquisition system will be started each time a *TRG command is sent.

DESE

This command sets and queries the bits in the Device Event Status Enable Register (DESER). The DESER is the mask that determines whether events are reported to the Standard Event Status Register (SESR), and entered into the Event Queue. For a more detailed discussion of the use of these registers, see Registers.

Group Status and Error

Syntax

```
DESE <NR1>
DESE?
```

Related Commands

*CLS

*ESE

*ESR?

EVENT?

EVMsg?

*SRE

*STB?

Arguments

<NR1> The binary bits of the DESER are set according to this value, which ranges from 1 through 255. For example, DESE 209 sets the DESER to the binary value 11010001 (that is, the most significant bit in the register is set to 1, the next most significant bit to 1, the next bit to 0, etc.).

The power-on default for DESER is all bits set if *PSC is 1. If *PSC is 0, the DESER maintains the previous power cycle value through the current power cycle.

NOTE. *Setting the DESER and ESER to the same value allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the *ESE command to set the ESER.*

Examples

`DESE 209` sets the DESER to binary 11010001, which enables the PON, URQ, EXE and OPC bits.

`DESE?` might return `DESE 186`, showing that the DESER contains the binary value 10111010.

DIAg:LOOP:OPTion

This command sets or queries the type of looping desired.

Group

Self Test

Syntax

```
DIAg:LOOP:OPTion {FAIL|ONCE|ALWAYS|ONFAIL|NTIMES}
DIAg:LOOP:OPTion?
```

Arguments

`Fail` - run until a failure is found, then halt.

`Once` - run through one loop.

`Always` - run forever.

`Onfail` - run until a failure is found, then loop on it.

`Ntimes` - run n number of loops.

Examples `DIAG:LOOP:OPTION ALWAYS` sets loop option to `ALWAYS`.

`DIAG:LOOP:OPTION?` returns current looping option value.

DIAg:LOOP:OPTion:NTIMes

This command sets or queries how many loops to run, if N-times is being used.

Group Self Test

Syntax
```
DIAg:LOOP:OPTion:NTIMes <NR1>
DIAg:LOOP:OPTion:NTIMes?
```

Arguments `<NR1>` is how many loops to run.

Examples `DIAG:LOOP:OPTION:NTIMES 2` sets diagnostics to loop 2 times.

`DIAG:LOOP:OPTION:NTIMES?` might return `DIAG:LOOP:OPTION:NTIMES 1` indicating diagnostics will run once.

DIAg:LOOP:STOP (No Query Form)

Request that diagnostics stop looping.

Group Self Test

Syntax
```
DIAg:LOOP:STOP
```

Examples `DIAG:LOOP:STOP` stops diagnostics looping.

DIAg:MODe

This command sets or queries the diagnostics mode.

Group Self Test

Syntax

```
DIAg:MODe {POST|EXTENDED|SERVICE}
DIAg:MODe?
```

Arguments

`POST` specifies the power on self test diagnostics.

`EXTENDED` specifies the extended diagnostics.

`SERVICE` specifies the service diagnostics.

Examples

`DIAG:MODE POST` specifies the power on self test diagnostics.

DIAg:RESUlt? (Query Only)

This query returns both the overall diagnostics test results and the results of each individual test area.

Group

Self Test

Syntax

```
DIAg:RESUlt?
```

Returns

The diagnostics results.

Examples

`DIAG:RESULT?` might return `DIAG:RESULT:FLAG "NOT RUN";LOG "PASS--IO,PASS--ANALOG,PASS--SYSTEM,PASS--ASIC,PASS--ACQ,PASS--SIGNAL,PASS--MEMORY"` indicating the diagnostics results.

DIAg:RESUlt:FLAg? (Query Only)

This query returns the status of the diagnostic test area that has been selected.

Group

Self Test

Syntax

```
DIAg:RESUlt:FLAg?
```

Related Commands

DIAg:RESUlt:FLAg?

Returns

The status of the diagnostics (single area).

Examples `DIAG:RESULT:FLAG?` might return `DIAG:RESULT:FLAG "NOT RUN"` indicating the diagnostics have not been run.

DIAg:RESUlt:LOG? (Query Only)

This query returns the test Pass/Fail status of each diagnostic area. It does not return the overall status.

Group Self Test

Syntax `DIAg:RESUlt:LOG?`

Returns The status of the diagnostic area.

Examples `DIAG:RESULT:LOG?` might return `:DIAG:RESULT:LOG "PASS--IO,PASS--ANALOG,PASS--SYSTEM,PASS--ASIC,PASS--ACQ,PASS--SIGNAL,PASS--MEMORY"` indicating the diagnostics result stored in the log.

DIAg:SELect (No Query Form)

This command selects or queries an available diagnostic area.

Group Self Test

Syntax
```
DIAg:SELect {ALL|IO|ANALOG|SYSTEM|ASIC|ACQ|SIGNAL|MEMORY}
DIAg:SELect?
```

Arguments The argument is the desired diagnostic area.

Examples `DIAG:SELECT IO` will select the `IO` group.

DIAg:STATE (No Query Form)

This command starts or aborts Self Test. Abort happens after group under test completes.

Group Self Test

Syntax `DIAg:STATE {EXECute|ABOrt}`

Arguments `EXECUTE` starts execution of the diagnostics.

`ABOrt` disables diagnostics capabilities and returns the instrument to a normal operating state.

Examples `DIAG:STATE ABORT` turns off diagnostics capabilities and returns the instrument to a normal operating state.

`DIAG:STATE?` might return `DIAG:STATE ABORT`, indicating that diagnostics are disabled.

DIGGRP<x>:D<x>:THReshold

Sets or queries the threshold level in volts for the specified digital channel. If the source channel doesn't exist, a hardware missing error event is set.

Group Digital

Syntax
```
DIGGRP<x>:D<x>:THReshold <NR3>
DIGGRP<x>:D<x>:THReshold?
```

Arguments `DIGGRP<x>` is the channel number.

`D<x>` is the digital channel bit number (0-7).

`<NR3>` is the threshold level in volts for the specified digital channel.

Examples `DIGGRP1:D1:THReshold 30.0E-3` sets the threshold to 30 mV.

`DIGGRP1:D1:THReshold?` might return `DIGGRP1:D1:THRESHOLD 0.0E+0` indicating the threshold is 0.0 V.

DISplay? (Query Only)

This query-only command returns the current Display settings.

Group Display Control

Syntax

```
DISplay?
```

Examples

DISPLAY? might return DISPLAY:CLOCK 1;COLOR:PALETTE:IMAGEVIEW MONOGREEN; RECORDVIEW TEMPERATURE;USER:GRATICULE 165,50,15;CH1 180,50,100; CH2 300,50,100;CH3 60,50,100;CH4 240,50,100;REF1 0,90,0; REF2 0,90,100;REF3 60,90,100;REF4 240,90,100;MATH1 160,50,100; MATH2 40,60,100;MATH3 120,60,100;MATH4 195,50,100; HISTOGRAM 320,50,100;CARET 150,50,100;MASK 0,25,75;MASKHIGHLIGHT 140,50,100;:DISPLAY:COLOR:MATHCOLOR DEFAULT;REFCOLOR DEFAULT; :DISPLAY:FILTER SINX;FORMAT YT;GRATICULE IRE;INTENSITY :WAVEFORM:IMAGEVIEW 81.0000;RECORDVIEW 81.0000;:DISPLAY :INTENSITY:AUTOBRIGHT 0; :DISPLAY:PERSISTENCE OFF;STYLE DOTS;TRIGBAR OFF;TRIGT 1; CURSORTICK LONG;VARPERSIST 2.6000;SCREENTEXT:STATE 1; LABEL1:NAME "";XPOS 100;YPOS 5;:DISPLAY:SCREENTEXT :LABEL2:NAME "THIS IS SCREEN TEXT";XPOS 100;YPOS 20; :DISPLAY:SCREENTEXT:LABEL3:NAME "";XPOS 100;YPOS 35; :DISPLAY:SCREENTEXT:LABEL4:NAME "";XPOS 100;YPOS 50; :DISPLAY:SCREENTEXT:LABEL5:NAME "";XPOS 100;YPOS 343; :DISPLAY:SCREENTEXT:LABEL6:NAME "";XPOS 100;YPOS 80; :DISPLAY:SCREENTEXT:LABEL7:NAME "";XPOS 100;YPOS 95; :DISPLAY:SCREENTEXT:LABEL8:NAME "";XPOS 100;YPOS 110; :DISPLAY:WAVEFORM 1.

DISplay:{CH<x>|Math<x>|REF<x>}:INVERTColor

This command sets or queries the Inverted mode color of the specified input source to the specified color. You can assign one of 48 unique colors to any channel, math, or reference waveform. These colors replace the default Inverted colors and remain in effect until you reset the colors.

Group

Display

Syntax

```
DISplay:{CH<x>|Math<x>|REF<x>}:INVERTColor COLOR<y>
DISplay:{CH<x>|Math<x>|REF<x>}:INVERTColor?
```

Arguments

CH<x> specifies the input channel for which you want to change the waveform color, where <x> is the channel number.

Math<x> specifies the math waveform for which you want to change the waveform color, where <x> is the math waveform number.

REF<x> specifies the reference waveform for which you want to change the waveform color, where <x> is the reference waveform number.

COLOR<y> specifies the color to assign to the specified waveform, where <y> = 0 to 47.

Examples

DISplay:Ch1:INVERTColor COLOR23 sets the channel 2 Inverted waveform color to the value 23.

DISplay:REF3:INVERTColor? might return COLOR2, indicating that the Inverted waveform color setting for reference waveform 3 is 2.

DISplay:{CH<x>|Math<x>|REF<x>}:NORMALColor

This command sets or queries the normal mode color of the specified input source to the specified color. You can assign one of 48 unique colors to any channel, math, or reference waveform. These colors replace the default normal colors and remain in effect until you reset the colors.

Group

Display

Syntax

```
DISplay:{CH<x>|Math<x>|REF<x>}:NORMALColor COLOR<y>
DISplay:{CH<x>|Math<x>|REF<x>}:NORMALColor?
```

Arguments

CH<x> specifies the input channel for which you want to change the waveform color, where <x> is the channel number.

Math<x> specifies the math waveform for which you want to change the waveform color, where <x> is the math waveform number.

REF<x> specifies the reference waveform for which you want to change the waveform color, where <x> is the reference waveform number.

COLOR<y> specifies the color to assign to the specified waveform, where <y> = 0 to 47.

Examples

DISplay:Ch1:NORMALColor COLOR13 sets the channel 2 normal waveform color to the value 13.

DISplay:REF3:NORMALColor? might return COLOR12, indicating that the normal waveform color setting for reference waveform 3 is 12.

DISplay:COLors

Sets or queries the color mode for the graticule and waveform display.

Group Display Control

Syntax
```
DISplay:COLors {NORMal|INVERTed}
DISplay:COLors?
```

Arguments `NORMal` specifies normal color mode.

`INVERTed` specifies inverted color mode.

Examples `DISPLAY:COLORS NORMAL` sets the display colors to normal.

`DISPLAY:COLORS?` might return `DISPLAY:COLORS INVERTED` indicating the display colors are inverted.

DISplay:GLObal:B<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified bus. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified bus is added already.

Group Display Control

Syntax `DISplay:GLObal:B<x>:STATE {<NR1>|OFF|ON}`

Arguments `<NR1>` = 0 disables the display of the specified bus; any other value enables display of the bus.

`ON` enables display of the specified bus.

`OFF` disables display of the specified bus.

Examples `DISplay:GLObal:B1:STATE OFF` turns off the display of bus 1.

`DISplay:GLObal:B1:STATE?` might return `DISPLAY:GLOBAL:B1:STATE 1` indicating the bus is displayed.

DISplay:GLObal:CH<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified channel (both analog and digital). Setting this value true (On or NR1

≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified channel is added already.

Group Display Control

Syntax `DISplay:GLObal:CH<x>:STATE {<NR1>|OFF|ON}`

Arguments `<NR1>` = 0 disables the display of the specified channel; any other value enables display of the channel.

`ON` enables display of the specified channel.

`OFF` disables display of the specified channel.

Examples `DISplay:GLObal:CH1:STATE OFF` turns off the display of channel 1.

`DISplay:GLObal:CH1:STATE?` might return `:DISPLAY:GLOBAL:CH1:STATE 0` indicating that channel 1 is not displayed.

DISplay:GLObal:MATH<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified math. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified math waveform is added already.

Group Display Control

Syntax `DISplay:GLObal:MATH<x>:STATE {<NR1>|OFF|ON}`

Arguments `<NR1>` = 0 disables the display of the specified math; any other value enables display of the math.

`ON` enables display of the specified math.

`OFF` disables display of the specified math.

Examples `DISplay:GLObal:MATH1:STATE 1` displays math 1.

`DISplay:GLObal:MATH1:STATE?` might return `:DISPLAY:GLOBAL:MATH1:STATE 1` indicating that math1 is displayed.

DISplay:GLObal:PLOT<x>:STATE

This command sets or queries the global state (display mode On or Off) of the specified time trend plot. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified plot is added already.

Group Display Control

Syntax `DISplay:GLObal:PLOT<x>:STATE {<NR1>|OFF|ON}`

Arguments `PLOT<x>` is the plot number.

`<NR1>` = 0 disables the display of the specified plot; any other value enables display of the plot.

`ON` enables display of the specified plot.

`OFF` disables display of the specified plot.

Examples `DISplay:GLObal:PLOT1:STATE ON` displays plot 1.

`DISplay:GLObal:PLOT1:STATE?` might return `:DISPLAY:GLOBAL:PLOT1:STATE 1` indicating plot 1 is displayed.

DISplay:GLObal:REF<x>:STATE

this command sets or queries the global state (display mode On or Off) of the specified reference waveform. Setting this value true (On or NR1 ≠ 0) turns on the source in the waveform view. Setting this value false (Off or NR1 = 0) turns off the source in the waveform view. This command only works if the specified reference waveform is added already.

Group Display Control

Syntax `DISplay:GLObal:REF<x>:STATE {<NR1>|OFF|ON}`

Arguments `REF<x>` is the Reference waveform number.

`<NR1>` = 0 disables the display of the specified reference; any other value enables display of the reference.

`ON` enables display of the specified reference.

OFF disables display of the specified reference.

Examples DISplay:GLObal:REF1:STATE 1 displays reference 1.

DISplay:GLObal:REF1:STATE? might return :DISPLAY:GLOBAL:REF1:STATE 1 indicating that reference 1 is displayed.

DISplay:INTENSITy? (Query Only)

This query-only command returns the waveform saturation level and screen saver settings.

Group Display Control

Syntax DISplay:INTENSITy?

Related Commands

Arguments None

Examples DISPLAY:INTENSITY? might return :DISPLAY:INTENSITY:BACKLIGHT 1, indicating the intensity is set to 1.

DISplay:INTENSITy:BACKLight

This command sets or queries the display backlight intensity setting.

Group Display Control

Syntax
```
DISplay:INTENSITy:BACKLight {LOW|MEDium|HIGH}
DISplay:INTENSITy:BACKLight?
```

Arguments LOW selects a low brightness level.

MEDium selects a moderate brightness level.

HIGH selects a full brightness level.

Examples DISPLAY:INTENSITY:BACKLIGHT LOW sets the display backlight to low brightness level.

DISPLAY:INTENSITY:BACKLIGHT? might return DISPLAY:INTENSITY:BACKLIGHT HIGH, indicating that the display backlight is set to full brightness level.

DISplay:INTENSITy:BACKLight:AUTODim:ENAble

Sets or queries the state of the display auto-dim feature. The default is enabled. Once the backlight has dimmed, any button push, knob turn or mouse movement returns the backlight value to the value set by :DISplay:INTENSITy:BACKLight.

Group Display Control

Syntax

```
DISplay:INTENSITy:BACKLight:AUTODim:ENAble {ON|OFF}
DISplay:INTENSITy:BACKLight:AUTODim:ENAble?
```

Related Commands DISplay:INTENSITy:BACKLight

Arguments ON enables the display auto-dim feature.

OFF disables the display auto-dim feature.

Examples DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE ON enables auto-dimming of the backlight.

DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE? might return DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE ON, indicating that auto-dimming of the backlight is enabled.

DISplay:INTENSITy:BACKLight:AUTODim:TIMe

Sets or queries the amount of time, in minutes, to wait for no user interface activity before automatically dimming the display. The time can range from a minimum of 10 minutes to a maximum of 1440 minutes (24 hours). The default is 10 minutes.

Group Display Control

Syntax

```
DISplay:INTENSITy:BACKLight:AUTODim:TIMe <NR1>
DISplay:INTENSITy:BACKLight:AUTODim:TIMe?
```

Arguments <NR1> is the amount of time, in minutes, to wait for no user interface activity before automatically dimming the display.

Examples `DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME 30` sets the backlight to autodim in 30 minutes.

`DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME?` might return `DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME 60`, indicating the backlight will auto-dim in 60 minutes.

DISplay:MATHFFTView<n>:CURSor:ROLOCATION

This command sets or queries the location to display the specified Math FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).

Group Cursor

Syntax
```
DISplay:MATHFFTView<n>:CURSor:ROLOCATION {GRATICULE|BADGE}
DISplay:MATHFFTView<n>:CURSor:ROLOCATION?
```

Arguments `MATHFFTView<x>` is the Math FFT plot number.

`GRATICULE` sets the Math FFT plot cursor readouts to display as part of the cursors in the plot view.

`BADGE` removes the Math FFT plot cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.

Examples `DISplay:MATHFFTView4:CURSor:ROLOCATION GRATICULE` sets the cursor readout information for Math FFT plot 4 to display in the plot graticule.

`DISplay:MATHFFTView2:CURSor:ROLOCATION?` might return `DISplay:MATHFFTView2:CURSor:ROLOCATION BADGE`, indicating that cursor readout information for Math FFT plot 2 is set to display as a badge in the Results Bar.

DISplay:MATHFFTView<x>:AUTOScale

This command sets or returns the enabled state of autoscale for Math/FFT waveforms.

Group Display Control

Syntax
```
DISplay:MATHFFTView<x>:AUTOScale {OFF|ON|<NR1>
DISplay:MATHFFTView<x>:AUTOScale?
```

Arguments MATHFFTView<x> is the Math-FFT waveform number.

OFF disables the autoscale feature.

ON enables the autoscale feature.

<NR1> = 0 disables the autoscale feature; any other value enables the autoscale feature.

Examples DISplay:MATHFFTView1:AUTOScale OFF disables the autoscale feature.

DISplay:MATHFFTView3:AUTOScale? might return DISPLAY:MATHFFTVIEW3:AUTOSCALE 1, indicating that autoscale is on for the Math3 FFT waveform.

DISplay:MATHFFTView<x>:CURSor:ASOUrce? (Query Only)

This command queries the Math-FFT waveform view source for cursor A.

Group Cursor

Syntax DISplay:MATHFFTView<x>:CURSor:ASOUrce?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples DISplay:MATHFFTView1:CURSor:ASOUrce? might return DISPLAY:MATHFFTVIEW1:CURSOR:ASOURCE MATH1, indicating the source is MATH 1.

DISplay:MATHFFTView<x>:CURSor:BSOUrce? (Query Only)

This command queries the Math-FFT waveform view source for cursor B.

Group Cursor

Syntax DISplay:MATHFFTView<x>:CURSor:BSOUrce?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples `DISplay:MATHFFTView5:CURSor:BSOUrce?` might return `DISPLAY:MATHFFTVIEW5:CURSOR:BSOURCE MATH5`, indicating the source is Math waveform MATH 5.

DISplay:MATHFFTView<x>:CURSor:DDT? (Query Only)

This command queries the delta Y over delta X (ΔY/Δ X) cursor readout value of the specified cursor in the specified Math-FFT view.

NOTE. *If the current cursor is set to horizontal mode, this command will time out.*

Group Cursor

Syntax `DISplay:MATHFFTView<x>:CURSor:DDT?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

Examples `DISplay:MATHFFTView<x>:CURSor:DDT?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:DDT 14.0056E-9`, indicating the ΔY/Δ X cursor readout value is 14.0056E-9.

DISplay:MATHFFTView<x>:CURSor:FUNCtion

This command sets or queries the cursor type for the specified Math-FFT view.

Group Cursor

Syntax
```
DISplay:MATHFFTView<x>:CURSor:FUNCtion
{WAVEform|VBArs|HBArs|SCREEN}
DISplay:MATHFFTView<x>:CURSor:FUNCtion?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`WAVEFORM` specifies to display the paired vertical cursors in YT display format for measuring waveform amplitude and time. Measurements are taken at where the cursor intersects the waveform, and tracks waveform changes.

`VBArs` specifies vertical bar cursors, which measure in horizontal units.

`HBArs` specifies horizontal bar cursors, which measure in vertical units.

SCREEN specifies to display both horizontal and vertical bar cursors, which display the horizontal and vertical positions of the cursors, not waveform levels. Use these cursors to measure anywhere in the waveform display area.

Examples DISplay:MATHFFTView5:CURSor:FUNCtion HBARS sets Math-FFT waveform 5 to use horizontal bar cursors.

DISplay:MATHFFTView1:CURSor:FUNCtion? might return DISplay:MATHFFTView1:CURSor:FUNCtion? SCREEN, indicating that Math-FFT waveform 2 is set to use both horizontal and vertical cursors.

DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition

This command sets or returns the position of horizontal cursor A for the specified Math-FFT view.

Group Cursor

Syntax
```
DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:HBArs:APOSition?
```

Arguments MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the cursor position of the specified cursor in the specified view.

Examples DISplay:MATHFFTView1:CURSor:HBArs:APOSition 50.0e-3 sets the position to 50 mV.

DISplay:MATHFFTView4:CURSor:HBArs:APOSition? might return DISPLAY:MATHFFTVIEW4:CURSOR:HBARS:APOSITION –274.2247E-3, indicating the A cursor position is –274.2247E-3.

DISplay:MATHFFTView<x>:CURSor:HBArs:AUNIts? (Query Only)

This command queries the vertical units of horizontal cursor A for the specified Math-FFT view.

Group Cursor

Syntax DISplay:MATHFFTView<x>:CURSor:HBArs:AUNIts?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples DISplay:MATHFFTView1:CURSor:HBArs:AUNIts? might return DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:AUNITS "dBm", indicating the vertical units are dBm.

DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition

This command sets or returns the position of horizontal cursor B for the specified Math-FFT view.

Group Cursor

Syntax
```
DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:HBArs:BPOSition?
```

Arguments MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the vertical cursor B position for the specified Math-FFT view.

Examples DISplay:MATHFFTView1:CURSor:HBArs:BPOSition 50e-3 sets the position to 0.005 vertical units.

DISplay:MATHFFTView1:CURSor:HBArs:BPOSition? might return DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:BPOSITION –40.4061, indicating the position is –40.4061 vertical units.

DISplay:MATHFFTView<x>:CURSor:HBArs:BUNIts? (Query Only)

This command queries the vertical units of horizontal cursor B for the specified Math-FFT view.

Group Cursor

Syntax DISplay:MATHFFTView<x>:CURSor:HBArs:BUNIts?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples `DISplay:MATHFFTView3:CURSor:HBArs:BUNIts?` might return `DISPLAY:MATHFFTVIEW3:CURSOR:HBARS:BUNITS "dBm"`, indicating the cursor units are dBm.

DISplay:MATHFFTView<x>:CURSor:HBArs:DELTa? (Query Only)

This command queries the horizontal cursor's delta value of the specified Math-FFT view.

Group Cursor

Syntax `DISplay:MATHFFTView<x>:CURSor:HBArs:DELTa?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

Examples `DISplay:MATHFFTView1:CURSor:HBArs:DELTa?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:HBARS:DELTA 30.61448`, indicating the cursor readout is 30.61488 units.

DISplay:MATHFFTView<x>:CURSor:MODe

This command sets or queries the cursor tracking mode of the specified Math-FFT view.

Group Cursor

Syntax `DISplay:MATHFFTView<x>:CURSor:MODe {INDEPENDENT|TRACK}`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`INDEPENDENT` allows independent adjustment of the two cursors.

`TRACK` ties the navigational functionality of the two cursors together. For cursor 1 adjustments, this ties the movement of the two cursors together; however, cursor 2 continues to move independently of cursor 1.

Examples `DISplay:MATHFFTView1:CURSor:MODe INDEPENDENT` allows independent adjustment of the two cursors.

DISplay:MATHFFTView1:CURSor:MODe? might return :DISPLAY:MATHFFTVIEW1:CURSOR:MODE INDEPENDENT, indicating the mode is set to independent.

DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTATVALUE? (Query Only)

This command queries the one over delta cursor readout value of the specified Math-FFT view.

Group Cursor

Syntax DISplay:MATHFFTView<x>:CURSor:ONEOVERDELTATVALUE?

Arguments MATHFFTView<x> is the Math-FFT waveform number.

Examples DISplay:MATHFFTView1:CURSor:ONEOVERDELTATVALUE? might return DISPLAY:MATHFFTVIEW1:CURSOR:ONEOVERDELTATVALUE 2.6005E-9, indicating the one over delta time value is 2.6005 nanoseconds.

DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition

This command sets or returns the vertical cursor A x-axis waveform measurement position of the specified Math-FFT view.

Group Cursor

Syntax
```
DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:SCREEN:AXPOSition?
```

Arguments MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the cursor position in MHz.

Examples DISPLAY:MATHFFTVIEW1:CURSOR:SCREEN:AXPOSITION 700.000E+6 sets the cursor position to 700 MHz.

DISPLAY:MATHFFTVIEW3:CURSOR:SCREEN:AXPOSITION? might return DISPLAY:MATHFFTVIEW3:CURSOR:SCREEN:AXPOSITION 125.000E+6, indicating the x-axis value of cursor A is 125 MHz.

DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOSition

This command sets or returns the vertical cursor A y-axis amplitude measurement value of the specified Math-FFT view.

Group Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:SCREEN:AYPOSition?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the cursor A position of the specified cursor in the specified view.

Examples `DISplay:MATHFFTView2:CURSor:SCREEN:AYPOSition 1.0e0` sets the cursor position to 1.0 dBm.

`DISplay:MATHFFTView5:CURSor:SCREEN:AYPOSition?` might return `DISPLAY:MATHFFTVIEW5:CURSOR:SCREEN:AYPOSITION 1.1741714106633`, indicating the cursor position is 1.174 vertical units (such as dBm).

DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOSition

This command sets or returns the vertical cursor Bx-axis waveform time measurement position of the specified Math-FFT view.

Group Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:SCREEN:BXPOSition?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the horizontal cursor B position of the specified cursor in the specified view.

Examples `DISplay:MATHFFTView1:CURSor:SCREEN:BXPOSition 3.5e9` sets the cursor position to 3.5 GHz.

DISplay:MATHFFTView1:CURSor:SCREEN:BXPOSition? might return DISPLAY:MATHFFTVIEW1:CURSOR:SCREEN:BXPOSITION 2.50E+9, indicating the cursor position is 2.5 GHz.

DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOSition

This command sets or returns the vertical cursor B y-axis amplitude measurement value of the specified Math-FFT view.

Group Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:SCREEN:BYPOSition?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

<NR3> is the vertical cursor B position of the specified cursor in the specified view.

Examples

DISplay:MATHFFTView1:CURSor:SCREEN:BYPOSition -50.0e0 sets the cursor position to -50.0.

DISplay:MATHFFTView4:CURSor:SCREEN:BYPOSition? might return DISPLAY:MATHFFTVIEW4:CURSOR:SCREEN:BYPOSITION -59.0460929065173, indicating the cursor position is at -59.0 dBm.

DISplay:MATHFFTView<x>:CURSor:STATE

This command sets or queries the visible state of cursors for the specified Math-FFT view.

Group Cursor

Syntax

```
DISplay:MATHFFTView<x>:CURSor:STATE {ON|OFF|<NR1>}
DISplay:MATHFFTView<x>:CURSor:STATE?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

1 or ON enables the cursors.

0 or OFF disables the cursors.

<NR1> = 0 turns off cursors; any other value displays cursors.

Examples `DISplay:MATHFFTView2:CURSor:STATE OFF` turns off the cursors for Math-FFT waveform 2.

`DISplay:MATHFFTView1:CURSor:STATE?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:STATE 1` indicating the cursors are enabled.

DISplay:MATHFFTView<x>:CURSor:VBArs:APOSition

This command sets or queries the horizontal cursor A position for the specified Math-FFT view.

Group Cursor

Syntax
```
DISplay:MATHFFTView<x>:CURSor:VBArs:APOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:VBArs:APOSition?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` sets the vertical cursor A position in the specified view.

Examples `DISplay:MATHFFTView1:CURSor:VBArs:APOSition 500.0e6` sets the cursor A position to 500 MHz.

`DISplay:MATHFFTView1:CURSor:VBArs:APOSition?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:APOSITION 585.533869115958E+6`, indicating cursor position is 585.5 MHz.

DISplay:MATHFFTView<x>:CURSor:VBArs:AUNIts? (Query Only)

This command queries the vertical cursor A measurement units for the specified Math-FFT view.

Group Cursor

Syntax
```
DISplay:MATHFFTView<x>:CURSor:VBArs:AUNIts?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

Examples `DISplay:MATHFFTView1:CURSor:VBArs:AUNIts?` might return `:DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:UNITS "dBm"`, indicating the cursor units are dBm.

DISplay:MATHFFTView<x>:CURSor:VBArs:BPOSition

This command sets or queries the vertical cursor B position for the specified Math-FFT view.

Group Cursor

Syntax
```
DISplay:MATHFFTView<x>:CURSor:VBArs:BPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:VBArs:BPOSition?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` sets the vertical cursor B position in the specified view.

Examples `DISplay:MATHFFTView1:CURSor:VBArs:BPOSition 2.0e9` sets the cursor B position to 2.0 GHz.

`DISplay:MATHFFTView1:CURSor:VBArs:BPOSition?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:BPOSITION 2.50E+9`, indicating the cursor position is at 2.5 GHz.

DISplay:MATHFFTView<x>:CURSor:VBArs:BUNIts? (Query Only)

This command queries the vertical cursor B measurement units for the specified Math-FFT view.

Group Cursor

Syntax
```
DISplay:MATHFFTView<x>:CURSor:VBArs:BUNIts?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

Examples `DISplay:MATHFFTView1:CURSor:VBArs:BUNIts?` might return `:DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:BUNITS "Hz"` indicating the cursor units are Hz.

DISplay:MATHFFTView<x>:CURSor:VBArs:DELTa? (Query Only)

This command queries the vertical cursor's delta T readout value for the specified Math-FFT view.

Group Cursor

Syntax `DISplay:MATHFFTView<x>:CURSor:VBArs:DELTa?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

Examples `DISplay:MATHFFTView1:CURSor:VBArs:DELTa?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:VBARS:DELTA 1.9144661308840E+9`, indicating the delta T cursor value is 1.9 GHz.

DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition

This command sets or queries the waveform cursor A position for the specified Math-FFT view.

Group Cursor

Syntax
```
DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:WAVEform:APOSition?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the waveform cursor A position in the specified plot view.

Examples `DISplay:MATHFFTView1:CURSor:WAVEform:APOSition 600.0e6` sets the position to 600 MHz.

`DISplay:MATHFFTView1:CURSor:WAVEform:APOSition?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:WAVEFORM:APOSITION 25.000E+6`, indicating the cursor position is 25 MHz.

DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition

This command sets or queries the waveform cursor B position for the specified Math-FFT view.

Group Cursor

Syntax
```
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition <NR3>
DISplay:MATHFFTView<x>:CURSor:WAVEform:BPOSition?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the waveform cursor B position in the specified plot view.

Examples `DISplay:MATHFFTView1:CURSor:WAVEform:BPOSition 2.0E+9` sets the cursor position to 2.0 GHz.

`DISplay:MATHFFTView1:CURSor:WAVEform:BPOSition?` might return `DISPLAY:MATHFFTVIEW1:CURSOR:WAVEFORM:BPOSITION 250.0E+6`, indicating the waveform cursor B position is 250 MHz.

DISplay:MATHFFTView<x>:GRIDlines

This command sets or queries the grid lines setting for the specified Math-FFT view.

Group Display Control

Syntax
```
DISplay:MATHFFTView<x>:GRIDlines {HORizontal|VERTical|BOTH}
DISplay:MATHFFTView<x>:GRIDlines?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`HORizontal` specifies horizontal grid lines.

`VERTical` specifies vertical grid lines.

`BOTH` specifies both vertical and horizontal grid lines.

Examples `DISplay:MATHFFTView1:GRIDlines VERTICAL` specifies vertical grid lines.

`DISplay:MATHFFTView1:GRIDlines?` might return `:DISPLAY:MATHFFTVIEW1:GRIDLINES BOTH` indicating both horizontal and vertical grid lines are displayed.

DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE

This command sets or queries the display state of the specified math waveform for the specified Math-FFT view.

Group Display Control

Syntax

```
DISplay:MATHFFTView<x>:MATH:MATH<x>:STATE {OFF|ON|<NR1>}
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

OFF disables displaying the specified Math-FFT view.

ON enables displaying the specified Math-FFT view.

<NR1> = 0 disables the specified Math-FFT view; any other value enables the specified Math-FFT view.

Examples

DISplay:MATHFFTView1:MATH:MATH1:STATE ON enables the specified Math-FFT view.

DISplay:MATHFFTView4:MATH:MATH4:STATE? might return :DISPLAY:MATHFFTVIEW1:MATH:MATH1:STATE 1, indicating the waveform Math-FFT 1 is displayed.

DISplay:MATHFFTView<x>:XAXIS:SCALE

This command sets or queries the x-axis scale (Linear or Log) for the specified Math-FFT view.

Group Display Control

Syntax

```
DISplay:MATHFFTView<x>:XAXIS:SCALE {LINEAr|LOG}
DISplay:MATHFFTView<x>:XAXIS:SCALE?
```

Arguments

MATHFFTView<x> is the Math-FFT waveform number.

LINEAr specifies a linear scale.

LOG specifies a logarithmic scale.

Examples `DISplay:MATHFFTView1:XAXIS:SCALE LOG` specifies a logarithmic scale.

`DISplay:MATHFFTView1:XAXIS:SCALE?` might return `DISPLAY:MATHFFTVIEW1:XAXIS:SCALE LINEAR`, indicating that the scale setting is linear.

DISplay:MATHFFTView<x>:YAXIS:SCALE

This command sets or queries the vertical scale setting (Linear or dBm) for the specified Math-FFT view.

Group Display Control

Syntax
```
DISplay:MATHFFTView<x>:YAXIS:SCALE {LINEAr|DBM}
DISplay:MATHFFTView<x>:YAXIS:SCALE?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`LINEAr` specifies a linear scale.

`DBM` specifies a dBm scale.

Examples `DISplay:MATHFFTView1:YAXIS:SCALE LINEAR` specifies a linear scale.

`DISplay:MATHFFTView1:YAXIS:SCALE?` might return `DISPLAY:MATHFFTVIEW1:YAXIS:SCALE DBM`, indicating that the scale is dBm.

DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM

This command sets or queries the value of the left edge of the zoom area for the specified Math-FFT view.

Group Zoom

Syntax
```
DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM <NR3>
DISplay:MATHFFTView<x>:ZOOM:XAXIS:FROM?
```

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the value of the left edge of the zoom x axis in the specified plot view.

Examples `DISplay:MATHFFTView1:ZOOM:XAXIS:FROM 800.e6` sets the x axis value of the left edge of the plot view to 800 MHz.

`DISplay:MATHFFTView1:ZOOM:XAXIS:FROM?` might return `DISPLAY:MATHFFTVIEW1:ZOOM:XAXIS:FROM 781.0E+6`, indicating the x axis value of the left edge of the plot view is at 781 MHz.

DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO

This command sets or queries the value of the right edge value of the zoom area for the specified Math-FFT view.

Group Zoom

Syntax `DISplay:MATHFFTView<x>:ZOOM:XAXIS:TO <NR3>`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the value of the right edge of the zoom x axis in the specified plot view.

Examples `DISplay:MATHFFTView1:ZOOM:XAXIS:TO 2.3E+9` sets the value of the right edge of the x axis to 2.3 GHz.

`DISplay:MATHFFTView1:ZOOM:XAXIS:TO?` might return `:DISPLAY:MATHFFTVIEW1:ZOOM:XAXIS:TO 2.343750E+9`, indicating the value of the right edge of the zoom x axis is 2.34 GHz.

DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM

This command sets or queries the bottom edge value of the zoom y-axis area for the specified Math-FFT view.

Group Zoom

Syntax `DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM <NR3>`
`DISplay:MATHFFTView<x>:ZOOM:YAXIS:FROM?`

Arguments `MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the bottom value of the zoom y axis in the specified plot view.

Examples

`DISplay:MATHFFTView1:ZOOM:YAXIS:FROM -75.0e0` sets the bottom value of the y axis to -75 dBm.

`DISplay:MATHFFTView1:ZOOM:YAXIS:FROM?` might return `DISPLAY:MATHFFTVIEW1:ZOOM:YAXIS:FROM -78.8258285893367` indicating the bottom value of the y axis is -78.8 dBm.

DISplay:MATHFFTView<x>:ZOOM:YAXIS:TO

This command sets or queries the top edge value of the zoom y-axis area for the specified Math-FFT view.

Group

Zoom

Syntax

```
DISplay:MATHFFTView<x>:ZOOM:YAXIS:TO <NR3>
DISplay:MATHFFTView<x>:ZOOM:YAXIS:TO?
```

Arguments

`MATHFFTView<x>` is the Math-FFT waveform number.

`<NR3>` is the top value of the zoom y axis in the specified plot view.

Examples

`DISplay:MATHFFTView1:ZOOM:YAXIS:TO 21.0E0` sets the top value of the zoom y axis in the specified plot view to 21 dBm.

`DISplay:MATHFFTView1:ZOOM:YAXIS:TO?` might return `DISPLAY:MATHFFTVIEW1:ZOOM:YAXIS:TO 21.1741714106633`, indicating the top value of the zoom y axis in the specified plot view is 21.17 dBm.

DISplay:PERSistence

This command sets or queries the display persistence for analog waveforms. Persistence is valid for wave views only.

Group

Display Control

Syntax

```
DISplay:PERSistence
{OFF|AUTO|INFPersist|INFInite|VARpersist|CLEAR}
DISplay:PERSistence?
```

Related Commands

DISplay:VARpersist

Arguments

`OFF` disables the persistence aspect of the display.

`AUTO` automatically set the persistence.

`INFPersist` sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.

`INFInite` sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.

`VARPersist` sets a display mode where set pixels are gradually dimmed.

`CLEAR` resets the persist time count down and clears the display of acquired points.

Examples

`DISPLAY:PERSISTENCE VARPERSIST` sets the persistence aspect of the display to fade set pixels according to the time set in the DISplay:VARpersist command.

`DISPLAY:PERSISTENCE?` might return `:DISPLAY:PERSISTENCE OFF`, indicating that the persistence aspect of the display is disabled.

DISplay:PERSistence:RESET (No Query Form)

This command controls the clearing of persistence data that has been built up over time. Persistence is valid for wave views only.

Group

Display Control

Syntax

`DISplay:PERSistence:RESET`

Related Commands

DISplay:PERSistence

DISplay:VARpersist

Examples

`DISPLAY:PERSISTENCE:RESET` clears the display of persistence data.

DISplay:PLOTView<n>:CURSor:ROLOCATION

This command sets or queries the location to display the specified plot cursor readouts (in the plot graticule or in a badge in the Results Bar).

Group

Cursor

Syntax

```
DISplay:PLOTView<n>:CURSor:ROLOCATION {GRATICULE|BADGE}
DISplay:PLOTView<n>:CURSor:ROLOCATION?
```

Arguments

PLOTView<x> is the waveform plot number.

GRATICULE sets the plot cursor readouts to display as part of the cursors in the plot view.

BADGE removes the plot cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.

Examples

DISplay:PLOTView1:CURSor:ROLOCATION GRATICULE sets the cursor readout information for plot 1to display in the plot graticule.

DISplay:PLOTView2:CURSor:ROLOCATION? might return DISplay:PLOTView2:CURSor:ROLOCATION BADGE, indicating that cursor readout information for plot 2 is set to display as a badge in the Results Bar.

DISplay:PLOTView<x>:AUTOScale

This command sets or queries the enabled state of autoscale for the specified plot.

Group

Display Control

Syntax

```
DISplay:PLOTView<x>:AUTOScale {OFF|ON|<NR1>}
DISplay:PLOTView<x>:AUTOScale?
```

Arguments

PLOTView<x> is the Plot waveform number.

OFF disables the autoscale feature.

ON enables the autoscale feature.

<NR1> = 0 disables the autoscale feature; any other value enables the autoscale feature.

Examples

DISplay:PLOTView1:AUTOScale OFF turns off plot autoscale.

DISplay:PLOTView3:AUTOScale? might return :DISPLAY:PLOTVIEW3:AUTOSCALE 0, indicating the plot autoscale is off for plot 3.

DISplay:PLOTView<x>:CURSor:ASOUrce? (Query Only)

This command queries the cursor source for plot cursor A.

Group Cursor

Syntax `DISplay:PLOTView<x>:CURSor:ASOUrce?`

Arguments `PLOTView<x>` is the Plot waveform number.

Examples `DISplay:PLOTView1:CURSor:ASOUrce?` might return `DISPLAY:PLOTVIEW1:CURSOR:ASOURCE PLOT1` indicating the source of the cursor is plot 1.

DISplay:PLOTView<x>:CURSor:BSOUrce? (Query Only)

This command queries the cursor source for plot cursor B.

Group Cursor

Syntax `DISplay:PLOTView<x>:CURSor:BSOUrce?`

Arguments `PLOTView<x>` is the Plot waveform number.

Examples `DISplay:PLOTView1:CURSor:BSOUrce?` might return `:DISPLAY:PLOTVIEW1:CURSOR:BSOURCE PLOT1` indicating the B cursor source is plot 1.

DISplay:PLOTView<x>:CURSor:DDT? (Query Only)

This command returns the delta V over delta T cursor readout value for the specified Plot view.

Group Cursor

Syntax `DISplay:PLOTView<x>:CURSor:DDT?`

Arguments PlotView<x> is the Plot waveform number.

Examples DISplay:PLOTView1:CURSor:DDT? might return :DISPLAY:PLOTVIEW1:CURSOR:DDT 131.0470E+6 indicating the cursor readout value is 131.0470E+6.

DISplay:PLOTView<x>:CURSor:FUNCtion

This command sets or queries the cursor mode for the specified Plot view.

Group Cursor

Syntax
```
DISplay:PLOTView<x>:CURSor:FUNCtion
{WAVEFORM|VBArs|HBArs|SCREEN}
DISplay:PLOTView<x>:CURSor:FUNCtion?
```

Arguments PLOTView<x> is the Plot waveform number.

WAVEFORM specifies to display the paired cursors in YT display format for measuring waveform amplitude and time.

VBArs specifies vertical bar cursors, which measure in horizontal units.

HBArs specifies horizontal bar cursors, which measure in vertical units.

SCREEN specifies to display both horizontal and vertical bar cursors, which measure the selected waveform in horizontal and vertical units. Use these cursors to measure anywhere in the waveform display area.

Examples DISplay:PLOTView1:CURSor:FUNCtion VBARS displays vertical bar cursors.

DISplay:PLOTView1:CURSor:FUNCtion? might return :DISPLAY:PLOTVIEW1:CURSOR:FUNCTION SCREEN indicating that both horizontal and vertical bar cursors are displayed.

DISplay:PLOTView<x>:CURSor:HBArs:APOSition

This command sets or queries the horizontal cursor A position for the specified Plot view.

Group Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:HBArs:APOSition <NR3>
DISplay:PLOTView<x>:CURSor:HBArs:APOSition?
```

Arguments

`PLOTView<x>` is the Plot waveform number.

`<NR3>` is the cursor position.

Examples

`DISplay:PLOTView<x>:CURSor:HBArs:APOSition 2.0` sets the cursor to 2 Volts.

`DISplay:PLOTView<x>:CURSor:HBArs:APOSition?` might return `:DISPLAY:PLOTVIEW1:CURSOR:HBARS:APOSITION 2.2186` indicating the HBArs cursors are at 2.2186 Volts.

DISplay:PLOTView<x>:CURSor:HBArs:AUNIts? (Query Only)

This command queries the horizontal cursor A vertical units for the specified Plot view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:HBArs:AUNIts?
```

Returns

`PLOTView<x>` is the Plot waveform number.

Examples

`DISplay:PLOTView1:CURSor:HBArs:AUNIts?` might return `:DISPLAY:PLOTVIEW1:CURSOR:HBARS:AUNITS "V"`, indicating the A units are volts.

DISplay:PLOTView<x>:CURSor:HBArs:BPOSition

This command sets or queries the horizontal cursor B position for the specified Plot view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:HBArs:BPOSition <NR3>
DISplay:PLOTView<x>:CURSor:HBArs:BPOSition?
```

Arguments PLOTView<x> is the Plot waveform number.

<NR3> is the HBArs vertical position.

Examples DISplay:PLOTView<x>:CURSor:HBArs:BPOSition 2.0 sets the cursor to 2 Volts.

DISplay:PLOTView<x>:CURSor:HBArs:BPOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:HBARS:BPOSITION 2.2186 indicating the HBArs cursors are at 2.2186 Volts.

DISplay:PLOTView<x>:CURSor:HBArs:BUNIts? (Query Only)

This command queries the cursor B vertical units for the specified Plot view.

Group Cursor

Syntax DISplay:PLOTView<x>:CURSor:HBArs:BUNIts??

Arguments PLOTView<x> is the Plot waveform number.

Examples DISplay:PLOTView1:CURSor:HBArs:BUNIts? might return :DISPLAY:PLOTVIEW1:CURSOR:HBARS:BUNITS "V" indicating the units are Volts.

DISplay:PLOTView<x>:CURSor:HBArs:DELTa? (Query Only)

This command queries the delta V cursor readout value for the specified Plot view.

Group Cursor

Syntax DISplay:PLOTView<x>:CURSor:HBArs:DELTa??

Arguments PLOTView<x> is the Plot waveform number.

Returns Returns an <NR3> that is the delta V cursor value.

Examples `DISplay:PLOTView1:CURSor:HBArs:DELTa?` might return `:DISPLAY:PLOTVIEW1:CURSOR:HBARS:DELTA 1.2543`, indicating the delta value is 1.2543 V.

DISplay:PLOTView<x>:CURSor:MODe

This command sets or queries the cursor tracking mode for the specified Plot view.

Group Cursor

Syntax
```
DISplay:PLOTView<x>:CURSor:MODe {INDEPENDENT|TRACK}
DISplay:PLOTView<x>:CURSor:MODe?
```

Arguments `PLOTView<x>` is the Plot waveform number.

`INDEPENDENT` allows independent adjustment of the two cursors.

`TRACK` ties the navigational functionality of the two cursors together. For cursor A adjustments, this ties the movement of the two cursors together; however, cursor B continues to move independently of cursor A.

Examples `DISplay:PLOTView1:CURSor:MODe TRACK` sets the cursor to track together.

`DISplay:PLOTView1:CURSor:MODe?` might return `:DISPLAY:PLOTVIEW1:CURSOR:MODE INDEPENDENT`, indicating independent adjustment of the cursors is allowed.

DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE? (Query Only)

This command sets or queries the one over delta T cursor readout value for the specified Plot view.

Group Cursor

Syntax `DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE?`

Arguments `PLOTView<x>` is the Plot waveform number.

Returns Returns an `<NR3>` that is the one over delta T cursor readout value (it may not be delta T, depending on the plot units).

Examples `DISplay:PLOTView<x>:CURSor:ONEOVERDELTATVALUE?` might return `:DISPLAY:PLOTVIEW1:CURSOR:ONEOVERDELTATVALUE 88.2295E+6` indicating the 1 over delta time value is 88.2295 MHz.

DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:AXPOSition?
```

Arguments `PLOTView<x>` is the Plot waveform number.

`<NR3>` is the horizontal cursor A position.

Examples `DISplay:PLOTView1:CURSor:SCREEN:AXPOSition -5.0` sets the A X cursor position to —5.0 ns.

`DISplay:PLOTView1:CURSor:SCREEN:AXPOSition?` might return `:DISPLAY:PLOTVIEW1:CURSOR:SCREEN:AXPOSITION -5.5460E-9` indication the cursor A X position is -5.546 ns.

DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:AYPOSition?
```

Arguments `PLOTView<x>` is the Plot waveform number.

`<NR3>` is the vertical cursor A position.

Examples `DISplay:PLOTView1:CURSor:SCREEN:AYPOSition 2.0` set the A cursor Y position to 2.0 Volts.

DISplay:PLOTView1:CURSor:SCREEN:AYPOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:SCREEN:AYPOSITION 1.9035 indicating the A cursor Y position is 1.9035 V.

DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

Group Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:BXPOSition?
```

Arguments PLOTView<x> is the Plot waveform number.

<NR3> is the horizontal cursor B position.

Examples DISplay:PLOTView1:CURSor:SCREEN:BXPOSition 59.0E-9 sets the B cursor X position to 59.0 ns.

DISplay:PLOTView1:CURSor:SCREEN:BXPOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:SCREEN:BXPOSITION 59.6802E-9 indicating the B cursor X position is 59.6802 ns.

DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition <NR3>
DISplay:PLOTView<x>:CURSor:SCREEN:BYPOSition?
```

Arguments PLOTView<x> is the Plot waveform number.

<NR3> is the vertical cursor B position.

Examples DISplay:PLOTView1:CURSor:SCREEN:BYPOSition 0.589 sets the B cursor Y position to 589 mV.

DISplay:PLOTView1:CURSor:SCREEN:BYPOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:SCREEN:BYPOSITION 589.0696E-3 indicating the B cursor Y position is 589.0696 mV.

DISplay:PLOTView<x>:CURSor:SPLITMODE

This command sets or queries the cursor source mode in the specified view.

Group Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:SPLITMODE {SAME|SPLIT}
DISplay:PLOTView<x>:CURSor:SPLITMODE?
```

Arguments

PLOTView<x> is the Plot waveform number.

SAME specifies that both cursors are on the same waveform.

SPLIT specifies that the cursors can be on different waveforms.

Examples

DISplay:PLOTView1:CURSor:SPLITMODE SAME sets the cursors to be on the same waveform.

DISplay:PLOTView1:CURSor:SPLITMODE? might return :DISPLAY:PLOTVIEW1:CURSOR:SPLITMODE SAME indicating both cursors are on the same waveform.

DISplay:PLOTView<x>:CURSor:STATE

This command sets or queries the visible state of the cursor of the specified cursor in the specified view.

Group Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:STATE {OFF|ON|<NR1>}
DISplay:PLOTView<x>:CURSor:STATE?
```

Arguments

PLOTView<x> is the Plot waveform number.

OFF disables the specified cursor.

ON enables the specified cursor.

<NR1> = 0 disables the specified cursor; any other value enables the specified cursor.

Examples

DISplay:PLOTView1:CURSor:STATE OFF disables the specified cursor.

DISplay:PLOTView5:CURSor:STATE? might return :DISPLAY:PLOTVIEW5:CURSOR:STATE 1 indicating cursors are on.

DISplay:PLOTView<x>:CURSor:VBArs:APOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:VBArs:APOSition <NR3>
DISplay:PLOTView<x>:CURSor:VBArs:APOSition?
```

Arguments

PLOTView<x> is the Plot waveform number.

<NR3> is the vertical cursor A position.

Examples

DISplay:PLOTView1:CURSor:VBArs:APOSition 50.0E-9 sets the VBARS APOSITION to 50 ns.

DISplay:PLOTView1:CURSor:VBArs:APOSition? might return :DISPLAY:PLOTVIEW1:CURSOR:VBARS:APOSITION -60.0796E-9 indicating the VBARS APOSITION is set to -60.08 ns.

DISplay:PLOTView<x>:CURSor:VBArs:BPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group

Cursor

Syntax

```
DISplay:PLOTView<x>:CURSor:VBArs:BPOSition <NR3>
DISplay:PLOTView<x>:CURSor:VBArs:BPOSition?
```

Arguments PLOTView<x> is the Plot waveform number.

<NR3> is the vertical cursor B position.

Examples :DISPLAY:PLOTVIEW1:CURSOR:VBARS:BPOSITION 50.0E-9 sets the VBARS BPOSITION to 50 ns.

:DISPLAY:PLOTVIEW1:CURSOR:VBARS:BPOSITION? might return :DISPLAY:PLOTVIEW1:CURSOR:VBARS:BPOSITION -60.0796E-9 indicating the VBARS BPOSITION is set to 60.08 ns.

DISplay:PLOTView<x>:CURSor:VBArs:DELTa? (Query Only)

This command queries the delta T cursor readout value of the specified cursor in the specified view.

Group Cursor

Syntax DISplay:PLOTView<x>:CURSor:VBArs:DELTa?

Arguments PLOTView<x> is the Plot waveform number.

Returns The delta T cursor readout value is returned.

Examples DISplay:PLOTView1:CURSor:VBArs:DELTa? might return :DISPLAY:PLOTVIEW1:CURSOR:VBARS:DELTA 119.7593E-9 indicating the VBARS DELTA is set to 119.76 ns.

DISplay:PLOTView<x>:CURSor:VBArs:UNIts? (Query Only)

This command queries the VBArs cursor readout units of the specified cursor in the specified view.

Group Cursor

Syntax DISplay:PLOTView<x>:CURSor:VBArs:UNIts?

Arguments PLOTView<x> is the Plot waveform number.

Returns The VBArs cursor readout units are returned.

Examples `DISplay:PLOTView<x>:CURSor:VBArs:UNIts?` might return `:DISPLAY:PLOTVIEW5:CURSOR:VBARS:UNITS "V"` indicating the units are Volts.

DISplay:PLOTView<x>:CURSor:WAVEform:APOSition

This command sets or queries the waveform cursor A horizontal position of the specified cursor in the specified view.

NOTE. *In case of XY plot, this command has no effect when used to set the value. In case of bathtub plot, this command sets or returns the cursor A vertical position. For all other plots, this command sets or returns the cursor A horizontal position.*

Group Cursor

Syntax
```
DISplay:PLOTView<x>:CURSor:WAVEform:APOSition <NR3>
DISplay:PLOTView<x>:CURSor:WAVEform:APOSition?
```

Arguments `PLOTView<x>` is the Plot waveform number.

`<NR3>` is the horizontal cursor A position.

Examples `DISplay:PLOTView1:CURSor:WAVEform:APOSition -50.0E-9` sets the waveform `APOSITION` to -50.0 ns.

`DISplay:PLOTView1:CURSor:WAVEform:APOSition?` might return `:DISPLAY:PLOTVIEW1:CURSOR:WAVEFORM:APOSITION -60.0796E-9` indicating the waveform `APOSITION` is set to -60.08 ns.

DISplay:PLOTView<x>:CURSor:WAVEform:BPOSition

This command sets or queries the waveform cursor B horizontal position of the specified cursor in the specified view.

NOTE. *In case of XY plot, this command has no effect when used to set the value. In case of bathtub plot, this command has no effect. Query returns invalid values. For all other plots, this command sets or returns the cursor B horizontal position.*

Group Cursor

Syntax
```
DISplay:PLOTView<x>:CURSor:WAVEform:BPOSition <NR3>
DISplay:PLOTView<x>:CURSor:WAVEform:BPOSition?
```

Arguments PLOTView<x> is the Plot waveform number.

<NR3> is the horizontal cursor B position.

Examples DISplay:PLOTView1:CURSor:WAVEform:BPOSition 3.0 sets the position to 3.0 V.

DISplay:PLOTView5:CURSor:WAVEform:BPOSition? might return :DISPLAY:PLOTVIEW5:CURSOR:WAVEFORM:BPOSITION 2.600517840 indicating the position is 2.5 V.

DISplay:PLOTView<x>:GRIDlines

This command sets or queries the Grid (graticule) lines setting of the specified plot. This command works for plots that have vertical and horizontal units associated with the graticule. For example, this command does not work for XY or XYZ plots.

Group Display Control

Syntax
```
DISplay:PLOTView<x>:GRIDlines {HORizontal|VERTical|BOTH}
DISplay:PLOTView<x>:GRIDlines?
```

Arguments PLOTView<x> is the Plot waveform number.

HORizontal specifies horizontal grid lines.

VERTical specifies vertical grid lines.

BOTH specifies both vertical and horizontal grid lines.

Examples DISplay:PLOTView1:GRIDlines Vertical sets the vertical grid lines to display.

DISplay:PLOTView1:GRIDlines? might return :DISPLAY:PLOTVIEW1:GRIDLINES BOTH indicating that both vertical and horizontal grid lines are displayed.

DISPLAY:PLOTVIEW<x>:XAXIS:SCALE

This command sets or queries the horizontal scale setting for applicable plots (Linear or Log) for the specified plot view.

Group Display

Syntax `DISPLAY:PLOTVIEW<x>:XAXIS:SCALE {LINEAR|LOG}`

Arguments `PLOTView<x>` is the Plot waveform number.

`LINEAR` creates a plot with linear scales.

`LOG` creates a plot with logarithmic scales.

Examples `DISPLAY:PLOTVIEW1:XAXIS:SCALE LOG` creates a plot with a logarithmic horizontal scale.

`DISPLAY:PLOTVIEW1:XAXIS:SCALE?` might return `:DISPLAY:PLOTVIEW1:XAXIS:SCALE LINEAR` indicating plots will have linear scales.

DISPLAY:PLOTVIEW<x>:YAXIS:SCALE

This command sets or queries the vertical scale setting for applicable plots (Linear or Log) in the specified plot view.

Group Display

Syntax `DISPLAY:PLOTVIEW<x>:YAXIS:SCALE {LINEAR|LOG}`

Arguments `PLOTView<x>` is the Plot waveform number.

`LINEAR` specifies a linear vertical scale.

`LOG` specifies a logarithmic vertical scale.

Examples `DISPLAY:PLOTVIEW1:YAXIS:SCALE LINEAR` sets the plot to use a linear vertical scale.

`DISPLAY:PLOTVIEW1:YAXIS:SCALE?` might return `:DISPLAY:PLOTVIEW1:YAXIS:SCALE LOG` indicating a logarithmic vertical scale is used.

DISplay:PLOTView<x>:ZOOM:XAXIS:FROM

This command sets or queries the value of the left edge of the specified plot.

Group Zoom

Syntax
```
DISplay:PLOTView<x>:ZOOM:XAXIS:FROM <NR3>
DISplay:PLOTView<x>:ZOOM:XAXIS:FROM?
```

Arguments `PLOTView<x>` is the Plot waveform number.

`<NR3>` is start of the zoom x-axis.

Examples `DISplay:PLOTView1:ZOOM:XAXIS:FROM 2.0` sets the start of the zoom x-axis to 2.0 V.

`DISplay:PLOTView3:ZOOM:XAXIS:FROM?` might return
`DISPLAY:PLOTVIEW3:ZOOM:XAXIS:FROM 1.9844803459459`
indicating the start of the zoom x-axis is 1.9845 V.

DISplay:PLOTView<x>:ZOOM:XAXIS:TO

This command sets or queries the value of the right edge of the specified plot.

Group Zoom

Syntax
```
DISplay:PLOTView<x>:ZOOM:XAXIS:TO <NR3>
DISplay:PLOTView<x>:ZOOM:XAXIS:TO?
```

Arguments `PLOTView<x>` is the Plot waveform number.

`<NR3>` is the end of the zoom x-axis.

Examples `DISplay:PLOTView1:ZOOM:XAXIS:TO 2.5` sets the end of the zoom x-axis to 2.5 V.

`DISplay:PLOTView3:ZOOM:XAXIS:TO?` might return
`DISPLAY:PLOTVIEW3:ZOOM:XAXIS:TO 2.2144883507508`
indicating the end of the zoom x-axis is 2.21 V.

DISplay:PLOTView<x>:ZOOM:YAXIS:FROM

This command sets or queries the bottom value of the zoom y-axis in the specified plot view.

Group Zoom

Syntax

```
DISplay:PLOTView<x>:ZOOM:YAXIS:FROM <NR3>
DISplay:PLOTView<x>:ZOOM:YAXIS:FROM?
```

Arguments PLOTView<x> is the Plot waveform number.

<NR3> is the bottom value of the zoom y-axis.

Examples DISplay:PLOTView1:ZOOM:YAXIS:FROM -1.0E+20 sets the bottom value of the zoom y-axis to -1E+20.

DISplay:PLOTView1:ZOOM:YAXIS:FROM? might return :DISPLAY:PLOTVIEW1:ZOOM:YAXIS:FROM -1.0E+21 indicating the bottom value of the zoom y-axis is -1E+21.

DISplay:PLOTView<x>:ZOOM:YAXIS:TO

This command sets or queries the top value of the zoom y-axis in the specified plot view.

Group Zoom

Syntax

```
DISplay:PLOTView<x>:ZOOM:YAXIS:TO <NR3>
DISplay:PLOTView<x>:ZOOM:YAXIS:TO?
```

Arguments PLOTView<x> is the Plot waveform number.

<NR3> is the top value of the zoom y-axis.

Examples DISplay:PLOTView1:ZOOM:YAXIS:TO 100 sets the top value of the zoom y-axis to 100.

DISplay:PLOTView1:ZOOM:YAXIS:TO? might return :DISPLAY:PLOTVIEW3:ZOOM:YAXIS:FROM 0.0E+0 indicating the top value of the zoom y-axis is 0.0 hits (for a histogram plot, or it could be Volts for an XY plot).

DISplay:REFFFTView<n>:CURSor:ROLOCATION

This command sets or queries the location to display the specified Reference FFT plot cursor readouts (in the plot graticule or in a badge in the Results Bar).

Group Cursor

Syntax

```
DISplay:REFFFTView<n>:CURSor:ROLOCATION {GRATICULE|BADGE}
DISplay:REFFFTView<n>:CURSor:ROLOCATION?
```

Arguments

PLOTView<x> is the Reference FFT plot number.

GRATICULE sets the Reference FFT plot cursor readouts to display as part of the cursors in the plot view.

BADGE removes the Reference FFT plot cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.

Examples

DISplay:REFFFTView1:CURSor:ROLOCATION GRATICULE sets the cursor readout information for Reference FFT plot 1 to display in the plot graticule.

DISplay:REFFFTView2:CURSor:ROLOCATION? might return DISplay:REFFFTView2:CURSor:ROLOCATION BADGE, indicating that cursor readout information for Reference FFT plot 2 is set to display as a badge in the Results Bar.

DISplay:REFFFTView<x>:AUTOScale

This command sets or queries the enabled state of auto-scale for plots.

Group Display Control

Syntax

```
DISplay:REFFFTView<x>:AUTOScale {OFF|ON|0|1}
DISplay:REFFFTView<x>:AUTOScale?
```

Arguments

REFFFTView<x> is the plot number.

<NR1> = 0 disables auto-scale on the specified refffftview; any other value turns this feature on.

OFF disables auto-scale on the specified refffftview.

ON enables the specified channel on the specified Waveform View.

Examples `DISplay:REFFFTView1:AUTOScale 1` enables auto-scale on the specified view.

`DISplay:REFFFTView5:AUTOScale?` might return `:DISPLAY:REFFFTVIEW5:AUTOSCALE 1` indicating auto-scale is on.

DISplay:REFFFTView<x>:CURSor:ASOUrce? (Query Only)

This command returns the cursor source for plot cursor A

Group Cursor

Syntax `DISplay:REFFFTView<x>:CURSor:ASOUrce?`

Arguments `PLOTView<x>` is the Plot waveform number.

Returns Returns the cursor source for plot cursor A.

Examples `DISplay:REFFFTView5:CURSor:ASOUrce?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:ASOURCE REF5` indicating the A cursor source is reference 5.

DISplay:REFFFTView<x>:CURSor:BSOUrce? (Query Only)

This command returns the cursor source for plot cursor B.

Group Cursor

Syntax `DISplay:REFFFTView<x>:CURSor:BSOUrce?`

Returns Returns the cursor source for plot cursor B.

Examples `DISplay:REFFFTView5:CURSor:BSOUrce?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:BSOURCE REF5` indicating the source of the B cursor is reference 5.

DISplay:REFFFTView<x>:CURSor:DDT? (Query Only)

This command returns the delta V over delta T cursor readout value of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:DDT?
```

Returns Returns the delta V over delta T cursor readout value

Examples `DISplay:REFFFTView5:CURSor:DDT?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:DDT 9.91E+37` indicating the delta V over delta T cursor readout value is 9.91E+37.

DISplay:REFFFTView<x>:CURSor:FUNCtion

This command sets or queries the cursor type of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:FUNCtion
{WAVEform|VBArs|HBArs|SCREEN}
DISplay:REFFFTView<x>:CURSor:FUNCtion?
```

Arguments

`HBArs` specifies horizontal bar cursors, which measure in vertical units.

`VBArs` specifies vertical bar cursors, which measure in horizontal units.

`SCREEN` specifies both horizontal and vertical bar cursors, which measure in horizontal and vertical units specified by the cursor sources. Use these cursors to measure anywhere in the waveform display area.

`WAVEform` specifies paired or split cursors in YT display format for measuring waveform amplitude and time. In XY and XYZ format, these cursors indicate the amplitude positions of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.

Examples `DISplay:REFFFTView5:CURSor:FUNCtion screen` sets the cursor function to screen.

DISplay:REFFFTView5:CURSor:FUNCtion? might return :DISPLAY:REFFFTVIEW5:CURSOR:FUNCTION WAVEFORM indicating the cursor function is set to waveform.

DISplay:REFFFTView<x>:CURSor:HBArs:APOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:HBArs:APOSition <NR3>
DISplay:REFFFTView<x>:CURSor:HBArs:APOSition?
```

Arguments <NR3> is the vertical cursor A position of the specified cursor in the specified view.

Examples DISplay:REFFFTView5:CURSor:HBArs:APOSition -20 sets the A cursor position to -20.

DISplay:REFFFTView5:CURSor:HBArs:APOSition? might return :DISPLAY:REFFFTVIEW5:CURSOR:HBARS:APOSITION -14.4762489421361 indicating the A cursor position is -14.48 dBm.

DISplay:REFFFTView<x>:CURSor:HBArs:AUNIts? (Query Only)

This command returns cursor A vertical units of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:HBArs:AUNIts?
```

Returns Returns the cursor A vertical units of the specified cursor in the specified view.

Examples DISplay:REFFFTView5:CURSor:HBArs:AUNIts? might return :DISPLAY:REFFFTVIEW5:CURSOR:HBARS:AUNITS "dBm" indicating the A cursor units are dBm.

DISplay:REFFFTView<x>:CURSor:HBArs:BPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:HBArs:BPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:HBArs:BPOSition?
```

Arguments <NR3> is the vertical cursor B position of the specified cursor in the specified view.

Examples `DISplay:REFFFTView5:CURSor:HBArs:BPOSition -90` sets the B cursor position to -90.

`DISplay:REFFFTView5:CURSor:HBArs:BPOSition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:HBARS:BPOSITION -100.6262359333038` indicating the B cursor position is -100.6.

DISplay:REFFFTView<x>:CURSor:HBArs:BUNIts? (Query Only)

This command returns the cursor B vertical units of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:HBArs:BUNIts?
```

Returns Returns the cursor B vertical units of the specified cursor in the specified view.

Examples `DISplay:REFFFTView5:CURSor:HBArs:BUNIts?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:HBARS:BUNITS "dBm"` indicating the units are dBm.

DISplay:REFFFTView<x>:CURSor:HBArs:DELTa? (Query Only)

This command returns the delta V cursor readout value of the specified cursor in the specified view.

Group Cursor

Syntax `DISplay:REFFFTView<x>:CURSor:HBArs:DELTa?`

Returns Returns the delta V cursor readout value of the specified cursor in the specified view.

Examples `DISplay:REFFFTView5:CURSor:HBArs:DELTa?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:HBARS:DELTA 59.9867396737569` indicating the delta V cursor readout value is 59.99.

DISplay:REFFFTView<x>:CURSor:MODe

This command sets or queries the cursor tracking mode of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:MODe {INDEPENDENT|TRACK}
DISplay:REFFFTView<x>:CURSor:MODe?
```

Arguments `TRACK` ties the navigational functionality of the two cursors together. For cursor A adjustments, this ties the movement of the two cursors together; however, cursor B continues to move independently of cursor A.

`INDEPENDENT` allows independent adjustment of the two cursors.

Examples `DISplay:REFFFTView5:CURSor:MODe TRACK` sets the cursors to track together.

`DISplay:REFFFTView5:CURSor:MODe?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:MODE INDEPENDENT` indicating the cursors move independently.

DISplay:REFFFTView<x>:CURSor:ONEOVERDELTATVALUE? (Query Only)

This command returns the one over delta T cursor readout value of the specified cursor in the specified view.

Group Cursor

Syntax `DISplay:REFFFTView<x>:CURSor:ONEOVERDELTATVALUE?`

Examples `DISplay:REFFFTView5:CURSor:ONEOVERDELTATVALUE?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:ONEOVERDELTATVALUE 2.540935140340E-9` indicating the one over delta T cursor readout value is 2.54 ns.

DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:SCREEN:AXPOSition?
```

Arguments `<NR3>` is the horizontal cursor A position of the specified cursor in the specified view.

Examples `DISplay:REFFFTView5:CURSor:SCREEN:AXPOSition 1.0E+6` sets the cursor position to 1.0 MHz.

`DISplay:REFFFTView5:CURSor:SCREEN:AXPOSition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:AXPOSITION 792.4465962305570E+3` indicating the A cursor horizontal position is 792.447 kHz.

DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:SCREEN:AYPOSition?
```

Arguments `<NR3>` is the vertical cursor A position of the specified cursor in the specified view.

Examples `DISplay:REFFFTView5:CURSor:SCREEN:AYPOSition -20.9` sets the position to -20 dBm.

`DISplay:REFFFTView5:CURSor:SCREEN:AYPOSition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:AYPOSITION -30.0132603262431` indicating the A cursor vertical position is -30.01 dBm.

DISplay:REFFFTView<x>:CURSor:SCREEN:BXPOSition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:SCREEN:BXPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:SCREEN:BXPOSition?
```

Arguments `<NR3>` is the horizontal cursor B position of the specified cursor in the specified view.

Examples `DISplay:REFFFTView5:CURSor:SCREEN:BXPOSition 300` sets the position to 300 MHz.

`DISplay:REFFFTView5:CURSor:SCREEN:BXPOSition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:BXPOSITION 394.3483403001212E+6` indicating the B cursor horizontal position is 394.348 MHz.

DISplay:REFFFTView<x>:CURSor:SCREEN:BYPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:SCREEN:BYPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:SCREEN:BYPOSition?
```

Arguments `<NR3>` is the vertical cursor B position of the specified cursor in the specified view.

Examples

`DISplay:REFFFTView5:CURSor:SCREEN:BYPOSition -80` sets the cursor position to -90 dBm.

`DISplay:REFFFTView5:CURSor:SCREEN:BYPOSition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:SCREEN:BYPOSITION -90.0` indicating the B cursor vertical position is -90 dBm.

DISplay:REFFFTView<x>:CURSor:SPLITMODE

This command sets or queries whether both cursors have same or different source.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:SPLITMODE {SAME|SPLIT}
DISplay:REFFFTView<x>:CURSor:SPLITMODE?
```

Arguments

`SAME` specifies both cursors have the same sources.

`SPLIT` specifies both cursors have different sources.

Examples

`DISplay:REFFFTView5:CURSor:SPLITMODE SPLIT` specified that the cursors have different sources.

`DISplay:REFFFTView5:CURSor:SPLITMODE?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:SPLITMODE SAME` indicating both cursors have the same source.

DISplay:REFFFTView<x>:CURSor:STATE

This command sets or queries the visible state of the cursor of the specified cursor n the specified view.

Group

Cursor

Syntax

```
DISplay:REFFFTView<x>:CURSor:STATE {OFF|ON|0|1}
DISplay:REFFFTView<x>:CURSor:STATE?
```

Arguments

`<NR1>` = 0 specifies the cursor is not visible; any other value displays the cursor.

`OFF` specifies the cursor is not visible.

`ON` displays the cursor.

Examples `DISplay:REFFFTView5:CURSor:STATE ON` specifies the cursor is visible.

`DISplay:REFFFTView5:CURSor:STATE?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:STATE 1` indicating the cursor is visible.

DISplay:REFFFTView<x>:CURSor:VBArs:APOSition

This command sets or queries the horizontal cursor A position of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:VBArs:APOSition <NR3>
DISplay:REFFFTView<x>:CURSor:VBArs:APOSition?
```

Arguments `<NR3>` is the horizontal cursor A position of the specified cursor in the specified view.

Examples `DISplay:REFFFTView5:CURSor:VBArs:APOSition 1.0E+6` sets the cursor to 1 MHz.

`DISplay:REFFFTView5:CURSor:VBArs:APOSition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:VBARS:APOSITION 792.4465962305570E+3` indicating the cursor position is 792.447 kHz.

DISplay:REFFFTView<x>:CURSor:VBArs:BPOSition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified view.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:VBArs:BPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:VBArs:BPOSition?
```

Arguments `<NR3>` is the horizontal cursor B position of the specified cursor in the specified view.

Examples `DISplay:REFFFTView5:CURSor:VBArs:BPOSition 300.0E+6` sets the cursor position to 300 MHz.

DISplay:REFFFTView5:CURSor:VBArs:BPOSition? might return :DISPLAY:REFFFTVIEW5:CURSOR:VBARS:BPOSITION 394.3483403001212E+6 indicating the B cursor horizontal position is 394.348 MHz.

DISplay:REFFFTView<x>:CURSor:VBArs:DELTa? (Query Only)

This command returns the delta T cursor readout value of the specified cursor in the specified view.

Group Cursor

Syntax DISplay:REFFFTView<x>:CURSor:VBArs:DELTa?

Returns Returns the delta T cursor readout value of the specified cursor in the specified view.

Examples DISplay:REFFFTView5:CURSor:VBArs:DELTa? might return :DISPLAY:REFFFTVIEW5:CURSOR:VBARS:DELTA 393.5558937038906E+6 indicating the delta T cursor readout value is 393.556 MHz.

DISplay:REFFFTView<x>:CURSor:VBArs:UNIts? (Query Only)

This command returns cursor A vertical units of the specified cursor in the specified view.

Group Cursor

Syntax DISplay:REFFFTView<x>:CURSor:VBArs:UNIts?

Returns Returns cursor A vertical units of the specified cursor in the specified view.

Examples DISplay:REFFFTView5:CURSor:VBArs:UNIts? might return :DISPLAY:REFFFTVIEW5:CURSOR:VBARS:UNITS "Hz" indicating the A cursor vertical units are Hz.

DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOSition? (Query Only)

This query-only command returns the value of the cursor A horizontal position.

Group Cursor

Syntax `DISplay:REFFFTView<x>:CURSor:WAVEform:AHPOSition?`

Examples `DISplay:REFFFTView1:CURSor:WAVEform:AHPOSition?` might return `:DISPLAY:REFFFTVIEW1:CURSOR:WAVEFORM:AHPOSITION 9.91E+37`.

DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition

Sets or returns the waveform cursor A position in the specified plot view.

NOTE. *In case of an XY plot, this command has no effect when used to set the value. In the case of a bathtub plot, this command sets or queries the cursor A vertical position. For all other plots, this command sets or queries the cursor A horizontal position.*

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition <NR3>
DISplay:REFFFTView<x>:CURSor:WAVEform:APOSition?
```

Arguments `<NR3>` is the waveform cursor A position in the specified plot view.

Examples `DISplay:REFFFTView5:CURSor:WAVEform:APOSition 7E+5` sets the position to 700 kHz.

`DISplay:REFFFTView5:CURSor:WAVEform:APOSition?` might return `:DISPLAY:REFFFTVIEW5:CURSOR:WAVEFORM:APOSITION 792.4465962305570E+3` indicating the A cursor waveform position is 792.4 kHz.

DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition? (Query Only)

This query-only command returns the value of the cursor A vertical position.

Group Cursor

Syntax `DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition?`

Examples `DISplay:REFFFTView<x>:CURSor:WAVEform:AVPOSition?` might return `:DISplay:REFFFTView1:CURSor:WAVEform:AVPOSition 1.605E+0` indicating the cursor vertical position is 1.6 dBm.

DISplay:REFFFTView<x>:CURSor:WAVEform:BHPOSition? (Query Only)

This query-only command returns the value of the cursor B horizontal position.

Group Cursor

Syntax `DISplay:REFFFTView<x>:CURSor:WAVEform:BHPOSition?`

Examples `DISplay:REFFFTView<x>:CURSor:WAVEform:BHPOSition?` might return `:DISPLAY:REFFFTVIEW1:CURSOR:WAVEFORM:BHPOSITION 9.91E+37`.

DISplay:REFFFTView<x>:CURSor:WAVEform:BPOSition

Sets or returns the waveform cursor B position in the specified plot view.

NOTE. *In case of an XY plot, this command has no effect when used to set the value. In the case of a bathtub plot, this command has no effect. Queries return invalid values. For all other plots, this command sets or queries the cursor B horizontal position.*

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:WAVEform:BPOSition <NR3>
DISplay:REFFFTView<x>:CURSor:WAVEform:BPOSition?
```

Arguments `<NR3>` is the waveform cursor B position in the specified plot view.

Examples `DISplay:REFFFTView5:CURSor:WAVEform:BPOSition 4E+8` sets the cursor position to 400 MHz.

DISplay:REFFFTView5:CURSor:WAVEform:BPOSition? might return :DISPLAY:REFFFTVIEW5:CURSOR:WAVEFORM:BPOSITION 394.3483403001212E+6 indicating the B waveform curosr position is 394.35 MHz.

DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOSition? (Query Only)

This query-only command returns the value of the cursor B vertical position.

Group Cursor

Syntax
```
DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOSition?
```

Examples DISplay:REFFFTView1:CURSor:WAVEform:BVPOSition? might return :DISplay:REFFFTView<x>:CURSor:WAVEform:BVPOSition -119.866E+0 indicating the cursor vertical position is -119.9 dBm.

DISplay:REFFFTView<x>:GRIDlines

This command sets or returns the grid lines setting of the plot.

Group Display Control

Syntax
```
DISplay:REFFFTView<x>:GRIDlines {HORizontal|VERTical|BOTH}
DISplay:REFFFTView<x>:GRIDlines?
```

Arguments HORizontal specifies horizontal grid lines.

VERTical specifies vertical grid lines.

BOTH specifies both horizontal and vertical grid lines.

Examples DISplay:REFFFTView5:GRIDlines hhorizontal specifies horizontal gird lines.

DISplay:REFFFTView5:GRIDlines? might return :DISPLAY:REFFFTVIEW5:GRIDLINES BOTH indicating that both vertical and horizontal grid lines are diplayed

DISplay:REFFFTView<x>:REF:REF<x>:STATE

This command sets or queries the state of the specified reference waveform in the specified Waveform View.

Group Display Control

Syntax

```
DISplay:REFFFTView<x>:REF:REF<x>:STATE boolean
DISplay:REFFFTView<x>:REF:REF<x>:STATE?
```

Arguments

<NR1> = 0 disables the specified reference; any other value turns this feature on.

OFF disables the display the specified reference.

ON enables the specified reference.

Examples

DISplay:REFFFTView5:REF:REF5:STATE 0 turns display of the reference off.

DISplay:REFFFTView5:REF:REF5:STATE? might return :DISPLAY:REFFFTVIEW5:REF:REF5:STATE 1 indicating the reference is displayed.

DISplay:REFFFTView<x>:XAXIS:SCALE

This command sets or queries the x-axis scale setting for Ref FFT.

Group Display Control

Syntax

```
DISplay:REFFFTView<x>:XAXIS:SCALE {LINEAr|LOG}
DISplay:REFFFTView<x>:XAXIS:SCALE?
```

Arguments

LINEAr specifies a linear scale.

LOG specifies a logarithmic scale.

Examples

DISplay:REFFFTView5:XAXIS:SCALE linear sets the x-axis scale to linear.

DISplay:REFFFTView5:XAXIS:SCALE? might return :DISPLAY:REFFFTVIEW5:XAXIS:SCALE LOG indicating x-axis scale for the specified reference view is logarithmic.

DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM

This command sets or returns the left edge of the zoom x-axis in the specified plot view.

Group Zoom

Syntax
```
DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM <NR3>
DISplay:REFFFTView<x>:ZOOM:XAXIS:FROM?
```

Arguments <NR3> is the left edge of the zoom x-axis in the specified plot view.

Examples DISplay:REFFFTView5:ZOOM:XAXIS:FROM 2.0E+6 sets the left edge of the specified view to 2.0 MHz.

DISplay:REFFFTView5:ZOOM:XAXIS:FROM? might return :DISPLAY:REFFFTVIEW5:ZOOM:XAXIS:FROM 1.3295739742362E+6 indicating the left edge of the specified view is 1.33 MHz.

DISplay:REFFFTView<x>:ZOOM:XAXIS:TO

This command sets or queries the right edge of the zoom x-axis in the specified plot view.

Group Zoom

Syntax
```
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO <NR3>
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO?
```

Arguments <NR3> is the right edge of the zoom x-axis in the specified plot view.

Examples DISplay:REFFFTView5:ZOOM:XAXIS:TO 2.0E+8 sets the right edge of the zoom x-axis to 200 MHz.

DISplay:REFFFTView5:ZOOM:XAXIS:TO? might return :DISPLAY:REFFFTVIEW5:ZOOM:XAXIS:TO 235.0376933178995E+6 indicating the right edge of the zoom x-axis is 235.0 MHz.

DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM

This command sets or queries the bottom value of the zoom y-axis in the specified plot view.

Group Zoom

Syntax

```
DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM <NR3>
DISplay:REFFFTView<x>:ZOOM:YAXIS:FROM?
```

Arguments <NR3> is the bottom value of the zoom y-axis in the specified plot view.

Examples DISplay:REFFFTView5:ZOOM:YAXIS:FROM -120.0 sets the bottom value of the zoom y-axis to -120.0 dBm.

DISplay:REFFFTView5:ZOOM:YAXIS:FROM? might return :DISPLAY:REFFFTVIEW5:ZOOM:YAXIS:FROM -129.7533120632172 indicating the bottom value of the zoom y-axis is -129.8 dBm.

DISplay:REFFFTView<x>:ZOOM:YAXIS:TO

This command sets or queries the top value of the zoom y-axis in the specified plot view.

Group Zoom

Syntax

```
DISplay:REFFFTView<x>:ZOOM:YAXIS:TO <NR3>
DISplay:REFFFTView<x>:ZOOM:YAXIS:TO?
```

Arguments <NR3> is the top value of the zoom y-axis in the specified plot view.

Examples DISplay:REFFFTView5:ZOOM:YAXIS:TO 0.0 sets the top value of the zoom y-axis to 0.0 dBm.

DISplay:REFFFTView5:ZOOM:YAXIS:TO? might return :DISPLAY:REFFFTVIEW5:ZOOM:YAXIS:TO 15.8820685863495 indicating the top value of the zoom y-axis is 15.9 dBm.

DISplay:SELect:BUS

This command sets or queries the overall selected bus. Sets are applied to all views that contain the source and the selected view is changed. When multiple buses are open, querying the command gives the correct result, but the bus cannot set.

Group Display Control

Syntax

```
DISplay:SELect:BUS BUS<x>
DISplay:SELect:BUS?
```

Arguments `BUS<x>` is the selected bus.

Examples `DISplay:SELect:BUS Bus2` sets Bus 2 as the selected bus.

`DISplay:SELect:BUS?` might return `:DISPLAY:SELECT:BUS BUS1` indicating Bus 1 is the selected bus.

DISplay:SELect:MATH

This command sets or queries the overall selected math. Sets are applied to all views that contain the source and the selected view is changed. When multiple Math are open, querying the command gives the correct result, but the required Math cannot be set.

Group Display Control

Syntax

```
DISplay:SELect:MATH MATH<x>
DISplay:SELect:MATH?
```

Arguments `MATH<x>` is the selected math.

Examples `DISplay:SELect:MATH MATH1` selects Math 1.

`DISplay:SELect:MATH?` might display `:DISPLAY:SELECT:MATH MATH2` indicating that Math 2 is selected.

DISplay:SELect:REFerence

This command sets or queries the overall selected reference waveform. Sets are applied to all views that contain the source and the selected view is changed.

Group Display Control

Syntax `DISplay:SELect:REFerence {NONE|REF<x>}`

Arguments Arguments are the selected reference.

Examples `DISplay:SELect:REFerence REF2` selects reference 2.

`DISplay:SELect:REFERENCE?` might return `DISPLAY:SELECT:REFERENCE NONE` indicating no reference is selected.

DISplay:SELect:SOUrce

This command sets or queries the overall selected source. Sets are applied to all views that contain the source and the selected view is changed.

Group Display Control

Syntax `DISplay:SELect:SOUrce {NONE|CH<x>|BUS<x>|MATH<x>|PLOT<x>|REF<x>}`

Arguments Arguments are the selected source.

Examples `DISplay:SELect:SOUrce Bus1` selects Bus1 as the selected source.

`DISplay:SELect:SOUrce?` might return `:DISPLAY:SELECT:SOURCE MATH1` indicating Math 1 is the selected source.

DISplay:SELect:SPECView<x>:SOUrce

This command sets or queries the spectrum trace (source channel) to select in the Spectrum View window.

Group Display

Syntax

```
DISplay:SELect:SPECView<x>:SOUrce {CH<x>}
DISplay:SELect:SPECView<x>:SOUrce?
```

Arguments

SPECView<x> specifies the Spectrum View number. <x> is always 1.

CH<x> specifies the spectrum trace (source channel) to select (make active) in the Spectrum View.

Examples

DISplay:SELect:SPECView1:SOUrce CH1 sets the spectrum trace associated with channel 1 as the selected trace.

DISplay:SELect:SPECView1:SOUrce? might return CH3, indicating that the spectrum trace associated with channel 3 is selected in the Spectrum View.

DISplay:SELect:VIEW

This command sets or queries the selected view.

Group

Display Control

Syntax

```
DISplay:SELect:VIEW
{WAVEVIEW1|MATHFFT<x>|PLOTVIEW<x>|REFFFT<x>}
```

Arguments

Arguments are the selected view.

Examples

DISplay:SELect:VIEW PLOTVIEW1 selects PLOTVIEW1 as the selected view.

DISplay:SELect:VIEW? might return :DISPLAY:SELECT:VIEW MATHFFT1 indicating MATHFFT1 is the selected view.

DISplay:SELect:WAVEView<x>:SOUrce

This command sets or queries the selected source in the given waveview.

Group

Display Control

Syntax

```
DISplay:SELect:WAVEView<x>:SOUrce
{CH<x>|MATH<x>|BUS<x>|REF<x>|PLOT<x>}
```

Arguments

Arguments are the selected source.

Examples

`DISplay:SELect:WAVEView1:SOUrce MATH1` sets `MATH1` as the selected source.

`DISplay:SELect:WAVEView1:SOUrce?` might return `:DISPLAY:SELECT:WAVEVIEW1:SOURCE CH2` indicating `CH2` is the selected source.

DISplay:SPECView<x>:CURSor:CURSOR:ANOISEDensity? (Query Only)

Returns the noise density of the selected RF trace at the Cursor A position

Group

Cursor

Syntax

`DISplay:SPECView<x>:CURSor:CURSOR:ANOISEDensity?`

Related Commands

DISplay:SPECView<x>:CURSor:CURSOR:BNOISE?

Arguments

SPECView<x> is the specified Spectrum View and must be SPECView1.

Returns

The noise density of the selected RF trace at the horizontal position of Cursor A in <RF Units>/Hz units, where <RF Units> are the vertical units of the RF trace.

Examples

`DISplay:SPECView1:CURSor:CURSOR:ANOISEDensity?` might return `–133.2350`, indicating a noise density of –133.2350 dBm/Hz at the Cursor A position.

DISplay:SPECView<x>:CURSor:CURSOR:ASOUrce

This command sets or queries the Cursor A source in the Spectrum View.

Group

Cursor

Syntax

`DISplay:SPECView<x>:CURSor:CURSOR:ASOUrce {AUTO|CH<x>}`
`DISplay:SPECView<x>:CURSor:CURSOR:ASOUrce?`

Arguments

`SPECView<x>` specifies the Spectrum View number. <x> is always 1.

`AUTO` specifies to use the selected analog channel as the Spectrum View source.

`CH<x>` specifies the analog channel to use as the Spectrum View source.

Examples

`DISplay:SPECView2:CURSor:CURSOR:ASOUrce CH1` sets the Cursor A source of Spectrum View 2 to Channel 1.

`DISplay:SPECView1:CURSor:CURSOR:ASOUrce?` might return `DISplay:SPECView1:CURSor:CURSOR:ASOUrce AUTO`, indicating the Cursor A source of Spectrum View 1 is the selected source.

DISplay:SPECView<x>:CURSor:CURSOR:BNOISE? (Query Only)

Returns the noise density of the selected RF trace at the Cursor B position when in Absolute mode, or the phase noise when in Delta mode.

Group

Cursor

Syntax

`DISplay:SPECView<x>:CURSor:CURSOR:BNOISE?`

Related Commands

DISplay:SPECView<x>:CURSor:CURSOR:ANOISEDensity?

DISplay:SPECView<x>:CURSor:CURSOR:READout

Arguments

SPECView<x> is the specified Spectrum View and must be SPECView1.

Returns

In Absolute mode: The noise density of the selected RF trace at the horizontal position of Cursor B in <RF Units>/Hz units, where <RF Units> are the vertical units of the RF trace.

In Delta mode: The phase noise of the selected RF trace at the horizontal position of Cursor B in dBc/Hz units.

Examples

`DISplay:SPECView1:CURSor:CURSOR:BNOISE?` might return `–60.1289`, indicating a phase noise of –60.1289 dBc/Hz at the Cursor B position in Delta mode.

`DISplay:SPECView1:CURSor:CURSOR:BNOISE?` might return `–47.922`, indicating a phase noise of –47.922 dBm/Hz at the Cursor B position in Absolute mode.

DISplay:SPECView<x>:CURSor:CURSOR:BSOUrce

This command sets or queries the Cursor B source in the Spectrum View.

Group

Cursor

Syntax

```
DISplay:SPECView<x>:CURSor:CURSOR:BSOUrce {AUTO|CH<x>}
DISplay:SPECView<x>:CURSor:CURSOR:BSOUrce?
```

Arguments

SPECView<x> specifies the Spectrum View number. <x> is always 1.

AUTO specifies to use the selected analog channel as the Spectrum View source.

CH<x> specifies the analog channel to use as the Spectrum View source.

Examples

DISplay:SPECView1:CURSor:CURSOR:ASOUrce CH1 sets the Cursor A source of Spectrum View 1 to Channel 1.

DISplay:SPECView1:CURSor:CURSOR:ASOUrce? might return DISplay:SPECView1:CURSor:CURSOR:ASOUrce AUTO, indicating the Cursor A source of Spectrum View 1 is the selected source.

DISplay:SPECView<x>:CURSor:CURSOR:HBArs:APOSition

This command queries the Cursor A horizontal bar's vertical position (where it crosses the spectrum signal) in the Spectrum View.

Group

Cursor

Syntax

```
DISplay:SPECView<x>:CURSor:CURSOR:HBArs:APOSition?
```

Related Commands

DISplay:SPECView<x>:CURSor:CURSOR:HBArs:AUNIts?

Arguments

SPECView<x> specifies the Spectrum View number. <x> is always 1.

Returns

The vertical position of Cursor A in the vertical units of the cursor source.

Examples

DISplay:SPECView1:CURSor:CURSOR:HBArs:APOSition? might return DISplay:SPECView1:CURSor:CURSOR:HBArs:APOSition -40.0 indicating the Cursor A vertical position is -40.0 units (such as dBm).

DISplay:SPECView<x>:CURSor:CURSOR:HBArs:AUNIts? (Query Only)

This command queries the horizontal bar Cursor A horizontal bar's amplitude units in the Spectrum View.

Group Cursor

Syntax `DISplay:SPECView<x>:CURSor:CURSOR:HBArs:AUNIts?`

Related Commands DISplay:SPECView<x>:CURSor:CURSOR:HBArs:APOSition

Arguments `SPECView<x>` specifies the Spectrum View number. <x> is always 1.

Returns The vertical units of the Spectrum View Cursor A as a Qstring.

Examples `DISplay:SPECView1:CURSor:CURSOR:HBArs:AUNIts?` might return `DISplay:SPECView1:CURSor:CURSOR:HBArs:AUNIts "dBmV"`, indicating the Cursor A vertical units are decibel millivolts.

DISplay:SPECView<x>:CURSor:CURSOR:HBArs:BPOSition

This command queries the horizontal bar Cursor B horizontal bar's vertical position (where it crosses the spectrum signal) in the Spectrum View.

Group Cursor

Syntax `DISplay:SPECView<x>:CURSor:CURSOR:HBArs:BPOSition?`

Related Commands DISplay:SPECView<x>:CURSor:CURSOR:HBArs:BUNIts?

Arguments `SPECView<x>` specifies the Spectrum View number. <x> is always 1.

Returns The vertical position of Cursor B in the vertical units of the cursor source.

Examples `DISplay:SPECView1:CURSor:CURSOR:HBArs:BPOSition?` might return `DISplay:SPECView1:CURSor:CURSOR:HBArs:BPOSition -67.8` indicating the Cursor A vertical position is -67.8 units (such as dBm).

DISplay:SPECView<x>:CURSor:CURSOR:HBArs:BUNIts? (Query Only)

This command queries the Cursor B horizontal bar's amplitude units in the Spectrum View.

Group Cursor

Syntax `DISplay:SPECView<x>:CURSor:CURSOR:HBArs:BUNIts?`

Related Commands DISplay:SPECView<x>:CURSor:CURSOR:HBArs:BPOSition

Arguments `SPECView<x>` specifies the Spectrum View number. <x> is always 1.

Returns The vertical units of the Spectrum View Cursor B as a Qstring.

Examples `DISplay:SPECView1:CURSor:CURSOR:HBArs:BUNIts?` might return `DISplay:SPECView1:CURSor:CURSOR:HBArs:BUNIts "dBmV"`, indicating the Cursor A vertical units are decibel millivolts.

DISplay:SPECView<x>:CURSor:CURSOR:READout

This command sets or queries the Cursor B readout type in the Spectrum View.

Group Cursor

Syntax
```
DISplay:SPECView<x>:CURSor:CURSOR:READout {ABSOLUTE|DELTA}
DISplay:SPECView<x>:CURSor:CURSOR:READout?
```

Arguments `SPECView<x>` specifies the Spectrum View number. <x> is always 1.

`ABSOLUTE` specifies the Cursor B readout displays absolute frequency and amplitude.

`DELTA` specifies the Cursor B readout displays delta frequency and amplitude from Cursor A.

Examples `DISplay:SPECView1:CURSor:CURSOR:READout ABSOLUTE` sets the readout type to Absolute for Cursor B.

`DISplay:SPECView1:CURSor:CURSOR:READout?` might return `DISplay:SPECView1:CURSor:CURSOR:READout DELTA`, indicating the readout type is Delta between Cursor A and B.

DISplay:SPECView<x>:CURSor:CURSOR:SPLITMODE

This command sets or queries whether both Spectrum View cursors have the same or different sources.

Group Cursor

Syntax
```
DISplay:SPECView<x>:CURSor:CURSOR:SPLITMODE {SAME|SPLIT}
DISplay:SPECView<x>:CURSor:CURSOR:SPLITMODE?
```

Arguments

`SPECView<x>` specifies the Spectrum View number. <x> is always 1.

`SAME` specifies that both Spectrum View cursors have the same source

`SPLIT` specifies that both Spectrum View cursors have independent sources.

Examples

`DISplay:SPECView1:CURSor:CURSOR:SPLITMODE SPLIT` sets the Spectrum View cursors have independent sources.

`DISplay:SPECView1:CURSor:CURSOR:SPLITMODE?` might return `DISplay:SPECView1:CURSor:CURSOR:SPLITMODE SAME`, indicating the Spectrum View cursors have the same source.

DISplay:SPECView<x>:CURSor:CURSOR:STATE

This command sets or queries the visibility state of the cursors in the Spectrum View.

Group Cursor

Syntax
```
DISplay:SPECView<x>:CURSor:CURSOR:STATE {ON|OFF}
DISplay:SPECView<x>:CURSor:CURSOR:STATE?
```

Arguments

`SPECView<x>` specifies the Spectrum View number. <x> is always 1.

`ON` enables viewing cursors in the Spectrum View.

`OFF` disables viewing cursors in the Spectrum View.

Examples

`DISplay:SPECView1:CURSor:CURSOR:STATE OFF` enables (displays) cursors in the Spectrum View.

DISplay:SPECView1:CURSor:CURSOR:STATE? might return DISplay:SPECView1:CURSor:CURSOR:STATE ON, indicating the Spectrum View cursors are enabled (being displayed).

DISplay:SPECView<x>:CURSor:CURSOR:VBArs:APOSition

This command sets or queries the Cursor A vertical bar's horizontal position in the Spectrum View.

Group Cursor

Syntax
```
DISplay:SPECView<x>:CURSor:CURSOR:VBArs:APOSition <NR3>
DISplay:SPECView<x>:CURSor:CURSOR:VBArs:APOSition?
```

Arguments SPECView<x> specifies the Spectrum View number. <x> is always 1.

<NR3> specifies the horizontal position of Cursor A in Hz. The available frequency range is set by the frequencies at the left and right edges of the Spectrum View for the cursor source.

Examples DISplay:SPECView1:CURSor:CURSOR:VBArs:APOSition 50E+6 sets the Spectrum View Cursor B position to 50 MHz.

DISplay:SPECView1:CURSor:CURSOR:VBArs:APOSition? might return DISplay:SPECView1:CURSor:CURSOR:VBArs:APOSition 48.5E+6, indicating the Spectrum View Cursor B position is 48.5 MHz.

DISplay:SPECView<x>:CURSor:CURSOR:VBArs:BPOSition

This command sets or queries the Cursor B vertical bar's horizontal position in the Spectrum View.

Group Cursor

Syntax
```
DISplay:SPECView<x>:CURSor:CURSOR:VBArs:BPOSition <NR3>
DISplay:SPECView<x>:CURSor:CURSOR:VBArs:BPOSition?
```

Arguments SPECView<x> specifies the Spectrum View number. <x> is always 1.

<NR3> specifies the horizontal position of Cursor B in Hz. The available frequency range is set by the frequencies at the left and right edges of the Spectrum View for the cursor source.

Examples `DISplay:SPECView1:CURSor:CURSOR:VBArs:BPOSition 55E+6` sets the Spectrum View Cursor B position to 55 MHz.

`DISplay:SPECView1:CURSor:CURSOR:VBArs:APOSition?` might return `DISplay:SPECView1:CURSor:CURSOR:VBArs:APOSition 482.5E+6`, indicating the Spectrum View Cursor B position is 42.5 MHz.

DISplay:SPECView<x>:CURSor:CURSOR:VBArs:UNIts? (Query Only)

This command queries the cursor horizontal units in the Spectrum View.

Group Cursor

Syntax `DISplay:SPECView<x>:CURSor:CURSOR:VBArs:UNIts?`

Arguments `SPECView<x>` specifies the Spectrum View number. <x> is always 1.

Returns The horizontal units of the Spectrum View as a Qstring.

Examples `DISplay:SPECView1:CURSor:CURSOR:VBArs:UNIts?` might return `DISplay:SPECView1:CURSor:CURSOR:VBArs:UNIts "Hz"`, indicating the cursor horizontal units are Hertz.

DISplay:SPECView<x>:CURSor:CURSOR:WAVEform:APOSition

This command sets or queries the Cursor A horizontal position in the Spectrum View.

Group Cursor

Syntax
```
DISplay:SPECView<x>:CURSor:CURSOR:WAVEform:APOSition <NR3>
DISplay:SPECView<x>:CURSor:CURSOR:WAVEform:APOSition?
```

Arguments `SPECView<x>` specifies the Spectrum View number. <x> is always 1.

`<NR3>` specifies the horizontal position of Spectrum View Cursor A in Hz. The range of values is the frequencies at the left and right edges of the Spectrum View for the cursor source

Examples `DISplay:SPECView1:CURSor:CURSOR:WAVEform:APOSition 22E+6` sets the Spectrum View Cursor B position to 22 MHz.

`DISplay:SPECView1:CURSor:CURSOR:WAVEform:APOSition?` might return `DISplay:SPECView1:CURSor:CURSOR:WAVEform:APOSition 48.5E+6`, indicating the Spectrum View Cursor A position is 48.5 MHz.

DISplay:SPECView<x>:CURSor:CURSOR:WAVEform:BPOSition

This command sets or queries the Cursor B horizontal position in the Spectrum View.

Group Cursor

Syntax
```
DISplay:SPECView<x>:CURSor:CURSOR:WAVEform:BPOSition <NR3>
DISplay:SPECView<x>:CURSor:CURSOR:WAVEform:BPOSition?
```

Arguments `SPECView<x>` specifies the Spectrum View number. <x> is always 1.

`<NR3>` specifies the horizontal position of Spectrum View Cursor B in Hz. The range of values is the frequencies at the left and right edges of the Spectrum View for the cursor source

Examples `DISplay:SPECView1:CURSor:CURSOR:WAVEform:BPOSition 22E+6` sets the Spectrum View Cursor B position to 22 MHz.

`DISplay:SPECView1:CURSor:CURSOR:WAVEform:BPOSition?` might return `DISplay:SPECView1:CURSor:CURSOR:WAVEform:BPOSition 48.5E+6`, indicating the Spectrum View Cursor B position is 48.5 MHz.

DISplay:SPECView<x>:HORZ

This command sets or queries the horizontal display scaling of the Spectrum View window.

Group Display

Syntax
```
DISplay:SPECView<x>:HORZ {LOG|LINEAr}
DISplay:SPECView<x>:HORZ?
```

Arguments

SPECView<x> specifies the Spectrum View number. <x> is always 1.

LINEAr sets the Spectrum View horizontal scaling to linear.

LOG sets the Spectrum View horizontal scaling to log.

Examples

DISplay:SPECView1:HORZ LOG sets the horizontal display scaling in the Spectrum View window to log.

DISplay:SPECView1:HORZ? might return :DISplay:SPECView1:HORZ LOG, indicating that the horizontal display scaling in the Spectrum View window is set to log.

DISplay:SPECView<x>:GRAticule

This command sets or queries the type of graticule shown in the Spectrum View window. The same grid is used for all spectrum traces.

Group

Display

Syntax

```
DISplay:SPECView<x>:GRAticule {GRId|TIMe|FULl|NONe}
DISplay:SPECView<x>:GRAticule?
```

Arguments

SPECView<x> specifies the Spectrum View number. <x> is always 1.

GRId specifies to show the Grid graticule (frame and grid) in the Spectrum View window.

TIMe specifies to show the Time graticule in the Spectrum View window.

FULl specifies to show the Full graticule (frame, a grid and cross hairs) in the Spectrum View window.

NONe specifies to show no graticule in the Spectrum View window.

Examples

DISplay:SPECView1:GRAticule TIME sets the Spectrum View window to show a Time graticule.

DISplay:SPECView1:GRAticule? might return GRID, indicating that the Spectrum View is showing a Grid graticule.

DISplay:SPECView<x>:INTENSITy:GRATicule

This command sets or queries the graticule saturation (brightness) level in the Spectrum View window.

NOTE. This command also changes the intensity of the vertical scale values on the graticule.

Group Display

Syntax
```
DISplay:SPECView<x>:INTENSITy:GRATicule <NR2>
DISplay:SPECView<x>:INTENSITy:GRATicule?
```

Arguments SPECView<x> specifies the Spectrum View number. <x> is always 1.

<NR2> specifies the Spectrum View graticule intensity lever as a percent between 0% and 100%.

Examples DISplay:SPECView1:INTENSITy:GRATicule 75 sets the Spectrum View graticule saturation level to 75%.

DISplay:SPECView1:INTENSITy:GRATicule? might return 66.0, indicating the Spectrum View graticule saturation level is at 66%.

DISplay:SPECView<x>:INTENSITy:WAVEform

This command sets or queries the spectrum trace saturation (brightness) level in the Spectrum View window.

Group Display

Syntax
```
DISplay:SPECView<x>:INTENSITy:WAVEform <NR2>
DISplay:SPECView<x>:INTENSITy:WAVEform?
```

Arguments SPECView<x> specifies the Spectrum View number. <x> is always 1.

<NR2> specifies the spectrum trace intensity lever as a percent between 1% and 100%.

Examples DISplay:SPECView1:INTENSITy:WAVEform 75 sets the spectrum trace saturation level to 75%.

DISplay:SPECView1:INTENSITy:WAVEform? might return 80.0, indicating the spectrum trace saturation level is at 80%.

DISplay:SPECView<x>:VIEWStyle

This command sets or queries the spectrum trace layout style used by the Spectrum View window.

Group Display

Syntax

```
DISplay:SPECView<x>:VIEWStyle {OVErlay|STAcked}
DISplay:SPECView<x>:VIEWStyle?
```

Arguments

`SPECView<x>` specifies the Spectrum View number. <x> is always 1.

`OVErlay` specifies the Spectrum View window to show all spectrum traces drawn on top of each other.

`STAcked` specifies the Spectrum View window to show each spectrum trace in a separate horizontal slice.

Examples

`DISplay:SPECView1:VIEWStyle OVErlay` sets the view style to overlay.

`DISplay:SPECView1:VIEWStyle?` might return `STACKED`, indicating the Spectrum View window is showing spectrum traces in stacked mode.

DISplay:VARpersist

This command sets or queries display persistence decay time, which is the approximate decay time for a freshly struck persistence sample.

Group Display Control

Syntax

```
DISplay:VARpersist <NR3>
DISplay:VARpersist?
```

Related Commands DISplay:PERSistence

Arguments

`<NR3>` indicates the persistence decay time and ranges from 0.5 to 100.

Examples

`DISPLAY:VARPERSIST 5` sets the persistence decay time to 5.

`DISPLAY:VARPERSIST?` might return `:DISPLAY:VARPERSIST 3.0000E-01`, indicating that persistence decay time is currently set to 0.300.

DISplay:WAVEform

This command globally enables or disables the waveform display. When disabled, the waveform is still acquired and held in memory, but it is not drawn to the screen. Disabling the waveform display may improve processing speed.

Group Cursor

Syntax

```
DISplay:WAVEform {<NR1>|OFF|ON}
DISplay:WAVEform?
```

Arguments

<NR1> enables or disables the waveform display. 0 disables the waveform display; any other value enables the waveform display.

ON enables the waveform display.

OFF disables the waveform display.

Examples

DISplay:WAVEform OFF disables the waveform display.

DISplay:WAVEform? might return DISplay:WAVEform 1, indicating that the waveform display is enabled.

DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION

This command sets or queries the location to display the Waveform View cursor readouts (in the Waveform View graticule or in a badge in the Results Bar).

Group Cursor

Syntax

```
DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION {GRATICULE|BADGE}
DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION?
```

Arguments

GRATICULE sets the Waveform View cursor readouts to display as part of the cursors in the plot view.

BADGE removes the Waveform View cursor readouts from the cursors in the graticule and displays the cursor information as a badge in the Results Bar.

Examples

DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION GRATICULE sets the cursor readout information for the Waveform View to display in the Waveform View graticule.

DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION? might return DISplay:WAVEView:CURSor:CURSOR1:ROLOCATION BADGE, indicating that cursor readout information for the Waveform View is set to display as a badge in the Results Bar.

DISplay:WAVEView:GRIDTYPE

This command sets or queries the Waveform View Graticule type.

Group Display

Syntax

```
DISplay:WAVEView:GRIDTYPE {MOVEABLE|FIXED}
DISplay:WAVEView:GRIDTYPE?
```

Arguments MOVEABLE sets the Waveform View so that both the waveform and the grid (graticule) move together when moving the waveform horizontally

FIXED sets the Waveform View so that the grid dows not move when moving the waveform horizontally.

Examples DISplay:WAVEView:GRIDTYPE MOVEABLE sets the Waveform View to enable both the graticule and the waveform to move together.

DISplay:WAVEView:GRIDTYPE? might return FIXED, indicating that the graticule will not change positions if the waveform moves horizontally.

DISplay:WAVEView<x>:BUS:B<x>:STATE

Sets or queries the state of the specified bus in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax

```
DISplay:WAVEView<x>:BUS:B<x>:STATE {OFF|ON|0|1}
DISplay:WAVEView<x>:BUS:B<x>:STATE?
```

Arguments 0 turns specified bus off.

1 turns the specified bus on.

ON turns the specified bus on.

OFF turns specified bus off.

Examples DISplay:WAVEView1:BUS:B1:STATE OFF turns specified bus off.

DISplay:WAVEView1:BUS:B1:STATE? might return :DISPLAY:WAVEVIEW1:BUS:B1:STATE 1 indicating the specified bus is on.

DISplay:WAVEView<x>:BUS:B<x>:VERTical:POSition

Sets or queries the vertical position of the specified bus in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax
```
DISplay:WAVEView<x>:BUS:B<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:BUS:B<x>:VERTical:POSition?
```

Arguments <NR3> is the vertical position of the specified bus.

Examples DISplay:WAVEView1:BUS:B1:VERTical:POSition 1.0e0 sets the position to 1 division.

DISplay:WAVEView1:BUS:B1:VERTical:POSition? might return :DISPLAY:WAVEVIEW1:BUS:B1:VERTICAL:POSITION 0.0E+0 indicating the position is 0.0 divisions.

DISplay:WAVEView<x>:CH<x>:STATE

Sets or queries the state of the specified channel in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax
```
DISplay:WAVEView<x>:CH<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:CH<x>:STATE?
```

Arguments

<NR1> = 0 disables the specified channel on the specified Waveform View; any other value turns this feature on.

OFF disables the display the specified channel on the specified Waveform View.

ON enables the specified channel on the specified Waveform View.

Examples

DISplay:WAVEView1:CH1:STATE ON turns on channel 1.

DISplay:WAVEView1:CH1:STATE? might return :DISPLAY:WAVEVIEW1:CH1:STATE 0 indicating channel 1 is off.

DISplay:WAVEView<x>:CH<x>:VERTical:POSition

Sets or queries the vertical position of the specified channel in the specified Waveform View in divisions. 0.0 divisions is center, 5.0 top of the window, and -5.0 the bottom of the window.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:CH<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:CH<x>:VERTical:POSition?
```

Arguments

<NR3> is the vertical position in divisions. 0.0 divisions is center, 5.0 top of the window, and -5.0 the bottom of the window.

Examples

DISplay:WAVEView1:CH1:VERTical:POSition 1.0e0 sets the position to 1 division.

DISplay:WAVEView1:CH1:VERTical:POSition? might return :DISPLAY:WAVEVIEW1:CH1:VERTICAL:POSITION -3.4400 indicating the channel 1 vertical position is -3.44 divisions.

DISplay:WAVEView<x>:CH<x>:VERTical:SCAle

Sets or queries the vertical scale of the specified channel in volts per division within the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax
```
DISplay:WAVEView<x>:CH<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:CH<x>:VERTical:SCAle?
```

Arguments <NR3> is the vertical scale of the specified channel.

Examples DISplay:WAVEView1:CH1:VERTical:SCAle 2.0e2 sets the vertical scale to 200 mV per division.

DISplay:WAVEView1:CH1:VERTical:SCAle? might return :DISPLAY:WAVEVIEW1:CH1:VERTICAL:SCALE 200.0000E-3 indicating the scale is 200 mV per division.

DISplay:WAVEView<x>:CH<x>_DALL:STATE

This command sets or queries the display state of the specified digital channel in the specified Waveform View.

Group Display Control

Syntax
```
DISplay:WAVEView<x>:CH<x>_DALL:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:CH<x>_DALL:STATE?
```

Arguments <NR1> = 0 disables the display of the specified channels on the specified Waveform View; any other value turns this feature on.

OFF disables the display of the specified channels on the specified Waveform View.

ON enables the display of the specified channels on the specified Waveform View.

Examples DISplay:WAVEView1:CH1_DALL:STATE ON enables the display of the specified channels on the specified Waveform View.

DISplay:WAVEView1:CH1_DALL:STATE? might return :DISPLAY:WAVEVIEW1:CH1_DALL:STATE 0 indicating specified digital channels in the specified Waveform View are off.

DISplay:WAVEView<x>:CH<x>_DALL:VERTical:POSition

This command sets or queries the vertical position of the specified digital channel in the specified Waveform View in divisions. The position ranges from 5.0 to -5.0 divisions.

Group Display Control

Syntax

```
DISplay:WAVEView<x>:CH<x>_DALL:VERTical:POSition <NR3>
DISplay:WAVEView<x>:CH<x>_DALL:VERTical:POSition?
```

Arguments <NR3> is the vertical position of the specified digital channel in the specified Waveform View in divisions.

Examples `DISplay:WAVEView1:CH1_DALL:VERTical:POSition 1.0e0` sets the vertical position to 1.0 divisions.

`DISplay:WAVEView1:CH1_DALL:VERTical:POSition?` might return `:DISPLAY:WAVEVIEW1:CH1_DALL:VERTICAL:POSITION 0.0E+0` indicating the vertical position is 0.0 divisions.

DISplay:WAVEView<x>:CH<x>_D<x>:STATE

This command sets or queries the display state of the specified digital channel in the specified Waveform View.

Group Display Control

Syntax

```
DISplay:WAVEView<x>:CH<x>_D<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:CH<x>_D<x>:STATE?
```

Arguments <NR1> = 0 disables the display of the specified channel on the specified Waveform View; any other value turns this feature on.

OFF disables the display of the specified channel on the specified Waveform View.

ON enables the display of the specified channel on the specified Waveform View.

Examples `DISplay:WAVEView1:CH1_D1:STATE OFF` turns off the specified digital channel.

DISplay:WAVEView1:CH1_D1:STATE? might return :DISPLAY:WAVEVIEW1:CH1_D1:STATE 1 indicating the specified digital channel is on.

DISplay:WAVEView<x>:CURSor? (Query Only)

This query returns the cursor parameters for the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group Cursor

Syntax DISplay:WAVEView<x>:CURSor?

Returns Returns the cursor parameters for the specified Waveform View.

Examples DISplay:WAVEView1:CURSor? might return
WAVEFORM;AUTO;2.4993299784074;2.4958885115371;
-3.4414668702805E-3;"A";"A";300.0E-6;2.2369669999970;
256.967000E-3;-300.0E-6;-300.0E-6;300.0E-6; "????
????";"s";"???? ????";600.0E-6;1.6666666666667E+3;
1;300.0E-6;-300.0E-6;SAME;4.1655499640123E+3;INDEPENDENT;AUTO

DISplay:WAVEView<x>:CURSor:CURSOR<x>? (Query Only)

This query returns the cursor parameters for the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor

Syntax DISplay:WAVEView<x>:CURSor:CURSOR<x>?

Returns Returns the cursor parameters for the specified cursor in the specified Waveform View.

Examples DISplay:WAVEView1:CURSor:CURSOR1? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR:WAVEFORM:BPOSITION 5.0E-6;APOSITION -5.0E-6;:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:BSOURCE AUTO;HBARS:BPOSITION 9.91E+37;BUNITS "V";AUNITS "V";APOSITION 9.91E+37;DELTA 9.91E+37;:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:VBARS:BPOSITION 5.0E-6;APOSITION -5.0E-6;UNITS "s";DELTA 10.0E-6;ALTERNATEB "???? ????";ALTERNATEA "???? ????";:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:STATE 1;FUNCTION WAVEFORM;SCREEN:AYPOSITION -2.0E-3;BXPOSITION -5.0E-6;BYPOSITION -2.0E-3;AXPOSITIO N -5.0E-6;:DISPLAY:WAVEVIEW1:CURSOR:CURSOR:MODE INDEPENDENT;ASOURCE AUTO;DDT 9.91E+37;ONEOVERDELTATVALUE 100.0E+3;SPLITM ODE SAME;LINESTYLE SOLID.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:ASOUrce

This command sets or queries the cursor A source of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.*

Group Cursor

Syntax
```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:ASOUrce
{AUTO|CH<x>|BUS<x>|MATH<x>|REF<x>|PLOT<x>}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:ASOUrce?
```

Arguments Arguments are the specified source waveform.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:ASOUrce CH1` sets the cursor 1 source to channel 1.

`DISplay:WAVEView1:CURSor:CURSOR1:ASOUrce?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:ASOURCE REF1` indicating the cursor A source is reference 1.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:BSOUrce

This command sets or queries the cursor B source of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:BSOUrce
{CH<x>|BUS<x>|MATH<x>|REF<x>|PLOT<x>}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:BSOUrce?
```

Arguments Arguments are the specified source waveform.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:BSOUrce CH1` sets the cursor B source to channel 1.

`DISplay:WAVEView1:CURSor:CURSOR1:BSOUrce?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:BSOURCE REF1` indicating the cursor b source is reference 1.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:DDT? (Query Only)

This query returns the delta V over delta T cursor readout value of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:DDT?
```

Returns The delta V over delta T cursor readout value of the specified cursor in the specified Waveform View.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:DDT?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:DDT 9.91E+37` indicating the delta V over delta T cursor readout value of the specified cursor is 9.91E+37.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCtion

This command sets or queries the cursor type of the specified cursor in the specified Waveform View.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.*

Group Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCtion
{SCREEN|WAVEFORM|VBArs|HBArs}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:FUNCtion?
```

Arguments HBArs specifies horizontal bar cursors, which measure in vertical units.

VBArs specifies vertical bar cursors, which measure in horizontal units.

SCREEN specifies both horizontal and vertical bar cursors, which measure in horizontal and vertical units specified by the Cursor 1 and Cursor 2 Sources. Use these cursors to measure anywhere in the waveform display area.

WAVEform specifies paired or split cursors in YT display format for measuring waveform amplitude and time. In XY and XYZ format, these cursors indicate the amplitude positions of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.

Examples `DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION VBArs` selects the vertical bar cursor type for the specified Waveform View and cursor.

`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:FUNCTION HBArs` indicating that the specified cursor is set to HBArs.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition

Sets or queries the HBARs vertical A position of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.*

Group Cursor

Syntax
```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:APOSition?
```

Arguments <NR3> is the vertical cursor A position of the specified cursor in the specified Waveform View. 0.0 divisions is center, 5.0 top of the waveview, and -5.0 the bottom of the waveview.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:HBArs:APOSition 3.0e0` sets the cursor position to 3 V.

`DISplay:WAVEView1:CURSor:CURSOR1:HBArs:APOSition?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:APOSITION 2.9303448275862` indicating the cursor position is 2.93 V.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:AUNIts? (Query Only)

This command queries the cursor A vertical units of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.*

Group Cursor

Syntax
```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:AUNIts?
```

Arguments <QString> is the cursor A vertical units of the specified cursor in the specified Waveform View.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:HBArs:AUNIts?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:AUNITS "V"` indicating the `AUNITS` are set to Volts.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition

Sets or queries the HBARs vertical B position of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BPOSition?
```

Arguments <NR3> is the vertical cursor B position of the specified cursor in the specified Waveform View.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:HBArs:BPOSition 4.0e-1` sets the position to 400 mV.

`DISplay:WAVEView1:CURSor:CURSOR1:HBArs:BPOSition?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:BPOSITION 387.9148706896567E-3` indicating the B cursor position is 387.9 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BUNIts? (Query Only)

This command queries the cursor B vertical units of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:BUNIts?
```

Returns <QString> is the cursor B vertical units of the specified cursor in the specified Waveform View.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:HBArs:BUNIts?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:BUNITS "V"` indicating the `BUNITS` are set to Volts.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:DELTa? (Query Only)

This command queries the delta V cursor readout value of the specified cursor in the specified Waveform View.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.*

Group Cursor

Syntax `DISplay:WAVEView<x>:CURSor:CURSOR<x>:HBArs:DELTa?`

Returns The delta V cursor readout value of the specified cursor in the specified Waveform View.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:HBArs:DELTa?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:HBARS:DELTA 2.612085129310` indicating the delta between the cursors is 2.61 V.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:MODe

Sets or queries the cursor tracking mode of the specified cursor in the specified Waveform View.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.*

Group Cursor

Syntax `DISplay:WAVEView<x>:CURSor:CURSOR<x>:MODe {INDEPENDENT|TRACK}`

Arguments `TRACK` ties the navigational functionality of the two cursors together. For cursor 1 adjustments, this ties the movement of the two cursors together; however, cursor 2 continues to move independently of cursor 1.

INDEPENDENT allows independent adjustment of the two cursors.

Examples DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:MODE TRACK specifies that the cursor positions move in unison.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:MODE? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:MODE TRACK, indicating that the two cursors move in unison.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:ONEOVERDELTATVALUE? (Query Only)

This query returns the one over delta T cursor readout value of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.*

Group Cursor

Syntax DISplay:WAVEView<x>:CURSor:CURSOR<x>:ONEOVERDELTATVALUE?

Returns The one over delta T cursor readout value of the specified cursor in the specified Waveform View.

Examples DISplay:WAVEView1:CURSor:CURSOR1:ONEOVERDELTATVALUE? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:ONEOVERDELTATVALUE 179.3188E+3 indicating the one over delta time value is 179.3 kHz.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AXPOSition

Sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.*

Group Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AXPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AXPOSition?
```

Arguments

<NR3> is the horizontal cursor A position of the specified cursor in the specified Waveform View.

Examples

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION 15.0E-3 sets the x position of screen cursor1 in waveview1 to 15 mV.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION -64.0000E-03 indicating that x position of the specified screen cursor in the specified Waveform View is set to -64 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition

This command sets or queries the vertical cursor A position of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.*

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:AYPOSition?
```

Arguments

<NR3> the vertical cursor A position of the specified cursor in the specified Waveform View.

Examples

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AYPOSITION 25.0E-3 sets the y position of the specified screen cursor of the specified Waveform View to 25 mV.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AYPOSITION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AYPOSITION -53.0000E-03 indicating that y position of the specified screen cursor of the specified Waveform View is set to -53 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition

Sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor

Syntax
```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BXPOSition?
```

Arguments <NR3> is the horizontal cursor B position of the specified cursor in the specified Waveform View.

Examples DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION 15.0E-3 sets the x position of the specified screen cursor in the specified Waveform View to 15 mV.

DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:AXPOSITION -64.0000E-03 indicating that x position of the specified screen cursor in the specified Waveform View is set to -64 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSition

This command sets or queries the vertical cursor B position of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor

Syntax
```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SCREEN:BYPOSition?
```

Arguments <NR3> the vertical cursor B position of the specified cursor in the specified Waveform View.

Examples `DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION 25.0E-3` sets the y position of the screen cursor of waveview1 to 25 mV.

`DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SCREEN:BYPOSITION -53.0000E-03` indicating that y position of the specified screen cursor of the specified Waveform View is set to -53 mV.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE

This command sets or queries whether both cursors have the same or different sources.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.*

Group Cursor

Syntax
```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE {SAME|SPLIT}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:SPLITMODE?
```

Arguments `SAME` specifies both cursors have the same source.

`SPLIT` specifies the cursors have different sources.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:SPLITMODE SPLIT` specifies the cursors have different sources.

`DISplay:WAVEView1:CURSor:CURSOR1:SPLITMODE?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:SPLITMODE SAME` indicating the cursors have the same source.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE

This command sets or queries the visible state of the specified cursor in the specified Waveform View.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.*

Group Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:CURSor:CURSOR<x>:STATE?
```

Arguments

<NR1> = 0 disables the specified cursor in the specified Waveform View; any other value turns this feature on.

OFF disables the specified cursor in the specified Waveform View.

ON enables the specified cursor in the specified Waveform View.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:STATE OFF turns the specified cursor off.

DISplay:WAVEView1:CURSor:CURSOR1:STATE? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:STATE 1 indicating the specified cursor in the specified Waveform View is on.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:APOSition

This command sets or queries the cursor A horizontal position of the specified cursor in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.*

Group

Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:APOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:APOSition?
```

Arguments

<NR3> is the horizontal cursor A position of the specified cursor in the specified Waveform View.

Examples

DISplay:WAVEView1:CURSor:CURSOR1:VBArs:APOSition -4.0e-6 set the cursor position to 4.0 µs.

DISplay:WAVEView1:CURSor:CURSOR1:VBArs:APOSition? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:APOSITION -5.0E-6 indicating the cursor A position is -5.0 µs.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:BPOSition

This command sets or queries the cursor B horizontal position of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:BPOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:BPOSition?
```

Arguments <NR3> is the horizontal cursor B position of the specified cursor in the specified Waveform View.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:VBArs:BPOSition 4.0e-6` sets the cursor position to 4.0 µs.

`DISplay:WAVEView1:CURSor:CURSOR1:VBArs:BPOSition?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:BPOSITION 3.2736951319481E-6` indicating the B cursor position is 3.27 µs.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:DELTa? (Query Only)

This query sets or returns the delta T cursor readout value of the specified cursor in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. Cursor<x> is the specified cursor and must be CURSOR1.

Group Cursor

Syntax

```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:DELTa?
```

Returns The delta T cursor readout value of the specified cursor in the specified Waveform View.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:VBArs:DELTa?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:DELTA 6.7926388747725E-6` indicating the delta T cursor readout value is 6.79 µs.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:UNIts? (Query Only)

This query returns cursor A vertical units of the specified cursor in the specified Waveform View.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.*

Group Cursor

Syntax `DISplay:WAVEView<x>:CURSor:CURSOR<x>:VBArs:UNIts?`

Returns The cursor A vertical units of the specified cursor in the specified Waveform View.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:VBArs:UNIts?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:VBARS:UNITS "s"` indicating the cursor units are seconds.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:APOSition

This command sets or queries the horizontal cursor A position of the specified cursor in the specified Waveform View.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.*

Group Cursor

Syntax
```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:APOSition <NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:APOSition?
```

Arguments `<NR3>` is the horizontal cursor A position of the specified cursor in the specified Waveform View.

Examples `DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:APOSition 1.5e-6` sets the position to 1.5 µs.

`DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:APOSition?` might return `:DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:WAVEFORM:APOSITION -1.2667480236557E-6` indicating the position is 1.27 µs.

DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:AVPOSition? (Query Only)

This command queries the vertical waveform value at the cursor A position in the specified Waveform View.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. .*

Group Cursor

Syntax `DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:AVPOSition?`

Arguments `<NR3>` is the horizontal cursor A position of the specified cursor in the specified Waveform View.

Examples `DISplay:WAVEView1:CURSor:CURSOR:WAVEform:AVPOSition?` might return `:DISplay:WAVEView1:CURSor:CURSOR:WAVEform:AVPOSition 1.35`, indicating that the vertical waveform value at the Cursor A position is 1.35 V.

DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition

This command sets or queries the horizontal cursor B position of the specified cursor in the specified Waveform View.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `Cursor<x>` is the specified cursor and must be `CURSOR1`.*

Group Cursor

Syntax
```
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition
<NR3>
DISplay:WAVEView<x>:CURSor:CURSOR<x>:WAVEform:BPOSition?
```

Arguments <NR3> is the horizontal cursor B position of the specified cursor in the specified Waveform View.

Examples DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:BPOSition 8.0e-9 sets the position to 8.0 ns.

DISplay:WAVEView1:CURSor:CURSOR1:WAVEform:BPOSition? might return :DISPLAY:WAVEVIEW1:CURSOR:CURSOR1:WAVEFORM:BPOSITION 711.6310718892624E-9 indicating the cursor position is 711.6 ns.

DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:BVPOSition? (Query Only)

This command queries the vertical waveform value at the cursor B position in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. .*

Group Cursor

Syntax DISplay:WAVEView<x>:CURSor:CURSOR:WAVEform:BVPOSition?

Arguments <NR3> is the horizontal cursor A position of the specified cursor in the specified Waveform View.

Examples DISplay:WAVEView1:CURSor:CURSOR:WAVEform:BVPOSition? might return :DISplay:WAVEView1:CURSor:CURSOR:WAVEform:BVPOSition -0.72, indicating that the vertical waveform value at the Cursor B position is -0.72 V.

DISplay:WAVEView<x>:FILTer

This command sets or queries the type of interpolation filter for the display.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax DISplay:WAVEView<x>:FILTer {SINX|LINear}

Arguments

`LINEAr` specifies linear interpolation, where acquired points are connected with straight lines.

`SINX` specifies sin(x)/x interpolation, where acquired points are fit to a curve.

Examples

`DISplay:WAVEView1:FILTer SINX` specifies sine-curve interpolation, when magnifying waveforms.

`DISplay:WAVEView1:FILTer?` might return `:DISplay:WAVEView1:FILTer LINEAR`, indicating that straight-line interpolation is specified for magnifying waveforms.

DISplay:WAVEView<x>:GRAticule

This command selects or queries the type of graticule that is displayed.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`.*

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:GRAticule {GRId|TIMe|FULl|NONe}
DISplay:WAVEView<x>:GRAticule?
```

Arguments

`GRId` specifies a frame and grid only.

`TIMe` specifies a time graticule only.

`FULl` specifies a frame, a grid and cross hairs.

`NONe` specified no graticule.

Examples

`DISplay:WAVEView1:GRAticule TIME` specifies a time graticule.

`DISplay:WAVEView1:GRAticule?` might return `:DISPLAY:WAVEVIEW1:GRATICULE GRID` indicating the graticule is a grid.

DISplay:WAVEView<x>:INTENSITy:GRATicule

This command sets or queries the graticule saturation level.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`.*

Group Display Control

Syntax
```
DISplay:WAVEView<x>:INTENSITy:GRATicule <NR2>
DISplay:WAVEView<x>:INTENSITy:GRATicule?
```

Arguments <NR2> is the graticule saturation level.

Examples DISplay:WAVEView1:INTENSITy:GRATicule 75 sets the saturation level to 75%.

DISplay:WAVEView1:INTENSITy:GRATicule? might return :DISPLAY:WAVEVIEW1:INTENSITY:GRATICULE 66.0000 indicating the saturation level is at 66%.

DISplay:WAVEView<x>:INTENSITy:WAVEform

This command sets or queries the waveform saturation level.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group Display Control

Syntax
```
DISplay:WAVEView<x>:INTENSITy:WAVEform <NR2>
DISplay:WAVEView<x>:INTENSITy:WAVEform?
```

Arguments <NR2> is the waveform saturation level.

Examples DISplay:WAVEView1:INTENSITy:WAVEform 75 sets the saturation level to 75%.

DISplay:WAVEView1:INTENSITy:WAVEform? might return :DISPLAY:WAVEVIEW1:INTENSITY:WAVEFORM 62.0000 indicating the saturation level is 62%.

DISplay:WAVEView<x>:MATH:MATH<x>:AUTOScale

This command sets or queries whether the specified math gets auto-scaled when the math equation changes within the specified Waveform View.

Group Display Control

Syntax
```
DISplay:WAVEView<x>:MATH:MATH<x>:AUTOScale {<NR1>|OFF|ON}
DISplay:WAVEView<x>:MATH:MATH<x>:AUTOScale?
```

Arguments <NR1> = 0 disables the autoscaling the math in the specified Waveform View; any other value turns this feature on.

OFF disables the autoscaling the math in the specified Waveform View.

ON enables the autoscaling the math in the specified Waveform View.

Examples DISplay:WAVEView1:MATH:MATH3:AUTOScale 1 enables the autoscaling the math in the specified Waveform View.

DISplay:WAVEView1:MATH:MATH3:AUTOScale? might return :DISPLAY:WAVEVIEW1:MATH:MATH3:AUTOSCALE 0 indicating that Math 3 will not auto-scale.

DISplay:WAVEView<x>:MATH:MATH<x>:STATE

This command sets or queries the state of the specified math waveform in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax
```
DISplay:WAVEView<x>:MATH:MATH<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:MATH:MATH<x>:STATE?
```

Arguments <NR1> = 0 disables the specified math in the specified Waveform View; any other value turns this feature on.

OFF disables the specified math in the specified Waveform View.

ON enables the specified math in the specified Waveform View.

Examples DISplay:WAVEView1:MATH:MATH1:STATE OFF disables the specified math in the specified Waveform View.

DISplay:WAVEView1:MATH:MATH2:STATE? might return :DISPLAY:WAVEVIEW1:MATH:MATH2:STATE 1 indicating the math is displayed.

DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition

This command sets or queries the vertical position in divisions of the specified math waveform in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group Display Control

Syntax

```
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:POSition?
```

Arguments <NR3> is the vertical position in divisions of the specified math waveform.

Examples DISplay:WAVEView1:MATH:MATH1:VERTical:POSition 0 sets the position to 0 divisions.

DISplay:WAVEView1:MATH:MATH1:VERTical:POSition? might return :DISPLAY:WAVEVIEW1:MATH:MATH1:VERTICAL:POSITION -2.2800 indicating the position is at -2.28 divisions.

DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle

Sets or queries the vertical scale of the specified math in volts per division within the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1.

Group Display Control

Syntax

```
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:MATH:MATH<x>:VERTical:SCAle?
```

Arguments <NR3> is the vertical scale of the specified math waveform.

Examples `DISplay:WAVEView1:MATH:MATH1:VERTical:SCAle 400e-3` sets the scale to 400 mV per division.

`DISplay:WAVEView1:MATH:MATH1:VERTical:SCAle?` might return `:DISPLAY:WAVEVIEW1:MATH:MATH1:VERTICAL:SCALE 395.0000E-3` indicating the scale is 395 mV.

DISplay:WAVEView<x>:PLOT:PLOT<x>:AUTOScale

This command sets or queries whether the specified trend gets auto-scaled when the new data is available within the specified Waveform View.

Group Display Control

Syntax
```
DISplay:WAVEView<x>:PLOT:PLOT<x>:AUTOScale {<NR1>|OFF|ON}
DISplay:WAVEView<x>:PLOT:PLOT<x>:AUTOScale?
```

Arguments `<NR1>` = 0 disables auto-scaling the specified plot in the specified Waveform View; any other value turns this feature on.

`OFF` disables auto-scaling the specified plot in the specified Waveform View.

`ON` enables auto-scaling the specified plot in the specified Waveform View.

Examples `DISplay:WAVEView1:PLOT:PLOT1:AUTOScale 1` enables auto-scaling the specified plot in the specified Waveform View.

`DISplay:WAVEView1:PLOT:PLOT1:AUTOScale?` might return `:DISPLAY:WAVEVIEW1:PLOT:PLOT1:AUTOSCALE 0` indicating that auto-scaling for plot 1 is off.

DISplay:WAVEView<x>:PLOT:PLOT<x>:STATE

This command sets or queries the state of the specified time trend plot waveform in the specified Waveform View.

Group Display Control

Syntax
```
DISplay:WAVEView<x>:PLOT:PLOT<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:PLOT:PLOT<x>:STATE?
```

Arguments <NR1> = 0 disables the specified plot in the specified Waveform View; any other value turns this feature on.

OFF disables the specified plot in the specified Waveform View.

ON enables the specified plot in the specified Waveform View.

Examples DISplay:WAVEView1:PLOT:PLOT1:STATE OFF disables the specified plot in the specified Waveform View.

DISplay:WAVEView1:PLOT:PLOT1:STATE? might return :DISPLAY:WAVEVIEW1:PLOT:PLOT1:STATE 1 indicating the specified plot in the specified Waveform View is on.

DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSition

This command sets or queries the vertical position of the specified time trend in the specified Waveform View in absolute units.

Group Display Control

Syntax
```
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:POSition?
```

Arguments <NR3> is the vertical position.

Examples DISplay:WAVEView1:PLOT:PLOT1:VERTical:POSition 2.0e0 sets the position to 2.0 units.

DISplay:WAVEView1:PLOT:PLOT1:VERTical:POSition? might return :DISPLAY:WAVEVIEW1:PLOT:PLOT1:VERTICAL:POSITION 1.9918 indicating the position is 1.99 units.

DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified time trend in units per division in the specified Waveform View.

Group Display Control

Syntax
```
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:SCAle?
```

Arguments <NR3> is the vertical scale.

Examples DISplay:WAVEView1:PLOT:PLOT1:VERTical:SCAle 725.0e-3 sets the scale to 725 mUnits.

DISplay:WAVEView1:PLOT:PLOT1:VERTical:SCAle? might return :DISPLAY:WAVEVIEW1:PLOT:PLOT1:VERTICAL:SCALE 730.0000E-3 indicating the scale is 730.0 mUnits.

DISplay:WAVEView<x>:REF:REF<x>:STATE

This command sets or queries the state of the specified reference waveform in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax
```
DISplay:WAVEView<x>:REF:REF<x>:STATE {<NR1>|OFF|ON}
DISplay:WAVEView<x>:REF:REF<x>:STATE?
```

Arguments <NR1> = 0 disables the specified reference in the specified Waveform View; any other value turns this feature on.

OFF disables the specified reference in the specified Waveform View.

ON enables the specified reference in the specified Waveform View.

Examples DISplay:WAVEView1:REF:REF1:STATE OFF disables the specified reference in the specified Waveform View.

DISplay:WAVEView1:REF:REF1:STATE? might return :DISPLAY:WAVEVIEW1:REF:REF1:STATE 1 indicating the specified reference in the specified Waveform View is on.

DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition

This command sets or queries the vertical position in divisions of the specified reference in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax
```
DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:REF:REF<x>:VERTical:POSition?
```

Arguments <NR3> is the vertical position in divisions.

Examples `DISplay:WAVEView1:REF:REF1:VERTical:POSition -2.5e0` sets the position to -2.5 divisions.

`DISplay:WAVEView1:REF:REF1:VERTical:POSition?` might return `:DISPLAY:WAVEVIEW1:REF:REF1:VERTICAL:POSITION -2.6400` indicating the position is set to -2.64 divisions.

DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified reference in volts per div within the specified Waveform View.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`.*

Group Display Control

Syntax
```
DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:REF:REF<x>:VERTical:SCAle?
```

Arguments <NR3> is the vertical scale of the specified reference waveform.

Examples `DISplay:WAVEView1:REF:REF1:VERTical:SCAle 400e-3` sets the scale to 400 mV per division.

`DISplay:WAVEView1:REF:REF1:VERTical:SCAle?` might return `:DISPLAY:WAVEVIEW1:REF:REF1:VERTICAL:SCALE 378.0000E-3` indicating the scale is 378 mV per division.

DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:POSition

This command sets or queries the vertical position of the specified Frequency vs. Time trace in the Waveform View, in divisions.

Conditions Requires option SV-RFVT

Group Display

Syntax
```
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:POSition?
```

Arguments `WAVEView<x>` is always WAVEView1.

`RF_FREQuency<x>` specifies the number of the Frequency vs. Time waveform.

`NR3` sets the vertical position in divisions. 0.0 divisions is center, 5.0 is the top of the Waveform View, and -5.0 is the top of the Waveform View.

Examples `DISplay:WAVEView1:RF_FREQuency2:VERTical:POSition 1.2` sets the vertical position of the Channel 2 Frequency vs. Time trace to 1.2 divisions.

`DISplay:WAVEView1:RF_FREQuency1:VERTical:POSition?` might return `DISplay:WAVEView1:RF_FREQuency1:VERTical:POSition -4.79`, indicating the Channel 1 Frequency vs. Time vertical position is -4.79 divisions.

DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified Frequency vs. Time trace in the Waveform View, in divisions.

Conditions Requires option SV-RFVT

Group Display

Syntax
```
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:RF_FREQuency<x>:VERTical:SCAle?
```

Arguments `WAVEView<x>` is always WAVEView1.

`RF_FREQuency<x>` specifies the number of the Frequency vs. Time waveform.

`NR3` sets the vertical scale, in Hertz per division.

Examples `DISplay:WAVEView1:RF_FREQuency3:VERTical:SCAle 5.0e6` sets the vertical scale of the Channel 3 Frequency vs. Time trace to 5.0 MHz per division.

DISplay:WAVEView1:RF_FREQuency2:VERTical:SCAle? might return DISplay:WAVEView1:RF_FREQuency2:VERTical:SCAle 2.0E3, indicating the vertical scale of the Channel 2 Frequency vs. Time trace is 2 KHz per division.

DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:POSition

This command sets or queries the vertical position of the specified Magnitude vs. Time trace in the Waveform View, in divisions.

Conditions Requires option SV-RFVT

Group Display

Syntax

```
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:POSition?
```

Arguments

WAVEView<x> is always WAVEView1.

RF_MAGnitude<x> specifies the number of the Magnitude vs. Time Frequency waveform.

NR3 sets the vertical position in divisions. 0.0 divisions is center, 5.0 is the top of the Waveform View, and -5.0 is the top of the Waveform View.

Examples

DISplay:WAVEView1:RF_MAGnitude2:VERTical:POSition 1.2 sets the vertical position of the Channel 2 Magnitude vs. Time trace to 1.2 divisions.

DISplay:WAVEView1:RF_MAGnitude1:VERTical:POSition? might return DISplay:WAVEView1:RF_MAGnitude1:VERTical:POSition -4.0, indicating the Channel 1 Magnitude vs. Time vertical position is -4.0 divisions.

DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified Magnitude vs. Time trace in the specified Waveform View in Volts, Watts, or dB per division.

Conditions Requires option SV-RFVT

Group Display

Syntax

```
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:RF_MAGnitude<x>:VERTical:SCAle?
```

Arguments

WAVEView<x> is always WAVEView1.

RF_MAGnitude<x> specifies the number of the Magnitude vs. Time Frequency waveform.

NR3 sets the vertical scale. The units for the vertical scale depend on the Magnitude Format setting. If the format is "Amplitude (linear)", then the units are Volts/div. If the format is "Power (linear)", then the units are Watts/div. If the format is "Power (log)", then the units are dB/div.

Examples

`DISplay:WAVEView1:RF_MAGnitude3:VERTical:SCAle 5.0` sets the vertical scale of the Channel 3 Magnitude vs. Time trace to 5.0 Volts per division.

`DISplay:WAVEView1:RF_MAGnitude2:VERTical:SCAle?` might return `DISplay:WAVEView1:RF_MAGnitude2:VERTical:SCAle 20.0E-3`, indicating the vertical scale of the Channel 2 Magnitude vs. Time trace is 20 mV per division.

DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:POSition

This command sets or queries the vertical position of the specified Phase vs. Time trace in the Waveform View, in divisions.

Conditions

Requires option SV-RFVT

Group

Display

Syntax

```
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:POSition?
```

Arguments

WAVEView<x> is always WAVEView1.

RF_PHASe<x> specifies the number of the Phase vs. Time Frequency waveform.

NR3 sets the vertical position in divisions. 0.0 divisions is center, 5.0 is the top of the Waveform View, and -5.0 is the top of the Waveform View.

Examples

`DISplay:WAVEView1:RF_PHASe2:VERTical:POSition 2.1` sets the vertical position of the Channel 2 Phase vs. Time trace to 2.1 divisions.

DISplay:WAVEView1:RF_PHASe1:VERTical:POSition? might return DISplay:WAVEView1:RF_PHASe1:VERTical:POSition -1.7, indicating the Channel 1 Phase vs. Time vertical position is -1.7 divisions.

DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:SCAle

This command sets or queries the vertical scale of the specified Phase vs. Time trace in the specified Waveform View in degrees per division.

Conditions Requires option SV-RFVT

Group Display

Syntax

```
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:SCAle <NR3>
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:SCAle?
```

Arguments

WAVEView<x> is always WAVEView1.

RF_MAGnitude<x> specifies the number of the Phase vs. Time Frequency waveform.

NR3 sets the vertical scale, in degrees per division.

Examples

DISplay:WAVEView1:RF_PHASe3:VERTical:SCAle 10.0 sets the vertical scale of the Channel 3 Phase vs. Time trace to 10 degrees per division.

DISplay:WAVEView1:RF_PHASe2:VERTical:SCAle? might return DISplay:WAVEView1:RF_PHASe2:VERTical:SCAle 50.0, indicating the vertical scale of the Channel 2 Phase vs. Time trace is set to 50 degrees per division.

DISplay:WAVEView<x>:STYle

This command sets or queries how the waveforms are displayed for analysis mode.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group Display Control

Syntax

```
DISplay:WAVEView<x>:STYle {VECtors|DOTsonly}
DISplay:WAVEView<x>:STYle?
```

Arguments

DOTs displays individual data points. New points immediately replace old ones.

VECtors connects adjacent data points. New points immediately replace old ones.

Examples

DISplay:WAVEView1:STYle VECTORS sets the display to connect adjacent data points.

DISplay:WAVEView1:STYle? might return :DISplay:WAVEView1:STYle DOTs, indicating that data points are not connected.

DISplay:WAVEView<x>:VIEWStyle

The command sets or queries the waveform layout style used by the display.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group

Display Control

Syntax

```
DISplay:WAVEView<x>:VIEWStyle {OVErlay|STAcked}
DISplay:WAVEView<x>:VIEWStyle?
```

Arguments

OVErlay specifies that the display view style used by the specified Waveform View is overlay.

STAcked specifies that the display view style used by the specified Waveform View is stacked.

Examples

DISplay:WAVEView1:VIEWStyle OVERLAY sets the view style to overlay.

DISplay:WAVEView1:VIEWStyle? might return :DISPLAY:WAVEVIEW1:VIEWSTYLE STACKED indicating the view style is stacked.

DISplay:WAVEView<x>:Zoom? (Query Only)

This query returns the zoom parameters of the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1.*

Group

Zoom

Syntax `DISplay:WAVEView<x>:Zoom?`

Returns Returns the zoom parameters of the specified Waveform View.

Examples `DISplay:WAVEView1:Zoom?` might return
`:DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:SCALE 2.5000;POSITION 70.0000;WINSCALE 400.0000E-9;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM 1:VERTICAL:SCALE 3.8000;POSITION 2.5951;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:STATE 0.`

DISplay:WAVEView<x>:ZOOM:ZOOM<x>? (Query Only)

This query returns the zoom parameters of the specified zoom in the specified Waveform View. <x> must be 1.

NOTE. *`WAVEView<x>` is the specified Waveform View and must be `WAVEView1`. `ZOOM<x>` is the specified zoom and must be `ZOOM1`.*

Group Zoom

Syntax `DISplay:WAVEView<x>:ZOOM:ZOOM<x>?`

Returns Returns the zoom parameters of the specified zoom in the specified Waveform View.

Examples `DISplay:WAVEView1:ZOOM:ZOOM1?` might return
`:DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:SCALE 2.5000;POSITION 70.0000;WINSCALE 400.0000E-9;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM 1:VERTICAL:SCALE 3.8000;POSITION 2.5951;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:STATE 0.`

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition

Sets or queries the horizontal zoom position (of the specified zoom in the specified Waveform View) of the zoomed waveform or zoom waveform in the display, around which the zoom waveform displays. It is freely movable around the acquisition settings (horizontal span). An acquired waveform or reference could extend off screen. The valid zoom area does not care about the waveform itself, only the user setting for acquisition.

For example, if horizontal scale is set to 1 second, position to 50, then the acquisition area will go from -5 s to +5 s. Zoom window 0 will focus on -5 s and zoom area 100 will focus on +5 s. If the instrument is stopped and the scale changed to 0.5 s, there will be data off the ends of the display. However, 0% zoom will put the user focus on -2.5 s, the lower bound of the acquisition span.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.*

Group Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:POSition?
```

Arguments <NR3> is a value from 0 to 100.00 and is the percent of the waveform that is to the left of screen center, when the zoom factor is 2× or greater.

Examples DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION 50 sets the horizontal position of the zoom box of waveview1 to 50 so that it is centered horizontally on the screen.

DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION? might return :DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:POSITION 10.0000, indicating that the horizontal position of the zoom box of waveview 1 is centered over the first major graticule division.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:SCALe

This command sets or queries the horizontal zoom factor of the specified zoom in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.*

Group Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:SCALe <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:SCALe?
```

Arguments <NR3> is the amount of expansion in the horizontal direction in 1-2-4 increments of the specified zoom in the specified Waveform View.

Examples DISplay:WAVEView1:ZOOM:ZOOM1:HORizontal:SCALe 5 sets the horizontal zoom factor of zoom1 in waveview1 to 5x.

DISplay:WAVEView1:ZOOM:ZOOM1:HORizontal:SCALe? might return :DISplay:WAVEView1:ZOOM:ZOOM1:HORizontal:SCALe 10, indicating that the zoom factor of waveview1 is set to 10x.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALe

This command sets or queries the overview window horizontal scale in the specified Waveform View.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.*

Group Zoom

Syntax
```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALe <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:HORizontal:WINSCALe?
```

Arguments <NR3> is the horizontal scale of the zoom window.

Examples DISplay:WAVEView1:ZOOM:ZOOM1HORizontal:WINSCALe? might return :DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:HORIZONTAL:WINSCALE 80.0000E-9 indicating that the horizontal scale of the overview window is 80 ns per division.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe

This command sets or queries the zoom display state of the specified zoom in the specified Waveform View. This command is equivalent to pushing the zoom button on the front panel.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.*

Group Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe {ON|OFF|<NR1>}
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:STATe?
```

Arguments

ON turns the specified zoom on.

OFF turns specified zoom off.

<NR1> = 0 disables the specified zoom; any other value enables the specified zoom.

Examples

DISplay:WAVEView1:ZOOM:ZOOM1:STATe ON turns the specified zoom on.

DISplay:WAVEView1:ZOOM:ZOOM1:STATe? might return :DISplay:WAVEView1:ZOOM:ZOOM1:STATe 1, indicating that specified zoom is on.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition

This command sets or queries the vertical position of the specified zoom in the specified Waveform View. It is freely movable within the confines of the acquired waveform. It is measured from the top to bottom of the acquisition window. The top of the zoom window is -5 * vertical zoom factor. The bottom of the zoom window is +5 * the vertical zoom factor. For a zoom of 5x, the position ranges from -25 to 25.

NOTE. *WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.*

Group

Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:POSition?
```

Arguments

NR3 is the vertical position of the specified zoom in the specified Waveform View. It is freely movable within the confines of the acquired waveform. The top of the zoom window is -5 * vertical zoom factor. The bottom of the zoom window is +5 * the vertical zoom factor. For a vertical zoom of 5x, the position ranges from -25 to 25.

Examples

DISplay:WAVEView1:ZOOM:ZOOM1:VERTical:POSition 50 sets the vertical position of the specified zoom in the specified Waveform View to 50, where the vertical zoom factor is 10x, top of screen is -50 and bottom if +50, and 0 is the vertical center.

DISplay:WAVEView1:ZOOM:ZOOM1:VERTical:POSition? might return :DISplay:WAVEView1:ZOOM:ZOOM1:VERTical:POSition 23.90000, indicating that the vertical position of the specified zoom in the specified Waveform View is set to 23.9, where the vertical zoom factor is 10x, top of screen is -50 and bottom if +50, and 0 is the vertical center.

DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALe

This command sets or queries the vertical zoom factor of the specified zoom in the specified Waveform View.

NOTE. WAVEView<x> is the specified Waveform View and must be WAVEView1. ZOOM<x> is the specified zoom and must be ZOOM1.

Group Zoom

Syntax

```
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALe <NR3>
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:VERTical:SCALe?
```

Arguments <NR3> is the amount of vertical expansion or compression. Based on the value that you entered, this command uses the nearest scale factor. Setting the vertical scale to 1 indicates unity (no zoom).

Examples DISplay:WAVEView1:ZOOM:ZOOM2:VERTical:SCALe 5 sets the vertical scale of zoom1 of waveview1 to 5×.

DISplay:WAVEView1:ZOOM:ZOOM2:VERTical:SCALe? might return :DISplay:WAVEView1:ZOOM:ZOOM2:VERTical:SCALe 2.0000, indicating that the vertical scale of zoom1 of waveview1 is 2×.

DISplay:WAVEView<y>:REF:REF<x>:FRAMe

This command sets or returns the selected frame of the specified analog ref. Each ref has a unique selected frame.

Group Display Control

Syntax

```
DISplay:WAVEView<y>:REF:REF<x>:FRAMe <NR1>
DISplay:WAVEView<y>:REF:REF<x>:FRAMe?
```

Arguments <NR1> is the selected frame of the specified analog ref.

Examples DISplay:WAVEView1:REF:REF2:FRAMe? might return :DISPLAY:WAVEVIEW1:REF:REF2:FRAME 1, indicating the selected frame is 1.

DISplay:WAVEView<y>:REF<x>_DALL:FRAMe

This command sets or returns the selected frame of the specified digital ref. Each ref has a unique selected frame.

Group Display Control

Syntax
```
DISplay:WAVEView<y>:REF<x>_DALL:FRAMe <NR1>
DISplay:WAVEView<y>:REF<x>_DALL:FRAMe?
```

Arguments <NR1> is the selected frame of the specified digital ref.

Examples DISplay:WAVEView<y>:REF<x>_DALL:FRAMe? might return :DISPLAY:WAVEVIEW1:REF1_DALL:FRAME 1, indicating the selected frame is 1.

DVM (No Query Form)

Resets the Digital Voltmeter measurements and history.

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax
```
DVM RESET
```

Arguments RESET specifies resetting DVM measurements and history.

Examples DVM RESET resets the DVM measurement and history.

DVM:AUTORange

Sets (or queries) the autorange state for the Digital Voltmeter.

NOTE. *the DVM will not autorange as long as the DVM source is the same channel as the trigger source.*

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax
```
DVM:AUTORange {0|1|OFF|ON}
DVM:AUTORange?
```

Arguments `1` or `ON` turns on autorange for the Digital Voltmeter.

`0` or `OFF` turns autorange off.

Examples `DVM:AUTOR ON` turns on autorange for the Digital Voltmeter.

DVM:MEASUrement:FREQuency? (Query Only)

This command returns the current frequency value for the DVM.

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax
```
DVM:MEASUrement:FREQuency?
```

Related Commands DVM:TRIGger:FREQuency:COUNTer

Examples `DVM:MEASU:FREQ?` might return `100.0000E+3,` which represents 100 kHz as the current frequency value for the DVM.

DVM:MEASUrement:HIStory:AVErage? (Query Only)

Returns the average DVM readout value over the history period. The history period is a constant period of 5 seconds.

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax `DVM:MEASUrement:HIStory:AVErage?`

Examples `DVM:MEASU:HIS:AVE?` might return `:DVM:MEASUREMENT:HISTORY:AVERAGE 429.3000E-3` which represents the average DVM readout value over the history period in volts.

DVM:MEASUrement:HIStory:MAXimum? (Query Only)

Returns the maximum readout value for the DVM function over the history period. The history period is a constant period of 5 seconds.

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax `DVM:MEASUrement:HIStory:MAXimum?`

Examples `DVM:MEASU:HIS:MAX?` might return `:DVM:MEASUREMENT:HISTORY:MAXIMUM 431.9000E-3`, which represents the maximum readout value for the DVM function over the history period in volts.

DVM:MEASUrement:HIStory:MINImum? (Query Only)

Returns the minimum readout value for the DVM over the history period. The history period is a constant period of 5 seconds.

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax `DVM:MEASUrement:HIStory:MINImum?`

Examples `DVM:MEASU:HIS:MINI?` might return `:DVM:MEASUREMENT:HISTORY:MINIMUM 430.9000E-3`, which represents the minimum readout value for the DVM function over the history period in volts.

DVM:MEASUrement:INFMAXimum? (Query Only)

Returns the maximum DVM readout value over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or DVM `RESET`.

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax `DVM:MEASUrement:INFMAXimum?`

Examples `DVM:MEASU:INFMAX?` might return `:DVM:MEASUREMENT:INFMAXIMUM 432.9000E-3`, which represents the maximum readout value (in volts)of the DVM function over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or `DVM RESET`.

DVM:MEASUrement:INFMINimum? (Query Only)

Returns the minimum readout value of the DVM over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or `DVM RESET`.

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax `DVM:MEASUrement:INFMINimum?`

Examples `DVM:MEASU:INMIN?` might return `:DVM:MEASUREMENT:INFMINIMUM 427.3000E-3,` which represents the minimum readout value of the DVM

function (in volts) over the entire time that the DVM has been on since the last change using the DVM:MODe or DVM:SOUrce commands or DVM RESET.

DVM:MEASUrement:VALue? (Query Only)

Returns the DVM readout value (the largest displayed value at the top of the DVM screen).

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax

```
DVM:MEASUrement:VALue?
```

Examples DVM:MEASU:VAL? might return :DVM:MEASUREMENT:VALUE 430.7000E-3, which represents the DVM value.

DVM:MODe

This command specifies (or queries) the mode to use for the Digital Voltmeter.

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax

```
DVM:MODe {ACRMS|ACDCRMS|DC|OFF}
DVM:MODe?
```

Arguments

ACRMS – displays the root-mean-square value of the acquired data, with the DC component removed.

ACDCRMS – displays the RMS value of the acquired data.

DC – displays the DC value of the acquired data.

OFF

Examples DVM:MODE DC sets the mode for the DVM to DC, which displays the DC value of the acquired data.

DVM:MOD? might return ACRMS, which indicates the mode is currently set to ACRMS.

DVM:SOUrce

This command sets (or queries) the source for the DVM.

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax
```
DVM:SOUrce {CH<x>}
DVM:SOUrce?
```

Arguments CH<x> specify which channel to use as the source for the DVM.

Examples DVM:SOURCE CH4 sets the DVM source to Channel 4.

DVM:TRIGger:FREQuency:COUNTer

This command sets or queries the state of the trigger frequency counter readout in the trigger badge.

Conditions Requires DVM option (free with product registration).

Group DVM

Syntax
```
DVM:TRIGger:FREQuency:COUNTer {0|1|OFF|ON}
DVM:TRIGger:FREQuency:COUNTer?
```

Arguments 1 or ON turns on the trigger frequency counter for the Digital Voltmeter.

0 or OFF turns it off.

Examples DVM:TRIGGER:FREQUENCY:COUNTER 0 turns off the counter.

DVM:TRIGGER:FREQUENCY:COUNTER? might return :DVM:TRIGGER:FREQUENCY:COUNTER 1 indicating the counter is on.

*ESE

This command sets and queries the bits in the Event Status Enable Register (ESER). The ESER prevents events from being reported to the Status Byte Register (STB). For a more detailed discussion of the use of these registers, see Registers.

Group Status and Error

Syntax

```
*ESE <NR1>
*ESE?
```

Related Commands *CLS

DESE

*ESR?

EVENT?

EVMsg?

*SRE

*STB?

Arguments <NR1> specifies the binary bits of the ESER according to this value, which ranges from 0 through 255.

The power-on default for the ESER is 0 if *PSC is 1. If *PSC is 0, the ESER maintains the previous power cycle value through the current power cycle.

NOTE. *Setting the DESER and the ESER to the same values allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the DESE command to set the DESER.*

Examples `*ESE 209` sets the ESER to binary 11010001, which enables the PON, URQ, EXE, and OPC bits.

`*ESE?` might return 186, showing that the ESER contains DESE the binary value 10111010.

*ESR? (Query Only)

This query-only command returns the contents of the Standard Event Status Register (SESR). *ESR? also clears the SESR (since reading the SESR clears it). For a more detailed discussion of the use of these registers, see Registers.

Group Status and Error

Syntax

```
*ESR?
```

Related Commands ALLEv?

*CLS

DESE

*ESE

EVENT?

EVMsg?

*SRE

*STB?

Examples `*ESR?` might return `*ESR 213`, showing that the SESR contains the binary value 11010101.

ETHERnet:DHCPbootp

This command sets the network configuration method to DHCP (that is ON) or static IP address (that is OFF).

Group Ethernet

Syntax

```
ETHERnet:DHCPbootp {ON|OFF}
ETHERnet:DHCPbootp?
```

Arguments `ON` enables the instrument to search the network for a DHCP server in order to automatically assign a dynamic IP address to the instrument.

NOTE. *Do not use DHCP searching if your instrument has been assigned a static address on a network. If you set this command to ON, the DHCP search will delete or change your static IP address information.*

`OFF` disables the instrument to search the network for a DHCP server.

Examples `ETHERNET:DHCPBOOTP ON` sets the instrument to search for a DHCP server and assign a dynamic IP address to the instrument.

ETHERnet:DNS:IPADDress

This command specifies the network Domain Name Server (DNS) IP address.

Group Ethernet

Syntax
```
ETHERnet:DNS:IPADDress <QString>
ETHERnet:DNS:IPADDress?
```

Arguments `<QString>` is a standard IP address value, enclosed in quotes.

Examples `ETHERNET:DNS:IPADDRESS "128.196.13.252"` sets the DNS IP address that the instrument uses to communicate with the network.

ETHERnet:DOMAINname

This command specifies the network domain name.

Group Ethernet

Syntax
```
ETHERnet:DOMAINname <QString>
ETHERnet:DOMAINname?
```

Arguments `<QString>` is the network domain name, enclosed in quotes.

Examples `ETHERNET:DOMAINNAME "Alpha1.Mycorp.com"` sets the domain name that the instrument uses to communicate with the network.

ETHERnet:ENET:ADDress? (Query Only)

Returns the Ethernet address (MAC address) value assigned to the instrument. This is assigned at the factory and can not be changed.

Group Ethernet

Syntax
```
ETHERnet:ENET:ADDress?
```

Examples `ETHERNET:ENET:ADDRESS?` returns an Ethernet address such as `08:00:11:01:02:03`

ETHERnet:GATEWay:IPADDress

This command specifies the network gateway IP address.

Group Ethernet

Syntax
```
ETHERnet:GATEWay:IPADDress <QString>
ETHERnet:GATEWay:IPADDress?
```

Related Commands ETHERnet:NETWORKCONFig

ETHERnet:IPADDress

ETHERnet:SUBNETMask

Arguments `<QString>` is a standard IP address value, enclosed in quotes.

Examples `ETHERNET:GATEWAY:IPADDRESS "128.143.16.1"` sets the gateway IP address.

ETHERnet:IPADDress

This command sets the IP address assigned to the instrument.

Group Ethernet

Syntax

```
ETHERnet:IPADDress <QString>
ETHERnet:IPADDress?
```

Related Commands

ETHERnet:NETWORKCONFig

ETHERnet:SUBNETMask

ETHERnet:GATEWay:IPADDress

Arguments

<QString> is a standard IP address value, enclosed in quotes.

Examples

ETHERNET:IPADDRESS "123.121.13.214" sets the instrument's IP address.

ETHERnet:LXI:LAN:RESET (No Query Form)

This command resets the LXI local area network.

Group

Ethernet

Syntax

```
ETHERnet:LXI:LAN:RESET
```

Examples

ETHERnet:LXI:LAN:RESET resets the LXI local area network.

ETHERnet:LXI:LAN:SERVICENAMe

This command sets or queries the service name used for the LXI interface.

Group

Ethernet

Syntax

```
ETHERnet:LXI:LAN:SERVICENAMe <QString>
ETHERnet:LXI:LAN:SERVICENAMe?
```

Arguments

<QString> is a quoted string of up to 64 characters that specifies the mDNS service name used for the LXI interface.

Examples

ETHERnet:LXI:LAN:SERVICENAMe? might return "Tektronix Oscilloscope MSO5 053CVV"

ETHERnet:LXI:LAN:STATus? (Query Only)

This query returns the LXI network status: one of OK, FAULT, or IDENTIFY. IDENTIFY indicates that the device identify mode is enabled.

Group Ethernet

Syntax

```
ETHERnet:LXI:LAN:STATus?
```

Related Commands ETHERnet:PING

ETHERnet:PING:STATus?

Returns OK — indicates the network is running and the instrument can "see" the network.

FAULT — indicates the network is not visible, or the network settings are incorrect.

IDENTIFY indicates that a message is being displayed on the instrument's front panel indicating which scope the user is currently accessing.

Examples ETHER:LXI:LAN:STAT? might return FAULT, indicating the network is not visible, or the network settings are incorrect.

ETHERnet:NAME

This command sets or queries the instrument Ethernet hostname assigned to the instrument.

Group Ethernet

Syntax

```
ETHERnet:NAME <QString>
ETHERnet:NAME?
```

Arguments <QString> is the network name assigned to the instrument, enclosed in quotes.

Examples ETHERNET:NAME "labscope1" sets the instrument's network name.

ETHERnet:NETWORKCONFig

This command specifies the Ethernet network configuration setting.

Group Ethernet

Syntax
```
ETHERnet:NETWORKCONFig {AUTOmatic|MANual}
ETHERnet:NETWORKCONFig?
```

Related Commands ETHERnet:NAME

ETHERnet:IPADDress

ETHERnet:SUBNETMask

ETHERnet:GATEWay:IPADDress

ETHERnet:DHCPbootp

Arguments `AUTOmatic` specifies that the instrument's IP address, subnet mask and gateway settings will be received from a DHCP server on the local network.

`MANual` specifies that the Ethernet settings will be configured manually, using ETHERnet:IPADDress, ETHERnet:SUBNETMask, and ETHERnet:GATEWay:IPADDress.

Examples `ETHERnet:NETWORKCONFig MANual` specifies to configure the Ethernet settings manually.

`ETHERnet:NETWORKCONFig?` might return AUTOMATIC, indicating the settings are being configured automatically.

ETHERnet:PING (No Query Form)

Sends a ping packet to the instrument gateway and sets the status accordingly.

Group Ethernet

Syntax
```
ETHERnet:PING EXECute
```

Examples `ETHERNET:PING` EXECute causes the instrument to ping the gateway IP address.

ETHERnet:PING:STATus? (Query Only)

Returns the results of sending the ETHERnet:PING command to ping the gateway IP address.

Group Ethernet

Syntax `ETHERnet:PING:STATus?`

Returns `OK` is returned if the computer at the gateway IP address answers.

`NORESPONSE` is returned if the computer at the gateway IP address does not answer.

`TRYING` is returned if the ping operation is still executing.

`NEVER` is returned if `ETHERnet:PING EXECute` has not been previously sent, indicating that no ping response has ever been received in response to an `ETHERnet:PING EXECute` command since the instrument was powered on.

ETHERnet:SUBNETMask

This command sets or queries the instrument subnet mask value.

Group Ethernet

Syntax
```
ETHERnet:SUBNETMask <QString>
ETHERnet:SUBNETMask?
```

Related Commands ETHERnet:NETWORKCONFig

ETHERnet:IPADDress

ETHERnet:GATEWay:IPADDress

Arguments `<QString>` is the subnet mask value, enclosed in quotes.

Examples `ETHERNET:SUBNETMASK "255.255.255.0"` sets the subnet mask value using standard IP address notation format.

EVENT? (Query Only)

This query-only command returns an event code from the Event Queue that provides information about the results of the last *ESR? read. `EVENT?` also removes the returned value from the Event Queue.

Group Status and Error

Syntax EVENT?

Related Commands ALLEv?
*CLS
DESE
*ESE
*ESR?
EVMsg?
*SRE
*STB?

Examples EVENT? might return :EVENT 110, showing that there was an error in a command header.

EVMsg? (Query Only)

This query-only command removes a single event code from the Event Queue that is associated with the results of the last *ESR? read and returns the event code with an explanatory message. For more information, see Event Handling.

Group Status and Error

Syntax EVMsg?

Related Commands ALLEv?
*CLS
DESE
*ESE
*ESR?
EVENT?
*SRE
*STB?

Returns The event code and message in the following format:

<Event Code><Comma><QString>[<Event Code><Comma><QString>...]<QString>::= <Message>;[<Command>]

where <Command> is the command that caused the error and may be returned when a command error is detected by the instrument. As much of the command will be returned as possible without exceeding the 60 character limit of the <Message> and <Command> string combined. The command string is right-justified.

Examples `EVMSG?` might return `:EVMSG 110,"Command header error"`.

EVQty? (Query Only)

This query-only command returns the number of events that are enabled in the queue. This is useful when using the ALLEv? query, since it lets you know exactly how many events will be returned.

Group Status and Error

Syntax `EVQty?`

Related Commands ALLEv?

EVENT?

EVMsg?

Examples `EVQTY?` might return `:EVQTY 3`, indicating the number of event codes in the Event Queue.

EYEMASK:MASK<x>:COUNt:HITS? (Query Only)

This command returns the total number of hit violations for all segments in the specified mask test.

Conditions Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group Mask

Syntax `EYEMASK:MASK<x>:COUNt:HITS?`

Related Commands PLOT:PLOT<x>:MASK?

Arguments MASK<x> is the number of the specified mask test in an eye diagram plot.

Examples EYEMASK:MASK2:COUNt:HITS? might return 211, indicating that mask 2 has a total of 211 violations for all segments in the mask.

EYEMASK:MASK<x>:COUNt:SEG<y>:HITS? (Query Only)

This command returns the number of hit violations for the specified segment (area).

Conditions Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group Mask

Syntax EYEMASK:MASK<x>:COUNt:SEG<y>:HITS?

Related Commands PLOT:PLOT<x>:MASK?

Arguments MASK<x> is the number of the specified mask test (or mask test plot?).

SEG<y> is the number of the mask segment for which to return hit violations data.

Returns Returns an integer number representing the number of mask test violations for the specified mask est and mask segment.

Examples EYEMASK:MASK1:COUNt:SEG3:HITS? might return 435, indicating that segment 3 of mask test 1 has 435 hit violations.

EYEMASK:MASK<x>:CREATor? (Query Only)

This query-only command returns the name of the eye diagram plot that created the mask.

Conditions Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group Mask

Syntax `EYEMASK:MASK<x>:CREATor?`

Related Commands PLOT:PLOT<x>:MASK?

Arguments `MASK<x>` is the number of the specified mask test (or mask test plot?).

Returns `<QString>` is a quoted string that contains the name of the eye diagram plot associated with the specified mask.

Examples `EYEMASK:MASK4:CREATor?` might return `"plot2"`.

EYEMASK:MASK<x>:ENAbled

This command enables or disables eye mask testing in the specified plot.

Conditions Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group Mask

Syntax
```
EYEMASK:MASK<x>:ENAbled {ON|OFF}
EYEMASK:MASK<x>:ENAbled?
```

Related Commands PLOT:PLOT<x>:MASK?

Arguments `MASK<x>` is the number of the specified mask test (or mask test plot?).

Returns Returns an integer number representing enabled (1) or disabled (0).

Examples `EYEMASK:MASK4:ENAbled ON` enables the eye mask test in the specified mask plot.

`EYEMASK:MASK2:ENAbled?` might return `0`, indicating that mask testing is disabled in the specified mask plot.

EYEMASK:MASK<x>:MASKfile

This command sets or queries the current mask definition file name for the specified mask test.

Conditions Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group Mask

Syntax

```
EYEMASK:MASK<x>:MASKfile <Qstring>
EYEMASK:MASK<x>:MASKfile?
```

Related Commands PLOT:PLOT<x>:MASK?

Arguments <x> is the number of the specified mask test (or mask test plot?).

<Qstring> is a quoted string that defines the file path that specifies the location of the mask file to use, in the format '[<path>]<filename.ext>'. Specifying a path is optional. If no path is entered, the instrument will search in the current working directory as set in FILESystem:CWD.

- If the file argument begins with a file path separator (forward slash character) or a drive designator such as C:, then the file name is interpreted as a full path.
- If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `EYEMASK:MASK1:MASKfile "/C:/Applications/Masks/Serial RapidIO/RIO Serial 2_5G.msk"` sets mask1 to use the RIO serial 2.5G mask file at the specified location

`EYEMASK:MASK4:MASKfile?` might return `"/media/C:/Applications/Masks/1394b/S400b T1.msk"`

EYEMASK:MASK<x>:TESt:SAMple:THReshold

This command sets or queries the total number of hit violations that will cause a mask test failure.

Conditions Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group Mask

Syntax
```
EYEMASK:MASK<x>:TESt:SAMple:THReshold <NR1>
EYEMASK:MASK<x>:TESt:SAMple:THReshold?
```

Related Commands PLOT:PLOT<x>:MASK?

Arguments MASK<x> is the number of the specified mask test (or mask test plot?).

<NR1> is a positive integer indicating the number of mask hits required to cause a fail condition for that mask test.

Examples EYEMASK:MASK1:TESt:SAMple:THReshold 15 sets the specified mask test to fail if 15 or more mask violations occur.

EYEMASK:MASK3:TESt:SAMple:THReshold? might return 20, indicating that the specified mask test is set to fail if 20 or more mask violations occur.

EYEMASK:MASK<x>:TESt:STATUS? (Query Only)

This query-only command returns the mask hit test status.

Conditions Requires option 5-DJA or 6-DJA (Advanced Jitter Analysis).

Group Mask

Syntax
```
EYEMASK:MASK<x>:TESt:STATUS?
```

Related Commands PLOT:PLOT<x>:MASK?

Arguments MASK<x> is the number of the specified mask test (or mask test plot?).

Returns This command returns a string with the mask test status. Valid status strings are:

- PASS. The number of mask hits is less than the target fail threshold.
- FAIL. The number of mask hits is greater than or equal to the target fail threshold.
- OFF. Mask testing disabled on the specified mask plot.

Examples `EYEMASK:MASK2:TESt:STATUS?` might return `Fail`, indicating that mask testing has failed in the specified mask plot.

FACtory (No Query Form)

This command (no query form) resets the instrument to its factory default settings. This command is equivalent to pressing the DEFAULT SETUP button located on the instrument front panel or selecting Default Setup from the File menu.

This command Performs the following in addition to what is done for the *RST command:

- Clears any pending OPC operations.
- Resets the following IEEE488.2 registers:
  - *ESE 0 (Event Status Enable Register)
  - *SRE 0 (Service Request Enable Register)
  - DESE 255 (Device Event Status Enable Register)
  - *PSC 1 (Power-on Status Clear Flag)
- Deletes all defined aliases.
- Enables command headers (:HEADer 1).

Group Save and Recall

Syntax `FACtory`

Related Commands *PSC

RECAll:SETUp

*RST

Arguments None

Examples `FACTORY` resets the instrument to its factory default settings.

FILESystem? (Query Only)

This query-only command returns the directory listing of the current working directory. This query is the same as the `FILESystem:DIR?` query.

Group File System

Syntax `FILESystem?`

Related Commands FILESystem:COPy

FILESystem:CWD

FILESystem:DELEte

FILESystem:DIR?

FILESystem:READFile

FILESystem:REName

FILESystem:WRITEFile

Arguments None.

Examples `FILESYSTEM?` might return `:FILESYSTEM:DIR "myFile.txt","myWaveform.wfm".`

FILESystem:COPy (No Query Form)

This command (no query form) copies a named file to a new file. The new file might be in a totally separate directory than the old file. You can only copy one file at a time using this command. Wild card characters are not allowed.

Group File System

Syntax `FILESystem:COPy {<source_file_path>,<destination_file_path>}`

Related Commands FILESystem:CWD

FILESystem:DELEte

Arguments `<source_file_path>` is a quoted string that defines the file name and path or directory. If the file path is within the current working directory, you need only specify the file name.

`<destination_file_path>` is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

Examples `FILESYSTEM:COPY "E:/setup1.set","E:/SETUPS/setup1.set"` copies the file named setup1.set, on the E drive to a file named setup1.set in the SETUPS directory on the E drive.

:FILESystem:CWD "E:/SETUPS"

:FILESystem:COPy ".","I:/Archive/SETUPS"

:FILESystem:COPy "../ch1.isf","I:/SavedWfms/ch1_new.isf"

FILESystem:CWD

This command sets or queries the current working directory. CWD is short for Current Working Directory. It changes the directory (folder) that the other FILESystem commands operate on.

NOTE. *There are three host ports on the front panel (E:, F:, and G:), and two on the back panel (H: and I:.*

Group File System

Syntax

```
FILESystem:CWD {<new_working_directory_path>}
FILESystem:CWD?
```

Arguments `<new_working_directory_path>` is a quoted string that defines the current working; a directory name can be up to 128 characters.

Examples `FILESYSTEM:CWD "E:/waveforms/"` changes the current working directory to a folder named "waveforms" on the USB flash drive installed in the "E:" USB Host port.

`FILESYSTEM:CWD?` might return `:FILESYSTEM:CWD "E:/"` Indicates the current working directory is the root folder of the E: USB flash drive port.

FILESystem:DELEte (No Query Form)

This command (no query form) deletes a named file or directory from a mass storage device. Once removed, the data in that file or directory can no longer be accessed. If the specified file is a directory, it must be empty before it can be deleted.

Group File System

Syntax `FILESystem:DELEte <file_path>`

Related Commands FILESystem:COPy

FILESystem:CWD

FILESystem:RMDir

Arguments `<file_path>` is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

Examples `FILESYSTEM:DELETE "NOT_MINE.SET"` deletes the file named NOT_MINE.SET from the folder referred to by the FILESYSTEM:CWDFILESystem:CWD command.

FILESystem:DIR? (Query Only)

This query-only command returns a comma separated list of quoted strings. Each string contains the name of a file or directory in the folder referred to by the FILESYSTEM:CWDFILESystem:CWD command.

Group File System

Syntax `FILESystem:DIR?`

Related Commands FILESystem:CWD

FILESystem:MKDir

Arguments None

Examples `FILESYSTEM:DIR?` might return :FILESYSTEM:DIR "161012_132039_000.wfm","161012_132039_001.wfm","161220_191452.png", "161220_191554.csv","170320_132925_0 00.wfm","170320_132929_000.set", "Ch2-Ch3_000.set","E:","F:","G:","H:","I:","J:","Rgjtest_000.set", "RgjSetupRuntTrig4chnl s_000.set","RgjSetup_000.set","ScrnShot1jeh2_000.wfm","ScrnShot1jeh_000.wfm", "ScrnShot_000.png","ScrnShot_000.wfm","Scrn _000.set","Scrn_001.set","Scrn_002.set","Scrn_003.set","Scrn_004.set", "Wfm_000.wfm","Wfm_001.wfm","Wfm_002.wfm","Wfm_003 .wfm","Wfm_004.wfm","Wfm_005.wfm","Wfm_006.wfm","digChans_000.wfm".

FILESystem:HOMEDir? (Query Only)

This query returns the current user's home directory.

Group File System

Syntax `FILESystem:HOMEDir?`

Returns The current user's home directory as a quoted string.

Examples `FILESYSTEM:HOMEDIR?` might return: `"C:"`.

FILESystem:LDIR? (Query Only)

Returns a comma separated list of every file, file size, type, modification date and time, and directory in the folder referred to by the FILESYSTEM:CWDFILESystem:CWD command. This is different than the :DIR query in that it provides a long output format with the file size, type, and modification date/time. Each entry is a semicolon separated list: < file name>;<type>;<size in bytes>;<date>;<time>

Group File System

Syntax `FILESystem:LDIR?`

Returns A comma separated list of every file, file size, type, modification date and time, and directory in the folder referred to by the FILESystem:CWDcommand.

Examples `FILESYSTEM:LDIR?` might return "tek0000CH1.isf;FILE;20342;2009-05-21;13:58:24", "TEMP;DIR;4096;2009-09-15;06:20:44".

FILESystem:MKDir (No Query Form)

This command (no query form) creates a new directory.

Group File System

Syntax `FILESystem:MKDir <directory_path>`

Related Commands FILESystem:CWD

FILESystem:DIR?

Arguments <directory_path> is a quoted string that specifies the directory to create.

Examples FILESYSTEM:MKDIR "E:\NewDirectory" creates the directory named *NewDirectory* at the root of the E drive.

These two commands create the directory MyNewSubDirectory within the existing directory MyDirectory at the root of the E drive:

```
FILESYSTEM:CWD "E:/MyDirectory"; FILESYSTEM:MKDIR
"MyNewSubDirectory"
```

FILESYSTEM:MOUNT:DRIVE

The command form mounts a network drive specified by the quoted string argument. The quoted string argument is a semicolon separated list of the following fields:

- Drive name - The drive name to be mounted. It is a case insensitive single letter followed by a colon. The drive name must be a letter between 'L:' and 'Z:', inclusive.
- Server identity - The server identity is the DNS name of the server or the IP address of the server.
- Path - The path to be mounted (e.g. /level1/level2/mydirectory).
- User name - The user name for the drive.
- User password - The password for the drive.
- Domain name - The domain/workgroup of the target mount.

Domain name, user name, and user password are optional and are only used for mounts requiring SMB/CIFS interworking (MS Windows and MacOS).

The query form returns a 0 or 1 to indicate that the drive name (quoted string) is currently mounted or not. A return of 1 indicates the drive is mounted. A return of 0 indicated the drive is not mounted.

Group File System

Syntax

```
FILESYSTEM:MOUNT:DRIVE <QString>
FILESYSTEM:MOUNT:DRIVE? <QString>
```

Examples

`FILESYSTEM:MOUNT:DRIVE "L:;000.000.0.0;/level1/level2/mydir"` specifies to mount the L: drive using the drive name, server identity, and path.

`FILESYSTEM:MOUNT:DRIVE? "L:"` might return 0, indicating the L: drive is not mounted.

FILESystem:READFile (No Query Form)

This command writes the contents of the specified file to the current interface. If the specified file does not exist or is not readable, an appropriate error event is posted.

Group File System

Syntax `FILESystem:READFile <QString>`

Related Commands FILESystem:CWD

Arguments `<QString>` is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

Examples `FILESYSTEM:READFILE "E:/test_data/tek00016CH1.csv"` reads the content of the specified file, if the file exists and is readable, and sends the content of the file to the current interface.

FILESystem:REName (No Query Form)

This command (no query form) assigns a new name to an existing file or folder.

Group File System

Syntax `FILESystem:REName <old_file_path>,<new_file_path>`

Related Commands FILESystem:CWD

Arguments `<old_file_path>` is a quoted string that defines the file or folder name and path. If the path is within the current working directory, you need only specify the file or folder name.

<new_file_path> is a quoted string that defines the file or folder name and path. If the path is within the current working directory, you need only specify the file or folder name.

Examples

FILESYSTEM:RENAME "E:/TEK00000.SET","E:/MYSETTING.SET" gives the file named TEK00000.SET the new name of MYSETTING.SET. The file remains in the root directory on the E drive.

FILESYSTEM:RENAME "e:/mySettings/tek00000.set","e:/setup1.set". This example illustrates how to move a file from one folder to another.

FILESystem:RMDir (No Query Form)

This command (no query form) deletes a named directory. The directory must be empty.

Group File System

Syntax FILESystem:RMDir <directory_path>

Related Commands FILESystem:CWD

Arguments <directory_path> is a quoted string that defines the folder name and path. If the folder path is within the current working directory, you need only specify the folder name.

Examples FILESYSTEM:RMDIR "E:/OldDirectory" removes the directory named OldDirectory from the root of the E drive.

FILESystem:UNMOUNT:DRIve (No Query Form)

This command unmounts the USB drive specified by the quoted string argument.

Group File System

Syntax FILESystem:UNMOUNT:DRIve <QString>

Related Commands

Arguments <QString> is a quoted string that specifies which USB drive to unmount. String is a case insensitive single letter followed by a colon.

Examples FILESYSTEM:UNMOUNT:DRIVE "G:" specifies to unmount the flash drive installed in the right-most USB slot on the front of the instrument.

FILESystem:WRITEFile (No Query Form)

This command (no query form) writes the specified block data to the specified file on the instruments file system. If the destination file cannot be written, an error event is posted.

Group File System

Syntax FILESystem:WRITEFile <file_path>,<data>

Related Commands FILESystem:CWD

Arguments <file path> is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

<data> is the specified block data to be written.

FPAnel:PRESS (No Query Form)

This command is used to emulate a button press. When used with knob enumerations, this command pushes the knob. Use the FPAnel:TURN command to emulate knob turns.

Group Miscellaneous

Syntax FPAnel:PRESS {AUTOset|BUS|CH1<x>|CLEAR|CURsor|DEFaultsetup|FASTAcq|FORCetrig|GPKNOB1|GPKNOB2|HIGHRES|HORZPOS|HORZScale|MATh|NEXt|PREv|REF|RUNSTop|SETTO50|SINGleseq|TOUCHSCReen|TRIGMode|TRIGSlope|USER|VERTPOS|VERTSCALE|ZOOM}

Arguments Arguments are instrument buttons.

Examples `FPANEL:PRESS FORCETRIG` emulates pressing the Force trigger button.

FPAnel:TURN

This command is used to emulate a knob turn. The optional NR1 specifies the number of clicks where negative values indicate counter clockwise. If not specified, the default of 1 click is used indicating the knob is turned clockwise 1 click.

Group Miscellaneous

Syntax

```
FPAnel:TURN {GPKNOB1|GPKNOB2|HORZPOS|HORZScale|PANKNOB|
TRIGLevel|VERTPOS|VERTSCALE|ZOOM} [,<NR1>]
```

Arguments Arguments are knobs that turn obtained during startup.

`<NR1>` is the number of clicks to turn the knob.

Examples `FPANEL:TURN TRIGLEVEL ,3` emulates turning the trigger Level knob 3 clicks in the clockwise direction.

HEADer

This command sets or queries the Response Header Enable State that causes the instrument to either include or omit headers on query responses.

NOTE. *This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk); these commands never return headers.*

Whether the long or short form of header keywords and enumerations are returned is dependent upon the state of :VERBose.

Group Miscellaneous

Syntax

```
HEADer {<NR1>|OFF|ON}
HEADer?
```

Related Commands VERBose

Arguments

<NR1> = 0 sets the Response Header Enable State to false; any other value sets this state to true.

OFF sets the Response Header Enable State to false. This causes the instrument to omit headers on query responses, so that only the argument is returned.

ON sets the Response Header Enable State to true. This causes the instrument to include headers on applicable query responses. You can then use the query response as a command.

Examples

HEADER OFF specifies that the instrument omits headers on query responses, so that only the argument is returned.

HEADER? might return :HEADER 1, indicating that the instrument is including headers on applicable query responses.

With :VERBose ON and :HEADer ON, the :ACQuire:MODe? query might return: :ACQUIRE:MODE SAMPLE.

With :VERBose OFF and :HEADer ON, the :ACQuire:MODe? query might return: :ACQ:MOD SAM

With :VERBose ON and :HEADer OFF, the :ACQuire:MODe? query might return: SAMPLE

With :VERBose OFF and :HEADer OFF, the :ACQuire:MODe? query might return: SAM

HORizontal? (Query Only)

Queries the current horizontal settings.

Group

Horizontal

Syntax

HORizontal?

Examples

HORIZONTAL? might return :HORIZONTAL:SAMPLERATE 6.2500E+9;SAMPLERATE:ANALYZEMODE:MINIMUM:VALUE AUTOMATIC;OVERRIDE 1;:HORIZONTAL:MODE AUTO;MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:MAXIMUM:ZOOMOVERRIDE 1;VALUE 1250;:HORIZONTAL:MODE:MANUAL:CONFIGURE HORIZONTALSCALE;:HORIZONTAL:MAIN:UNITS S;:HORIZONTAL:DIVISIONS 10;RECORDLENGTH 62500;DELAY:TIME 0.0E+0;MODE 1;:HORIZONTAL:POSITION 50.0000;SCALE 1.0000E-6;ACQDURATION 10.0000E-6.

HORizontal:ACQDURATION? (Query Only)

This query returns the timebase duration.

Group Horizontal

Syntax `HORizontal:ACQDURATION?`

Returns `<NR3>` returns the duration of the acquisition.

Examples `HORIZONTAL:ACQDURATION?` might return `:HORIZONTAL:ACQDURATION 5.0E-9`, indicating the acquisition duration is 5.0 us.

HORizontal:DELay:MODe

This command sets or queries the horizontal delay mode.

Group Horizontal

Syntax
```
HORizontal:DELay:MODe {OFF|ON|<NR1>}
HORizontal:DELay:MODe?
```

Related Commands HORizontal:POSition

Arguments `OFF` sets the Horizontal Delay Mode to off. This causes the HORizontal:POSition command to operate like the HORIZONTAL POSITION knob on the front panel.

`ON` sets the Horizontal Delay Mode to on. This causes the `HORizontal:DELay:TIMe` command to operate like the HORIZONTAL POSITION knob on the front panel.

`<NR1>` = 0 sets the Horizontal Delay Mode to off; any other value sets this mode to on.

Examples `HORIZONTAL:DELAY:MODE OFF` sets the Horizontal Delay Mode to off, allowing the horizontal position command to operate like the HORIZONTAL POSITION knob on the front panel.

`HORIZONTAL:DELAY:MODE?` might return `HORIZONTAL:DELAY:MODE OFF` indicating that the Horizontal Delay Mode is off and that the horizontal position command operates like the HORIZONTAL POSITION knob on the front panel.

HORizontal:DELay:TIMe

This command sets or queries the horizontal delay time that is used when delay mode is on.

Group Horizontal

Syntax
```
HORizontal:DELay:TIMe <NR3>
HORizontal:DELay:TIMe?
```

Related Commands HORizontal:POSition

Arguments `NR3` is the delay in seconds.

Examples `HORizontal:DELay:TIME 0.3` sets the delay of acquisition data so that the resulting waveform is centered 300 ms after the trigger occurs.

HORizontal:DIVisions? (Query Only)

This query-only command returns the number of graticule divisions.

Group Horizontal

Syntax
```
HORizontal:DIVisions?
```

Examples `HORIZONTAL:DIVISIONS?` might return `:HORIZONTAL:DIVISIONS 10.0000`, indicating that the waveform is displayed across ten divisions.

HORizontal:FASTframe? (Query Only)

This query returns all information under horizontal:fastframe.

Group Horizontal

Syntax
```
HORizontal:FASTframe?
```

Examples `HORIZONTAL:FASTFRAME?` might return `:HORIZONTAL:FASTFRAME:SELECTED 1;SUMFRAME:STATE 0;:HORIZONTAL:FASTFRAME:COUNT 2;MULTIPLEFRAMES:MODE OFF;:HORIZONTAL:FASTFRAME:STATE 0;REF:INCLUDE 0;FRAME 1`, the current FastFrame settings.

HORizontal:FASTframe:COUNt

This command sets or returns the number of frames.

Group Horizontal

Syntax
```
HORizontal:FASTframe:COUNt <NR1>
HORizontal:FASTframe:COUNt?
```

Arguments `<NR1>` is the number of frames.

Examples `:HORizontal:FASTframe:COUNt 10` sets the number of frames to be acquired to 10. If this is beyond the maximum number of frames, the value will be constrained. For example, if 8 is the current maximum (based on record length) 8 will be set instead of 10.

HORizontal:FASTframe:MAXFRames? (Query Only)

This query returns the maximum number of frames.

Group Horizontal

Syntax `HORizontal:FASTframe:MAXFRames?`

Examples `HORIZONTAL:FASTFRAME:MAXFRAMES?` might return `:HORIZONTAL:FASTFRAME:MAXFRAMES 2`, indicating the maximum number of frames is 2.

HORizontal:FASTframe:MULtipleframes:MODe

This command sets or returns the overlay display type.

Group Horizontal

Syntax
```
HORizontal:FASTframe:MULtipleframes:MODe {OFF|OVERlay}
HORizontal:FASTframe:MULtipleframes:MODe?
```

Arguments `OFF` specifies only displaying the selected frame.

`OVERlay` specifies overlaying all frames with the temperature palette. The summary frame is not included in the overlay. The selected frame is drawn in blue on top of all other frames.

Examples `:HORizontal:FASTframe:MULtipleframes:MODE OVERlay` will cause the entire acquired set of frames to be rendered in a single image, similar to how FastAcq appears. The currently selected frame is drawn on top in blue.

HORizontal:FASTframe:REF:FRAme

This command sets or returns the reference frame number.

Group Horizontal

Syntax
```
HORizontal:FASTframe:REF:FRAme <NR1>
HORizontal:FASTframe:REF:FRAme?
```

Arguments `<NR1>` is the reference frame number.

Examples `HORIZONTAL:FASTFRAME:REF:FRAME?` might return `:HORIZONTAL:FASTFRAME:REF:FRAME 1`, indicating the reference frame number is 1.

HORizontal:FASTframe:REF:INCLUde

This command sets or returns whether the reference frame delta information is shown in the display.

Group Horizontal

Syntax
```
HORizontal:FASTframe:REF:INCLUde {<NR1>|OFF|ON}
HORizontal:FASTframe:REF:INCLUde?
```

Arguments ON displays the delta information.

OFF does not display the delta information.

<NR1> a 0 indicates the delta information is off; any other value displays the delta information.

HORizontal:FASTframe:SELECTED

This command sets or returns the selected frame number for acquired frames. Refs have their own selected frames.

Group Horizontal

Syntax
```
HORizontal:FASTframe:SELECTED <NR1>
HORizontal:FASTframe:SELECTED?
```

Arguments <NR1> is the selected frame number for acquired frames.

Examples HORIZONTAL:FASTFRAME:SELECTED? might return :HORIZONTAL:FASTFRAME:SELECTED 1, indicating the selected frame number is 1.

HORizontal:FASTframe:STATE

This command sets or returns the state of FastFrame. Acquisition modes Envelope and Average are not compatible with FastFrame. If FastFrame is on, an attempted set to those acquisition modes will fail and revert to Sample mode. If FastFrame is turned on while in one of those acquisition modes, the acquisition mode is changed to Sample.

Group Horizontal

Syntax
```
HORizontal:FASTframe:STATE {<NR1>|OFF|ON}
HORizontal:FASTframe:STATE?
```

Arguments ON indicates FastFrame is active.

OFF indicates that FastFrame is off.

<NR1> A 0 turns off FastFrame; any other value activates FastFrame.

Examples `HORIZONTAL:FASTFRAME:STATE?` might return `:HORIZONTAL:FASTFRAME:STATE 0`, indicating FastFrame is off.

HORizontal:FASTframe:SUMFrame? (Query Only)

This command sets or returns the summary frame type. Turning on Summary Frame does not adjust the numberFrames value as long as there is room for an additional frame. If there is not enough room then numberFrames will be reduced by 1. The numberFrames value is always the number of frames to acquire.

Group Horizontal

Syntax
```
HORizontal:FASTframe:SUMFrame? {NONe|AVErage|ENVelope}
HORizontal:FASTframe:SUMFrame??
```

Arguments `NONe` sets the Summary frame to off.

`AVErage` sets the Summary frame to average of all acquired frames.

`ENVelope` sets the Summary frame to envelope of all acquired frames.

Examples `HORIZONTAL:FASTFRAME:SUMFRAME?` might return `:HORIZONTAL:FASTFRAME:SUMFRAME NONE`, indicating the summary frame is off.

HORizontal:FASTframe:SUMFrame:STATE

This command sets or returns the state of FastFrame summary frame. Summary frame mode is set automatically based on the acquisition mode. When in Sample mode, the summary frame type is set to Average. When in Peak Detect mode, the summary frame type is set to Envelope. When in High Res mode, the summary frame type is set to Average.

Group Horizontal

Syntax
```
HORizontal:FASTframe:SUMFrame:STATE {<NR1>|OFF|ON}
HORizontal:FASTframe:SUMFrame:STATE?
```

Arguments `ON` indicates summary frame is active.

`OFF` indicates that summary frame is off.

<NR1> a 0 turns off summary frame; any other value activates the summary frame.

Examples HORIZONTAL:FASTFRAME:SUMFRAME:STATE? might return :HORIZONTAL:FASTFRAME:SUMFRAME:STATE 0, indicating the summary frame is off.

HORizontal:FASTframe:TIMEStamp:ALL? (Query Only)

This query-only command returns the time stamp of all frames. The format is (Frame #: TimeStamp, Frame #: TimeStamp, and so on). Each time-stamp string is of the form DD.MM.YYYY.HH:MM::SS.xxxxxxxxxxx.

Group Horizontal

Syntax HORizontal:FASTframe:TIMEStamp:ALL?

Returns Returns all of the timestamps.

Examples HORIZONTAL:FASTFRAME:TIMESTAMP:ALL? might return :HORIZONTAL:FASTFRAME:TIMESTAMP:ALL "1: 31.12.1969.16:00:00.000000000000", indicating the only time

HORizontal:FASTframe:TIMEStamp:DELTa? (Query Only)

This query returns the time difference between the Selected and Reference time-stamps.

Group Horizontal

Syntax HORizontal:FASTframe:TIMEStamp:DELTa?

Returns Return value is in seconds and fraction of a second.

Examples HORIZONTAL:FASTFRAME:TIMESTAMP:DELTA? might return :HORIZONTAL:FASTFRAME:TIMESTAMP:DELTA "0.000000000000" indicating the time difference is 0.0 seconds.

HORizontal:FASTframe:TIMEStamp:REFerence? (Query Only)

This query returns the time-stamp of the FastFrame Reference frame.

Group Horizontal

Syntax `HORizontal:FASTframe:TIMEStamp:REFerence?`

Examples `HORIZONTAL:FASTFRAME:TIMESTAMP:REFERENCE?` might return `:HORIZONTAL:FASTFRAME:TIMESTAMP:REFERENCE "31.12.1969.17:00:00.000000000000"` indicating the reference timestamp is 31.12.1969.17:00:00.000000000000.

HORizontal:FASTframe:TIMEStamp:SELECTED? (Query Only)

This query returns the time-stamp of the FastFrame Selected acquired frame.

Group Horizontal

Syntax `HORizontal:FASTframe:TIMEStamp:SELECTED?`

Examples `HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED?` might return `:HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED "31.12.1969.17:00:00.000000000000"` indicating the timestamp of the selected frame is 31.12.1969.17:00:00.000000000000.

HORizontal:FASTframe:XZEro:ALL? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command for all frames. This value is in units of WFMOutpre:XUNit. The format is a string of the form (frame #:zxero, frame #:xzero, and so on).

Group Horizontal

Syntax `HORizontal:FASTframe:XZEro:ALL?`

Returns This returns the XZERO values for all of the frames.

Examples `HORIZONTAL:FASTFRAME:XZERO:ALL?` might return `:HORIZONTAL:FASTFRAME:XZERO:ALL "1:  1.6000E-10"`, indicating the time between the trigger sample and the actual trigger.

HORizontal:FASTframe:XZEro:REF? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command for the reference frame. This value is in units of WFMOutpre:XUNit.

Group Horizontal

Syntax `HORizontal:FASTframe:XZEro:REF?`

Examples `HORIZONTAL:FASTFRAME:XZERO:REF?` might return `:HORIZONTAL:FASTFRAME:XZERO:REF 1.0` indicating the sub-sample time is 1.0.

HORizontal:FASTframe:XZEro:SELECTED? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command for the selected frame. This value is in units of WFMOutpre:XUNit.

Group Horizontal

Syntax `HORizontal:FASTframe:XZEro:SELECTED?`

Examples `HORIZONTAL:FASTFRAME:XZERO:SELECTED?` might return `:HORIZONTAL:FASTFRAME:XZERO:SELECTED 1.0` indicating the sub-sample time is 1.0.

HORizontal:MAIn:INTERPRatio? (Query Only)

This query-only command returns the Horizontal interpolation ratio.

Group Horizontal

Syntax `HORizontal:MAIn:INTERPRatio?`

Examples `HORIZONTAL:MAIN:INTERPRATIO?` might return `:HORIZONTAL:MAIN:INTERPRATIO 10.0`.

HORizontal:MODE

This command set or queries the horizontal operating mode.

Group Horizontal

Syntax
```
HORizontal:MODE {AUTO|MANual}
HORizontal:MODE?
```

Related Commands HORizontal:MODE:RECOrdlength

HORizontal:MODE:SAMPLERate

HORizontal:MODE:SCAle

Arguments `AUTO` selects the automatic horizontal model. Auto mode automatically adjusts the sample rate and record length to provide a high acquisition rate in Fast Acq or signal fidelity in analysis. Record length is read only.

`MANUAL` selects the manual horizontal model. Manual mode lets you change the sample rate, horizontal scale, and record length. These values interact. For example, when you change record length then the horizontal scale also changes.

Examples `HORIZONTAL:MODE AUTO` sets the horizontal mode to auto.

`HORIZONTAL:MODE?` might return `:HORIZONTAL:MODE MANUAL`, indicating that the horizontal mode is manual.

HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:VALue

Sets or queries the horizontal FastAcq maximum record length.

Group Horizontal

Syntax

```
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:VALue
<NR1>
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:
VALue?
```

Arguments

<NR1> is the horizontal FastAcq maximum record length.

Examples

HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:MAXIMUM:VALUE 6250 sets the maximum value to 6.25 k.

HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:ZOOMOVERride

Sets or queries the flag which allows override of the horizontal FastAcq maximum record length.

Group

Horizontal

Syntax

```
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:
ZOOMOVERride {OFF|ON|0| 1}
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:
ZOOMOVERride?
```

Arguments

OFF does not allow override of the horizontal FastAcq maximum record length.

ON allows override of the horizontal FastAcq maximum record length.

0 does not allow override of the horizontal FastAcq maximum record length.

1 allows override of the horizontal FastAcq maximum record length.

Examples

HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:MAXIMUM:ZOOMOVERRIDE 0 does not allow override of the horizontal FastAcq maximum record length.

HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:MAXIMUM:ZOOMOVERRIDE? might return :HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:MAXIMUM:ZOOMOVERRIDE 1 indicating that override of the horizontal FastAcq maximum record length is allowed.

HORizontal:MODe:MANual:CONFIGure

Sets or queries which horizontal control (scale or record length) will primarily change when the sample rate is changed in Manual mode. If the selected control (scale or record length) reaches a limit then the unselected control (record length or scale) may also change.

Group Horizontal

Syntax

```
HORizontal:MODe:MANual:CONFIGure
{HORIZontalscale|RECORDLength}
HORizontal:MODe:MANual:CONFIGure?
```

Arguments

`HORIZontalscale` will change when sample rate is adjusted.

`RECORDLength` will change when sample rate is adjusted.

Examples `HORIZONTAL:MODE:MANUAL:CONFIGURE RECORDLength` allows the record length be adjusted when sample rate is changed in Manual mode.

HORizontal:MODE:RECOrdlength

This command sets or queries the record length.

Group Horizontal

Syntax

```
HORizontal:MODE:RECOrdlength <NR1>
HORizontal:MODE:RECOrdlength?
```

Arguments `<NR1>` is the record length in samples. Manual mode lets you change the record length, while the record length is read only for Automatic mode.

Examples `HORIZONTAL:MODE:RECORDLENGTH 1000` sets the record length to 1000 samples.

`HORIZONTAL:MODE:RECORDLENGTH?` might return `:HORIZONTAL:MODE:RECORDLENGTH 1000000`, indicating that the record length is set to 1,000,000 samples

HORizontal:MODE:SAMPLERate

This command sets or queries the sample rate.

Group Horizontal

Syntax
```
HORizontal:MODE:SAMPLERate <NR1>
HORizontal:MODE:SAMPLERate?
```

Arguments <NR1> is the sample rate in samples per second.

Examples HORIZONTAL:MODE:SAMPLERATE 1e6 sets the sample rate to 1 million samples per second.

HORIZONTAL:MODE:SAMPLERATE? might return :HORIZONTAL:MODE:SAMPLERATE 5.0000E+6, indicating that the sample rate is set to 5 million samples per second.

HORizontal:MODE:SCAle

This command sets or queries the horizontal scale.

Group Horizontal

Syntax
```
HORizontal:MODE:SCAle <NR1>
HORizontal:MODE:SCAle?
```

Arguments <NR1> is the horizontal scale in seconds per division.

Examples HORIZONTAL:MODE:SCALE 2e-9 sets the horizontal scale to 2 ns per division.

HORIZONTAL:MODE:SCALE? might return :HORIZONTAL:MODE:SCALE 20.0000E-6, indicating that the horizontal scale is set to 10 µs per division.

HORizontal:POSition

This command sets or queries the horizontal position as a percent of screen width. When Horizontal Delay Mode is turned off, this command is equivalent to adjusting the HORIZONTAL POSITION knob on the front panel. When Horizontal Delay Mode is turned on, the horizontal position is forced to 50%.

Group Horizontal

Syntax
```
HORizontal:POSition <NR3>
HORizontal:POSition?
```

Arguments <NR3> is from 0 to ≈100 and is the position of the trigger point on the screen (0 = left edge, 100 = right edge).

Examples HORIZONTAL:POSITION 10 sets the trigger position of the waveform such that 10% of the display is to the left of the trigger position.

HORIZONTAL:POSITION? might return :HORIZONTAL:POSITION 5.0000E+01 indicates the trigger point displayed on the screen.

HORizontal:PREViewstate? (Query Only)

This query returns the display system preview state.

Group Horizontal

Syntax
```
HORizontal:PREViewstate?
```

Returns <NR1> = 1 if the system is in the preview state.

<NR1> = 0 if the system is not in the preview state.

Examples HORIZONTAL:PREVIEWSTATE? might return :HORIZONTAL:PREVIEWSTATE 0 indicating the system is not in the preview state.

HORizontal:RECOrdlength

This command sets or queries the horizontal record length. To change the record length the Horizontal Mode must be set to Manual.

Group Horizontal

Syntax
```
HORizontal:RECOrdlength <NR1>
HORizontal:RECOrdlength?
```

Arguments <NR1> is the horizontal record length.

Examples HORIZONTAL:RECORDLENGTH 1000 sets the record length to 1000 samples.

HORIZONTAL:RECORDLENGTH? might return :HORIZONTAL:RECORDLENGTH 1000000, indicating that the record length is set to 1,000,000 samples

HORizontal:ROLL? (Query Only)

Queries the horizontal roll mode status.

Group Horizontal

Syntax HORizontal:ROLL?

Returns ON indicates roll mode is active.

OFF indicates that summary frame is off.

<NR1> a 0 indicates roll mode is off; any other value activates roll mode.

Examples HORIZONTAL:ROLL? might return :HORIZONTAL:ROLL 0 indicating roll mode is off.

HORizontal:SAMPLERate

This command sets or queries the horizontal sample rate.

Group Horizontal

Syntax
```
HORizontal:SAMPLERate <NR3>
HORizontal:SAMPLERate?
```

Arguments <NR3> is the horizontal sample rate in samples per second.

Examples HORIZONTAL:SAMPLERATE 1e6 sets the sample rate to 1 million samples per second.

HORIZONTAL:SAMPLERATE? might return :HORIZONTAL:SAMPLERATE 5.0000E+6, indicating that the sample rate is set to 5 million samples per second.

HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide

Sets or queries the flag which allows override of the horizontal analyze minimum sample rate.

Group Horizontal

Syntax

```
HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide
{OFF|ON|0|1}
HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide?
```

Arguments

0 does not allow override of the horizontal analyze minimum sample rate.

1 allows override of the horizontal analyze minimum sample rate.

OFF does not allow override of the horizontal analyze minimum sample rate.

ON allows override of the horizontal analyze minimum sample rate.

Examples

HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE OFF does not allow override of the horizontal analyze minimum sample rate.

HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE? might return :HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE 1 indicating that override of the horizontal analyze minimum sample rate is allowed.

HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue

Sets or queries the minimum sample rate used by Analysis Automatic horizontal mode.

Group Horizontal

Syntax

```
HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue
{AUTOmatic|<NR3>}
HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue?
```

Related Commands

Arguments

AUTOmatic allows the instrument to set the minimum value.

<NR3> is the minimum sample rate.

Examples

`HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue AUTOMATIC` allows the instrument to set the minimum value.

`HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue?` might return `HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:VALUE AUTOMATIC` indicating the instrument automatically sets the minimum value.

HORizontal:SCAle

This command sets or queries the horizontal scale.

Group

Horizontal

Syntax

```
HORizontal:SCAle <NR3>
HORizontal:SCAle?
```

Arguments

`<NR3>` is the horizontal scale in time per division.

Returns

The current horizontal scale is returned.

Examples

`HORIZONTAL:SCALE 20e-9` sets the horizontal scale to 20 ns/division.

`HORIZONTAL:SCALE?` might return `HORIZONTAL:SCALE 20e-9` indicating the horizontal scale is set to 20 ns/division.

ID? (Query Only)

This query-only command returns identifying information about the instrument and related firmware similar to that returned by the *IDN? IEEE488.2 common query but does not include the instrument serial number.

Group

Miscellaneous

Syntax

`ID?`

Related Commands

*IDN?

Examples `ID?` might return `ID TEK/MSO54,CF:91.1CT,FV:1.2.0.2886`, indicating that the instrument model number is set to MSO54, codes and formats is CF:91.1CT, and firmware version is FV:1.2.0.

*IDN? (Query Only)

This query-only command returns the instrument identification code.

Group Miscellaneous

Syntax `*IDN?`

Related Commands ID?

Examples `*IDN?` might return `TEKTRONIX,MSO54,C100123,CF:91.1CT FV:1.2.0.2886`, indicating the instrument model number, serial number, codes and formats number, and firmware version number.

LIC:UNINSTALL? (Query Only)

Returns the exit license indicated for the user to return to their TekAMS account. Active licenses can be specified by their nomenclature. TransactionIDs can be used to specify an active license or a previously uninstalled license. In either case, the exit-license is returned as block-data.

Group Miscellaneous

Syntax `LIC:UNINSTALL? <QString>`

Arguments `<QString>` is the nomenclature of an active license or a TransactionIDs to specify an active license or a previously uninstalled license.

Returns The exit-license is returned as block-data.

Examples `LIC:UNINSTALL? “LIC5-SRAERO”` uninstalls the given license and returns the license block data.

`LIC:UNINSTALL? “569765772”` uninstalls the license with the given transaction ID and returns the license block data.

LICense? (Query Only)

This query-only command returns all license parameters.

Group Miscellaneous

Syntax `LICense?`

Examples `LICENSE?` might return `:LICENSE:GMT "2016-05-23T17:05:10-07:00";HID "TMS-AAA9CS4US5SGJN6X";LIST "5-BW-1000";COUNT 1`.

LICENSE:APPID? (Query Only)

This query returns a comma-separated list of the active application IDs. If a string argument is provided, a "0" or "1" is returned, according to whether the string matches an active application ID.

Group Miscellaneous

Syntax `LICENSE:APPID? {<QString>}`

Returns This query returns a comma-separated list of the active application IDs. If a string argument is provided, a "0" or "1" is returned, according to whether the string matches an active application ID.

Examples `LIC:APPID?` might return "`BW5-2000, AFG, DVM, DJA`", which is a complete list of the active applications.

`LIC:APPID? "AFG"` would return "1" because the app is active.

LICense:COUNt? (Query Only)

This query returns a count of the number of active licenses installed.

Group Miscellaneous

Syntax `LICense:COUNt?`

Returns A count of the number of active licenses installed.

Examples `LICENSE:COUNT?` might return `:LICENSE:COUNT 2` indicating that 2 active licenses are installed

LICense:ERRor? (Query Only)

This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. This command is an alias for ALLEV?.

Group Miscellaneous

Syntax `LICense:ERRor?`

Related Commands ALLEv?

*ESR?

EVMsg?

Examples LICense:ERRor? might return `0, "No events to report- queue empty"`.

LICense:GMT? (Query Only)

This query returns the GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.

Group Miscellaneous

Syntax `LICense:GMT?`

Returns The GMT time in ISO 8601 format, the local date, 24 hour time and time-zone offset.

Examples `LICENSE:GMT?` might return `:LICENSE:GMT "2016-05-23T17:05:10-07:00"` indicating the local GMT time.

LICense:HID? (Query Only)

This query returns the instrument HostID unique identifier.

Group Miscellaneous

Syntax `LICense:HID?`

Returns The instrument HostID unique identifier.

Examples `LICENSE:HID?` might return `:LICENSE:HID "TMS-9CS4US5SGJN6X"`

LICense:INSTall (No Query Form)

This command accepts a <block_data> license and installs it on the instrument. Restarting the instrument may be necessary to fully activate the additional capabilities.

Group Miscellaneous

Syntax `LICense:INSTall <block_data>`

Arguments `<block_data>` is the license in block data format.

Examples `LICENSE:INSTALL` <block_data>

LICense:ITEM? (Query Only)

This query returns the details pertaining to a specific license. The NR1 argument is zero-indexed. If no argument is provided, zero is assumed.

Group Miscellaneous

Syntax `LICense:ITEM? <NR1>`

Arguments <NR1> is the zero-indexed argument specifying a specific license.

Returns The details pertaining to a specific license.

Examples `LICENSE:ITEM? 0` might return `"5-BW-1000,Fixed,2116-06-15T14:55:54-07:00,11870047,BW5-1000,1 GHz bandwidth on 5 series oscilloscopes"`

LICense:LIST? (Query Only)

This query returns the active license nomenclatures as a comma-separated list of strings. Duplicate nomenclatures, that is, the same license but with different expiration dates, are included.

Group Miscellaneous

Syntax `LICense:LIST?`

Returns The active license nomenclatures as a comma-separated list of strings.

Examples `LICENSE:LIST?` might return `:LICENSE:LIST "5-BW-1000","SUP5-RL125M"`.

LICense:VALidate? (Query Only)

This query accepts a license nomenclature as an argument and returns True (1) if that nomenclature is active and any required hardware is installed, or False (0) if either the nomenclature is not active or required hardware is not installed.

Group Miscellaneous

Syntax `LICense:VALidate? <QString>`

Arguments `<QString>` is the license nomenclature.

Returns `True (1)` if that nomenclature is active and any required hardware is installed.

`False (0)` if either the nomenclature is not active or required hardware is not installed.

Examples LICENSE:VALIDATE? "AFG" might return :LICENSE:VALIDATE "AFG",0 indicating the license is not active.

LOCk

This command enables or disables the touch screen and all front panel buttons and knobs. There is no front panel equivalent.

When the front panel is locked, the front panel commands will not work and will not generate error events. You can work around a locked front panel, by using the appropriate programmatic interface commands, instead of the front-panel commands. For example, to set the trigger level to 50%, you could use TRIGger:A SETLevel. To force a trigger, you could use TRIGger FORCe.

Group Miscellaneous

Syntax
```
LOCk {ALL|NONe}
LOCk?
```

Related Commands UNLock

TOUCHSCReen:STATe

Arguments ALL disables all front panel controls and the touch screen.

NONe enables all front panel controls and the touch screen. The UNLock ALL command only unlocks the front panel controls.

If the instrument is in the Remote With Lockout State (RWLS), the LOCk NONe command has no effect. For more information, see the ANSI/IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

Examples LOCK ALL locks the front panel controls and touch screen.

LOCK? might return :LOCK NONE, indicating that the front panel controls are enabled.

*LRN? (Query Only)

This query-only command returns the commands that list the instrument settings, allowing you to record or “learn” the current instrument settings. You can use these commands to return the instrument to the state it was in when you made the *LRN? query. This command is identical to the SET? command.

Group Miscellaneous

Syntax *LRN?

Related Commands SET?

Examples *LRN? might return the following response: *RST;:PARAMBATCHING 0;:HEADER 1;:VERBOSE 1;:ALIAS:STATE 0;:ACQUIRE:STATE 1;:ACQUIRE:NUMENV INFINITE;:FASTACQ:STATE 0;:FASTACQ:PALETTE TEMPERATURE;:HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:VALUE AUTOMATIC;:SELECT:CH1 1;:SELECT:CH2 0;:SELECT:CH3 0;:SELECT:CH4 0;:SELECT:CH5 0;:SELECT:CH6 0;:SELECT:CH7 0;:SELECT:CH8 0;:CH1:BANDWIDTH 250.0000E+6;:CH2:BANDWIDTH 500.0000E+6;:CH3:BANDWIDTH 500.0000E+6;:CH4:BANDWIDTH 500.0000E+6;:CH5:BANDWIDTH 500.0000E+6;:CH6:BANDWIDTH 500.0000E+6;:CH7:BANDWIDTH 120.0000E+6;:CH8:BANDWIDTH 500.0000E+6;:CH1:BANDWIDTH:ENHANCED 0;:CH2:BANDWIDTH:ENHANCED 0;:CH3:BANDWIDTH:ENHANCED 0;:CH4:BANDWIDTH:ENHANCED 0;:CH5:BANDWIDTH:ENHANCED 0;:CH6:BANDWIDTH:ENHANCED 0;:CH7:BANDWIDTH:ENHANCED 0;:CH8:BANDWIDTH:ENHANCED 0;:DATA:DESTINATION REF1;:DATA:ENCDG ASCII;:DATA:SOURCE CH1;:DATA:START 1;:DATA:STOP 1000000;:CH1:PROBETYPE ANALOG;:CH2:PROBETYPE ANALOG;:CH3:PROBETYPE ANALOG;:CH4:PROBETYPE ANALOG;:CH5:PROBETYPE DIGITAL;:CH6:PROBETYPE ANALOG;:CH7:PROBETYPE ANALOG;:CH8:PROBETYPE ANALOG;:HORIZONTAL:MODE AUTO;:HORIZONTAL:SAMPLERATE:ANALYZEMODE:MINIMUM:OVERRIDE 1;:HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:MAXIMUM:ZOOMOVERRIDE 1;:HORIZONTAL:MODE:AUTOMATIC:FASTACQ:RECORDLENGTH:MAXIMUM:VALUE 1250;:HORIZONTAL:MODE:MANUAL:CONFIGURE HORIZONTALSCALE;:HORIZONTAL:SAMPLERATE 6.2500E+9;:DISPLAY:WAVEVIEW1:ZOOM:ZOOM1:VERTICAL:SCALE 1.0000;:TRIGGER:B:TYPE EDGE;:TRIGGER:A:TYPE EDGE;:ACQUIRE:SEQUENCE:NUMSEQUENCE 1;:ACQUIRE:SEQUENCE:MODE NUMACQS;:ACQUIRE:MODE SAMPLE;:ACQUIRE:STOPAFTER RUNSTOP;:ACQUIRE:FASTACQ:PALETTE TEMPERATURE;:ACQUIRE:FASTACQ:STATE 0;:ACQUIRE:NUMAVG 16T.

MAINWindow:BADGe:BRINgtoview (No Query Form)

This command is used to automatically scroll the specified badge to make it visible. The input to the command is the badge title name.

Group Miscellaneous

Syntax `MAINWindow:BADGe:BRINgtoview <Qstring>`

Arguments `<Qstring>` specifies a badge title name to make the badge visible.

Examples `MAINWindow:BADGe:BRINgtoview "Meas 1"` makes sure the Meas1 badge is shown in the RRB.

MAINWindow:FONTSize (No Query Form)

Sets the font size for UI text elements. Font size range is 12 to 20 points.

Group Miscellaneous

Syntax `MAINWindow:FONTSize <NR1>`

Arguments `<NR1>` specifies the font size.

Examples `MAINWindow:FONTSize 18` sets the UI text size to 18 points.

MAINWindow:RRBDisplaystate (No Query Form)

Sets the display state of the Results readout bar to ON (displayed) or OFF (not displayed).

Group Miscellaneous

Syntax `MAINWindow:RRBDisplaystate {1|ON|0|OFF}`

Arguments `1` or `ON` enables Autoset to change vertical settings.

`0` or `OFF` disables Autoset from changing vertical settings.

Examples `MAINWindow:RRBDisplaystate 1` expands (displays) the Results readout bar.

MASK:DELete (No Query Form)

This command deletes all mask segments of the specified mask test.

Conditions Requires option MTM.

Group Mask

Syntax `MASK:DELete “MASK<x>”`

Arguments `MASK<x>` specifies the mask test. This argument is enclosed in quotes.

Examples `MASK:DELETE “mask2”` will delete all mask segments of mask test 2.

MASK:MASK<x>:COUNT? (Query Only)

This command returns the total number of mask hits in all segments and the number of mask hits in each individual mask segment for the specified mask test.

Conditions Requires option MTM.

Group Mask

Syntax `MASK:MASK<x>:COUNT?`

Arguments `MASK<x>` specifies the mask test.

Returns `<NR1>` is the total number of mask hits in all segments and the number of mask hits in each individual mask segment.

Examples `MASK:MASK1:COUNT?` might return `:MASK:MASK1:COUNT:HITS 211;SEG1:HITS;:MASK:MASK1:COUNT:SEG2:HITS 199; MASK:MASK1:COUNT:SEG3:HITS 12`, indicating the number of mask hits on segment 1, 2, and 3 for mask test 1.

MASK:MASK<x>:COUNT:HITS? (Query Only)

This command returns the total number of mask hits in all mask segments for the specified mask test.

Conditions Requires option MTM.

Group Mask

Syntax

```
MASK:MASK<x>:COUNT:HITS?
```

Arguments MASK<x> specifies the mask test.

Returns <NR1> is the total number of mask hits in all mask segments.

Examples MASK:MASK1:COUNT:HITS? might return :MASK:MASK1:COUNT:HITS 211, indicating 211 mask hits for mask test 1.

MASK:MASK<x>:DEFinedby

This command sets or queries whether the specified mask is defined by segments or tolerances. Segment masks are defined by one or more polygons. Tolerance masks are defined by specified horizontal and vertical tolerances around the mask source.

Conditions Requires option MTM.

Group Mask

Syntax

```
MASK:MASK<x>:DEFinedby {SEGments|TOLerances}
MASK:MASK<x>:DEFinedby?
```

Related Commands MASK:MASK<x>:TOLerance:HORizontal

MASK:MASK<x>:TOLerance:VERTical

MASK:MASK<x>:TOLerance:UPDatenow

Arguments MASK<x> specifies the mask number.

SEGments defines the mask by segments.

TOLerances defines the mask by horizontal and vertical tolerances around mask source.

Examples MASK:MASK1:Definedby SEGments sets mask 1 to be defined by segments.

MASK:MASK1:Definedby? might return :MASK:MASK1:Definedby TOLerances, indicating mask 1 is defined by the horizontal and vertical tolerances.

MASK:MASK<x>:DISplay

This command sets or queries the display state of the mask used for the specified mask test.

Conditions Requires option MTM.

Group Mask

Syntax
```
MASK:MASK<x>:DISplay {ON|OFF}
MASK:MASK<x>:DISplay?
```

Arguments MASK<x> specifies the mask test.

ON sets the display state of the specified mask to on.

OFF sets the display state of the specified mask to off.

Examples MASK:MASK1:DISPLAY OFF disables the display of the mask used by mask test 1.

MASK:MASK1:DISPLAY? might return :MASK:MASK1:DISPLAY OFF, indicating the display state for mask test 1 is off.

MASK:MASK<x>:LIST? (Query Only)

This command queries the list of segments in the mask used by the specified mask test.

Conditions Requires option MTM.

Group Mask

Syntax `MASK:MASK<x>:LIST?`

Arguments `MASK<x>` specifies the mask test.

Returns A comma separated list of segment names.

Examples `MASK:MASK1:LIST?` might return `:MASK:MASK1:LIST SEG1,SEG2,SEG3`, indicating that segment 1, 2, and 3 are used by mask test 1.

MASK:MASK<x>:SEG<x>COUNT:HITS? (Query Only)

The command returns the total number of mask hits in the specified mask segment of the specified mask test.

Conditions Requires option MTM.

Group Mask

Syntax `MASK:MASK<x>:SEG<x>COUNT:HITS?`

Arguments `MASK<x>` specifies the mask test.

`SEG<x>` specifies the mask segment.

Returns `<NR1>` is the total number of mask hits in the specified segment.

Examples `MASK:MASK1:COUNT:SEG2:HITS?` might return `:MASK:MASK1:COUNT:SEG2:HITS 211`, indicating 211 mask 1 hits in segment 2.

MASK:MASK<x>:SEG<x>:POINTS

This command sets or queries the X/Y coordinates of all points in the designated mask segment. Mask vertices are in time/volts (currently limited to 1024

characters). The set form defines new points in the mask, replacing any existing points.

Conditions Requires option MTM.

Group Mask

Syntax
```
MASK:MASK<x>:SEG<x>:POINTS
MASK:MASK<x>:SEG<x>:POINTS?
```

Arguments MASK<x> specifies the mask test.

SEG<x> specifies the mask segment.

Examples MASK:MASK1:SEG1:POINTS "-40E-9,-200E-3,0,0,-40E-9,0" sets the X/Y coordinates of all the points in segment 1 of mask test 1.

MASK:MASK1:SEG1:POINTS? might return :MASK:MASK1:SEG1:POINTS "-40E-9,-200E-3,0,0,-40E-9,0", indicating the X/Y coordinates of all the points in segment 1 of mask test 1.

MASK:MASK<x>:SOUrce

This command sets or queries analog source for the specified mask test.

Conditions Requires option MTM.

Group Mask

Syntax
```
MASK:MASK<x>:SOUrce {CH<x>|REF<x>|MATH<x>|RFvsTime}
MASK:MASK<x>:SOUrce?
```

Arguments Arguments are the available analog sources.

MASK<x> specifies the mask test.

Examples MASK:MASK1:SOURCE CH2 sets the analog source for mask test 1 to Channel 2.

MASK:MASK1:SOURCE? might return :MASK:MASK1:SOURCE CH2, indicating the analog source for mask test 1 is Channel 2.

MASK:MASK<x>:TESt:STATE

This command sets or queries the state of the specified mask test.

Conditions Requires option MTM.

Group Mask

Syntax

```
MASK:MASK<x>:TESt:STATE {ON|OFF}
MASK:MASK<x>:TESt:STATE?
```

Arguments

MASK<x> specifies the mask test.

ON sets the mask test state to ON. When the state is ON the Pass/Fail status and hit count information are reset and the mask test is started.

OFF sets the mask test state to OFF. When the mask test completes the state is set to OFF.

Examples

MASK:MASK1:TEST:STATE ON sets the state of mask test 1 to ON.

MASK:MASK1:TEST:STATE? might return :MASK:MASK1:TEST:STATE OFF, indicating mask test 1 is OFF.

MASK:MASK<x>:TESt:STATUS? (Query Only)

This command returns the status of the specified mask test.

Conditions Requires option MTM.

Group Mask

Syntax

```
MASK:MASK<x>:TESt:STATUS?
```

Arguments

MASK<x> specifies the mask test.

Returns Return values are PASSING, FAILING, PASSED, or FAILED. When the total hit count equals or exceeds the minimum threshold the status changes from PASSING to FAILING. When the test completes the status changes to PASSED or FAILED. Default value when test state is first set ON or after CLEAR is PASSING.

Examples MASK:MASK1:TEST:STATUS? might return :MASK:MASK1:TEST:STATUS PASSED, indicating mask test 1 passed.

MASK:MASK<x>:TEST:THReshold

This command sets or queries the number of waveform violations needed for the specified mask test to change from PASS to FAIL.

Conditions Requires option MTM.

Group Mask

Syntax
```
MASK:MASK<x>:TEST:THReshold <NR1>
MASK:MASK<x>:TEST:THReshold?
```

Arguments MASK<x> specifies the mask test.

<NR1> specifies the threshold value.

Examples MASK:MASK1:TEST:THRESHOLD 10 sets the number of waveform violations to 10 for mask test 1 to fail.

MASK:MASK1:TEST:THRESHOLD? might return :MASK:MASK1:TEST:THRESHOLD 10, indicating 10 waveform violations on mask test 1.

MASK:MASK<x>:TOLerance:HORizontal

This command sets or queries the mask horizontal tolerance.

Conditions Requires option MTM.

Group Mask

Syntax
```
MASK:MASK<x>:TOLerance:HORizontal <NR3>
MASK:MASK<x>:TOLerance:HORizontal?
```

Related Commands MASK:MASK<x>:TOLerance:UPDatenow

Arguments MASK<x> specifies the mask number.

<NR3> is the tolerance in units of graticule divisions. The maximum is 1 division.

Examples MASK:MASK1:TOLerance:HORizontal .200 sets the tolerance of mask 1 to 200 mdiv or 20% of a division. The mask is not recalculated until an UPDatenow command is sent.

MASK:MASK1:TOLerance:HORizontal? might return :MASK:MASK1:TOLerance:HORizontal .200, indicating the tolerance of mask 1 is 200 mdiv.

MASK:MASK<x>:TOLerance:UPDatenow (No Query Form)

This command causes the tolerance mask to be recalculated with the current horizontal and vertical tolerances.

Conditions Requires option MTM.

Group Mask

Syntax MASK:MASK<x>:TOLerance:UPDatenow

Related Commands MASK:MASK<x>:TOLerance:HORizontal

MASK:MASK<x>:TOLerance:VERTical

Arguments MASK<x> specifies the mask number.

Examples MASK:MASK1:TOLerance:UPDatenow recalculates the tolerance of mask 1 with the current horizontal and vertical tolerance settings.

MASK:MASK<x>:TOLerance:VERTical

This command sets or queries the mask vertical tolerance.

Conditions Requires option MTM.

Group Mask

Syntax

```
MASK:MASK<x>:TOLerance:VERTical <NR3>
MASK:MASK<x>:TOLerance:VERTical?
```

Related Commands

MASK:MASK<x>:TOLerance:UPDatenow

Arguments

MASK<x> specifies the mask number.

<NR3> is the tolerance in units of graticule divisions. The maximum is 1 division.

Examples

MASK:MASK1:TOLerance:VERTical .100 sets the tolerance of mask 1 to 100 mdiv or 10% of a division. The mask is not recalculated until an UPDatenow command is sent.

MASK:MASK1:TOLerance:VERTical? might return :MASK:MASK1:TOLerance:VERTical .100, indicating the tolerance of mask 1 is 100 mdiv.

MASK:TESt:WAVEforms

This command sets or queries the number of waveform acquisitions to test during mask testing. The number of waveforms applies to all mask tests.

Conditions

Requires option MTM.

Group

Mask

Syntax

```
MASK:TESt:WAVEforms <NR1>
MASK:TESt:WAVEforms?
```

Arguments

MASK<x> specifies the mask test.

<NR1> specifies the number of waveform acquisitions.

Examples

MASK:TEST:WAVEFORMS 1k sets the number of waveform acquisitions to test to 1000.

MASK:TEST:WAVEFORMS? might return :MASK:TEST:WAVEFORMS 1k, indicating the number of waveform acquisitions to test is 1000.

MATH:ADDNew (No Query Form)

This command adds the specified math.

Group Math

Syntax `MATH:ADDNew <QString>`

Related Commands MATH:LIST?

MATH:DELete

Arguments `<QString>` is the quoted string specifying the math waveform to add. The argument is of the form "`MATH<NR1>`", where `<NR1>` is ≥1.

Examples `MATH:ADDNEW "MATH2"` adds MATH2.

MATH:DELete (No Query Form)

This command deletes the specified math.

Group Math

Syntax `MATH:DELete <QString>`

Related Commands MATH:ADDNew

MATH:LIST?

Arguments `<QString>` is a quoted string specifying the math waveform to delete. The quoted string is of the form "`MATH<NR1>`", where `<NR1>` is ≥1.

Examples `MATH:DELETE "MATH1"`, which deletes the `MATH1` waveform.

MATH:LIST? (Query Only)

This query returns a comma separated list of all currently defined math waveforms.

Group Math

Syntax `MATH:LIST?`

Related Commands MATH:ADDNew

MATH:DELete

Returns All currently defined math waveforms.

Examples `MATH:LIST?` Might return `:MATH:LIST MATH1,MATH4`, indicating `MATH1` and `MATH4` are the currently defined math waveforms.

MATH:MATH<x>:AVG:MODE

This command sets or queries the math average mode flag. If the flag is set to 1, math averaging is turned on. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:AVG:MODE {<NR1>|OFF|ON}`

Related Commands MATH:MATH<x>:AVG:WEIGht

Arguments `<NR1>` = 0 turns off average mode, and any other integer turns on average mode.

`OFF` turns off average mode.

`ON` turns on average mode.

Examples `MATH:MATH2:AVG:MODE ON` sets the average mode on.
`MATH:MATH1:AVG:MODE?` might return `:MATH:MATH1:AVG:MODE 0`, indicating average mode is off.

MATH:MATH<x>:AVG:WEIGht

This command sets or queries the number of acquisitions at which the averaging algorithm will begin exponential averaging. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:AVG:WEIGht <NR1>`

Related Commands MATH:MATH<x>:AVG:MODE

Arguments `<NR1>` is the number of acquisitions at which the averaging algorithm will begin exponential averaging.

Examples `MATH:MATH2:AVG:WEIGHT` 143 sets the number of acquisitions needed to begin averaging to 20 `MATH:MATH3:AVG:WEIGHT?` might return `:MATH:MATH3:AVG:WEIGHT` 20 indicating the number of acquisitions needed to begin averaging is 8.

MATH:MATH<x>:DEFine

This command allows you to define new waveforms using mathematical expressions. The query form of this command returns the math definition for the specified math waveform. The math waveform is specified by x.

You can specify a math expression from waveforms, measurements and scalar sources, functions, operands, and numerical constants.

Math expressions can be simple, such as Ch1, which specifies that a waveform should show the signal source of Channel 1 with no mathematical computation. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources (including other math waveforms), functions, and operands. As an example, you can enter the expression Log(Ch1+Ch2), which specifies that the signals from channels 1 and 2 are to be algebraically added, and the base 10 log of the sum is to be shown as the final math waveform.

Group Math

Syntax
```
MATH:MATH<x>:DEFine <QString>
MATH:MATH<x>:DEFine?
```

Arguments `<QString>` quoted string argument is the mathematical expression that defines the waveform. `MATH:MATH<x>:DEFINE?` is for use when the `MATH:MATH<x>:TYPE` is `ADVANCED`.

Examples `MATH:MATH2:DEFine "CH1+CH2"` adds the Channel 1 and Channel 2, defines the Math2 waveform to be Channel1 and Channel2 added together.

`MATH:MATH1:DEFine?` might return

`:MATH:MATH1:DEFine "CH2*REF2"` as the expression that defines Math 1 waveform.

MATH:MATH<x>:FUNCtion

This command sets or queries the basic math arithmetic function. The math waveform is specified by x.

NOTE. *This command does not affect the same Math equation in Advanced math (also accessed via the command `MATH:MATH<x>:DEFINE`).*

Group Math

Syntax `MATH:MATH<x>:FUNCtion {ADD|SUBtract|MULTiply|DIVide}`

Arguments ADD sets the basic math function to add.

SUBtract sets the basic math function to subtract.

MULTiply sets the basic math function to multiply.

DIVide sets the basic math function to divide.

Examples `MATH:MATH2:FUNCTION MULTIPLY` sets the basic math function to multiply.
`MATH:MATH1:FUNCTION?` might return `:MATH:MATH1:FUNCTION ADD` indicating the current basic math function is addition.

MATH:MATH<x>:GATing

This command specifies or returns the gating setting. It only applies to Math FFT plots. The math waveform is specified by x.

Group Math

Syntax
```
MATH:MATH<x>:GATing {NONE|SCREEN|CURSor}
MATH:MATH<x>:GATing?
```

Related Commands MATH:MATH<x>:TYPe

Arguments NONE turns off math gating.

SCREEN turns on gating, using the left and right edges of the screen.

CURSor limits math to the portion of the waveform between the vertical bar cursors, even if they are off screen.

Examples MATH:MATH3:GATING CURSOR sets the spectral math plot to be gated by the cursors.

MATH:MATH2:GATING? might return :MATH:MATH:2:GATING SCREEN which indicates the spectral math plot is gated by the screen.

MATH:MATH<x>:LABel:COLor

This command sets or queries color of the specified math's label. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<x>:LABel:COLor <QString>

Arguments <QString> is the color of the label. To return the color to the default color, send an empty string as in this example: :MATH:MATH1:LABEL:COLOR "".

Examples MATH:MATH3:LABEL:COLOR "GREEN" sets the Math 3 label color to green. MATH:MATH1:LABEL:COLOR? might return :MATH:MATH1:LABEL:COLOR "BLUE" indicating the color of the Math1 label is blue.

MATH:MATH<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified math label. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<x>:LABel:FONT:BOLD {<NR1>|OFF|ON}

Arguments <NR1> = 0 turns off bold, and any other integer turns on bold.

OFF turns off bold.

ON turns on bold.

Examples MATH:MATH3:FONT:BOLD ON set the math 3 label to bold.

MATH:MATH2:FONT:BOLD? might return :MATH:MATH2:FONT:BOLD 0 indicating the math 2 label is not currently bold.

MATH:MATH<x>:LABel:FONT:ITALic

This command sets or queries italic state of the specified math label. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<x>:LABel:FONT:ITALic {<NR1>|OFF|ON}

Arguments <NR1> = 0 turns off italic, and any other integer turns on italic.

OFF turns off italic.

ON turns on italic.

Examples MATH:MATH4:FONT:ITALIC ON set the math 4 label to be italic.

MATH:MATH1:FONT:ITALIC? might return :MATH:MATH1:FONT:ITALIC 0 indicating the math 1 label is not currently italic.

<NR1> is the font size of the label.

MATH:MATH<x>:LABel:FONT:SIZE

This command sets or queries font size of the specified math label. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<x>:LABel:FONT:SIZE <NR1>

Arguments

Examples MATH:MATH4:LABEL:FONT:SIZE 32 sets the math 4 label size to 32.

MATH:MATH2:LABEL:FONT:SIZE? might return :MATH:MATH2:LABEL:FONT:SIZE 14 indicating the math 2 label size is currently 14.

MATH:MATH<x>:LABel:FONT:TYPE

This command sets or queries font type of the specified math label, such as Arial or Times New Roman. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<x>:LABel:FONT:TYPE <QString>

Arguments <QString> is the name of the font type.

Examples MATH:MATH2:LABEL:FONT:TYPE "Serif" sets the math 2 label font type to Serif.

MATH:MATH3:LABEL:FONT:TYPE? might return :MATH:MATH3:LABEL:FONT:TYPE "Monospace" indicating the math 3 font type is currently Monospace.

MATH:MATH<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified math label. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}

Arguments <NR1> = 0 turns off underline, and any other integer turns on underline.

OFF turns off underline.

ON turns on underline.

Examples MATH:MATH3:FONT:UNDERLINE ON sets the math 3 label to be underlined.

MATH:MATH2:FONT:UNDERLINE? might return :MATH:MATH2:FONT:UNDERLINE 0 indicating the math 2 label is currently not underlined.

MATH:MATH<x>:LABel:NAMe

This command sets or queries the label string, which is used for annotating the math waveform on the screen. The math waveform to which the label is attached is specified by x.

Group Math

Syntax
```
MATH:MATH<x>:LABel:NAMe <QString>
MATH:MATH<x>:LABel:NAMe?
```

Arguments <QString> specifies the label to annotate the math waveform.

Examples MATH:MATH2:LABel:NAMe "PROBE POINT7" assigns "Probe point7" Math 2 waveform.

MATH:MATH2:LABel:NAMe? might return :MATH:MATH2:LABel:NAMe "Probe point7", indicating that Probe point 7 is the label for the Math 2 waveform.

MATH:MATH<x>:LABel:XPOS

This command sets or queries the X position of the specified math label. Maths are specified by x.

Group Math

Syntax
```
MATH:MATH<x>:LABel:XPOS <NR1>
MATH:MATH<x>:LABel:XPOS?
```

Related Commands MATH:MATH<x>:LABel:YPOS

Arguments <NR1> is the location (in pixels) where the label for the selected math waveform is displayed, relative to the left edge of the display.

Examples

`MATH:MATH2:LABEL:XPOS 5` moves the waveform label for the Math 2 waveform so that it begins 5 pixels to the right of the left edge of the screen.

`MATH:MATH2:LABEL:XPOS?` might return `:MATH:MATH2:LABEL:XPOS 2.5`, indicating that the waveform label for the Math 2 waveform is currently 2.5 pixels to the right of the left edge of the display.

MATH:MATH<x>:LABel:YPOS

This command sets or queries the y-position of the specified math label. The Math waveform is specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:LABel:YPOS <NR1>
MATH:MATH<x>:LABel:YPOS?
```

Related Commands

MATH:MATH<x>:LABel:XPOS

Arguments

<NR1> is the location (in pixels) where the label for the selected math waveform is displayed, relative to the baseline of the waveform.

Examples

`MATH:MATH2:LABEL:YPOS -2.5` moves the waveform label for the Math 2 waveform to 2.5 pixels below the baseline of the waveform.

`MATH:MATH2:LABEL:YPOS?` might return `:MATH:MATH2:LABEL:YPOS 0`, indicating that the waveform label for the Math 2 waveform is currently located at the baseline of the waveform.

MATH:MATH<x>:SOUrce<x>

This command sets or queries the specified math source. The source in the command can be either 1 or 2. This command sets the Basic Math components in the user interface, with two sources and a function. You would also need to set the math type to Basic to see the change in the user interface but this will not effect the programmable interface. The math waveform and source are specified by x.

Group

Math

Syntax

```
MATH:MATH<x>:SOUrce<x> {CH<x>|MATH<x>|REF<x>}
```

Related Commands MATH:MATH<x>:TYPe

MATH:MATH<x>:FUNCtion

Arguments Arguments are possible math sources. SOURCE1 and SOURCE2 are for use when the MATH:MATH<X>:TYPE is BASIC.

Examples MATH:MATH2:SOURCE1 CH1 sets the first source of math 2 to Channel 1.

MATH:MATH3:SOURCE2? might return :MATH:MATH3:SOURCE2 REF3 indicating the second source of Math 3 is Ref 3.

MATH:MATH<x>:SPECTral:HORZ

This command sets or queries the horizontal display scale of the spectral math waveform. The math waveform is specified by x.

Group Math

Syntax MATH:MATH<X>:SPECTral:HORZ {LOG|LINEAr}

Arguments LINEAr sets the SpectralMag units to linear.

LOG sets the SpectralMag units to log.

Examples MATH:MATH2:SPECTRAL:HORZ LOG sets the horizontal display scale of the spectral math waveform to log.

MATH:MATH3:SPECTRAL:HORZ? might return :MATH:MATH3:SPECTRAL:HORZ LINEAR indicating the horizontal display scale of the Math 3 spectral math waveform is currently set to linear.

MATH:MATH<x>:SPECTral:MAG

This command sets or queries the units of the SpectralMag function in the specified math definition string. The Math waveform is specified by x.

Group Math

Syntax MATH:MATH<X>:SPECTral:MAG {LINEAr|DBM}
MATH:MATH<X>:SPECTral:MAG?

Arguments `LINEAR` sets the SpectralMag units to linear.

`DBM` sets the SpectralMag units to decibels. It also sets the Ref Level Offset to a value that is the equivalent of 1 mW into 50 Ω.

Examples `MATH:MATH2:SPECTral:MAG DBM` sets the SpectralMag units for Math 2 waveform to decibels.

`MATH:MATH2:SPECTral:MAG?` might return `:MATH:MATH2:SPECTral:MAG DBM`, indicating that the SpectralMag units for Math 2 waveform are set to decibels.

MATH:MATH<x>:SPECTral:PHASE

This command sets or queries the units of a SpectralPhase function in the specified math definition string. The Math waveform is specified by x.

Group Math

Syntax
```
MATH:MATH<x>:SPECTral:PHASE {DEGrees|RADians|GROUPDelay}
MATH:MATH<x>:SPECTral:PHASE?
```

Arguments `DEGREES` sets the SpectralPhase units to degrees.

`RADIANS` sets the SpectralPhase units to radians.

`GROUPDELAY` sets the SpectralPhase units to groupdelay, which computes the derivative of unwrapped phase spectrum. Units are expressed in seconds.

Examples `MATH:MATH2:SPECTral:PHASE DEGREES` sets the SpectralPhase units for Math 2 waveform to degrees.

`MATH:MATH2:SPECTral:PHASE?` might return `:MATH:MATH2:SPECTral:PHASE RADIANS`, indicating that the SpectralPhase units for Math 2 waveform are set to radians.

MATH:MATH<x>:SPECTral:SOUrce

This command sets or queries the specified spectral math source. This only works with a math of type FFT. The math waveform is specified by x.

Group Math

Syntax

```
MATH:MATH<x>:SPECTral:SOUrce {CH<x>|MATH<x>|REF<x>}
MATH:MATH<x>:SPECTral:SOUrce?
```

Related Commands MATH:MATH<x>:TYPe

Arguments Arguments are math sources. MATH:MATH<x>:SPECTRAL:SOURCE is for use when the MATH:MATH<x>:TYPE is FFT.

Examples MATH:MATH3:SPECTRAL:SOURCE REF3 sets the source of the Spectral Math waveform to Ref 3.

MATH:MATH2:SPECTRAL:SOURCE? might return :MATH:MATH2:SPECTRAL:SOURCE CH4 indicating the Math 2 spectral source is Channel 4.

MATH:MATH<x>:SPECTral:SUPPress

This command sets or queries whether suppression threshold for the specified math waveform is enabled. This is only applied when Spectral Plot type is Phase. The math waveform is specified by x.

Group Math

Syntax

```
MATH:MATH<x>:SPECTral:SUPPress {OFF|ON|0|1}
MATH:MATH<x>:SPECTral:SUPPress?
```

Related Commands MATH:MATH<x>:SPECTral:TYPE

Arguments 0 disables suppression threshold for the specified math waveform.

1 enables suppression threshold for the specified math waveform.

ON enables suppression threshold for the specified math waveform.

OFF disables suppression threshold for the specified math waveform.

Examples MATH:MATH3:SPECTRAL:SUPPRESS ON enables the suppression threshold on the Math 3 spectral waveform.

MATH:MATH2:SPECTRAL:SUPPRESS? might return :MATH:MATH2:SPECTRAL:SUPPRESS 0 indicating the suppression threshold is disabled on the Math 2 spectral waveform.

MATH:MATH<x>:SPECTral:SUPPress:VALue

This command sets or queries in volts the value of suppression threshold of the specified math waveform. This requires the Spectral type to be Phase and the Suppression to be enabled for this PI command to have any affect. The math waveform is specified by x.

Group Math

Syntax

```
MATH:MATH<x>:SPECTral:SUPPress:VALue <NR3>
MATH:MATH<x>:SPECTral:SUPPress:VALue?
```

Related Commands MATH:MATH<x>:SPECTral:TYPE

MATH:MATH<x>:SPECTral:SUPPress

Arguments <NR3> is the value of suppression threshold of the specified math waveform in volts.

Examples MATH:MATH3:SPECTRAL:SUPPRESS:VALUE 100.0E-3 sets the suppression threshold of Math 3 to 100 mV.

MATH:MATH1:SPECTRAL:SUPPRESS:VALUE? might return MATH:MATH1:SPECTRAL:SUPPRESS:VALUE 10.0000E+3 indicating the suppression threshold of Math 1 is currently set to 10 kV.

MATH:MATH<x>:SPECTral:TYPE

This command sets or queries the FFT type selected for spectral analysis. The math waveform is specified by x.

Group Math

Syntax

```
MATH:MATH<x>:SPECTral:TYPE {MAGNitude|PHASe|REAL|IMAGinary}
MATH:MATH<x>:SPECTral:TYPE?
```

Arguments MAGNitude specifies the magnitude spectral function.

PHASe specifies the phase spectral function.

REAL specifies the real spectral function.

IMAGinary specifies the imaginary spectral function.

Examples `MATH:MATH1:SPECTral:TYPE REAL` specifies the real spectral function.

`MATH:MATH1:SPECTral:TYPE?` might return `:MATH:MATH1:SPECTRAL:TYPE MAGNITUDE` indicating the math is the magnitude spectral function.

MATH:MATH<x>:SPECTral:UNWRap

This command sets or queries whether phase unwrap of the spectral analyzer output data is enabled. The Math waveform is specified by x.

Group Math

Syntax

```
MATH:MATH<x>:SPECTral:UNWRap {OFF|ON|0|1}
MATH:MATH<x>:SPECTral:UNWRap?
```

Arguments `0` disables phase unwrap for the specified math waveform.

`1` enables phase unwrap for the specified math waveform.

`ON` enables phase unwrap for the specified math waveform.

`OFF` disables phase unwrap for the specified math waveform.

Examples `MATH1:SPECTRAL:UNWRAP ON` enables phase unwrap of the spectral analyzer output data.

`MATH1:SPECTRAL:UNWRAP?` might return `:MATH1:SPECTRAL:UNWRAP 0`, indicating that the phase unwrap of the spectral analyzer output data is disabled.

MATH:MATH<x>:SPECTral:UNWRap:DEGrees

This command sets or queries how many degrees adjacent phase values can jump before being unwrapped. This requires the Spectral type to be Phase and the `UNWRAP` to be enabled for this PI command to have any affect. The math waveform is specified by x.

Group Math

Syntax

```
MATH:MATH<x>:SPECTral:UNWRap:DEGrees <NR3>
MATH:MATH<x>:SPECTral:UNWRap:DEGrees?
```

Related Commands MATH:MATH<x>:SPECTral:TYPE

MATH:MATH<x>:SPECTral:UNWRap

Arguments <NR3> is the value of unwrap phase in degrees.

Examples MATH:MATH2:SPECTRAL:UNWRAP:DEGREES 90 sets the unwrap phase of the spectral Math 2 to 90 degrees.

MATH:MATH2:SPECTRAL:UNWRAP:DEGREES? might return :MATH:MATH2:SPECTRAL:UNWRAP:DEGREES 180 indicating unwrap phase of the spectral math 3 waveform is 180 degrees.

MATH:MATH<x>:SPECTral:WINdow

This command sets or queries the window function used to apply the specified FFT window to the input data for the specified math waveform. The Math waveform is specified by x. A spectral window determines what the filter shape of the spectral analyzer will be in the frequency domain. It can be described by a mathematical function that is multiplied point-by-point times the input data to the spectral analyzer.

Following is a list of arguments that specify the window function used to multiply the input data. The windows are listed in the order of their ability to resolve frequencies (resolution bandwidth).

Group Math

Syntax
```
MATH:MATH<x>:SPECTral:WINdow {RECTANGular|HAMMing|
HANNing|BLACKMANHarris|KAISERBessel|GAUSSian|
FLATTOP2|TEKEXPonential}
MATH:MATH<x>:SPECTral:WINdow?
```

Related Commands MATH:MATH<x>:TYPe

Arguments RECTANGular window function is equivalent to multiplying all gate data by one.

HAMMing window function is based on a cosine series.

HANNing window function is based on a cosine series.

BLACKMANHarris window function is based on a cosine series.

KAISERBessel window function is based on a cosine series.

GAUSSian window function has the best localization characteristics in the joint time/frequency plane

FLATTOP2 window function is a cosine series window with a flattened frequency response lobe.

TEKEXPonential window has an exponential nonsymmetrical shape in the time domain and a triangular shape in the frequency domain.

Examples

MATH2:SPECTRAL:WINDOW GAUSSIAN applies a Gaussian window to the spectral analyzer input data.

MATH2:SPECTRAL:WINDOW? might return :MATH2:SPECTRAL:WINDOW BLACKMANHarris, indicating that the window function used to multiply the spectral analyzer input data is the BLACKMANHarris function.

MATH:MATH<x>:TYPe

This command sets or queries the math type. The math waveform is specified by x.

Group

Math

Syntax

MATH:MATH<x>:TYPe {BASic|FFT|ADVanced}

Arguments

BASic set the type to basic math.

FFT sets the type to FFT math, which can use any live analog or reference waveform in the time domain. NOTE. You can also use FFT as part of a math expression by declaring the type

ADVanced. See examples for the command MATH:MATH<x>:DEFine.

ADVanced sets the type to advanced math.

Examples

MATH:MATH2:TYPE BASIC sets the type of Math 2 to basic.

MATH:MATH2:TYPE? might return :MATH:MATH2:TYPE FFT indicating the type of Math 4 is currently FFT.

MATH:MATH<x>:VUNIT

This command specifies or returns the math custom vertical units. The math waveform is specified by x.

Group Math

Syntax `MATH:MATH<x>:VUNIT <QString>`

Arguments `<QString>` is the custom vertical units.

Examples `MATH:MATH2:VUNIT` "Small"

`MATH:MATH2:VUNIT?` might return `:MATH:MATH2:VUNIT` "Large" indicating the Math 2 vertical unit is set to "Large".

MATHArbflt<x>:FILepath

This command or query sets the file path for a file of filter coefficients for the specified arbitrary filter. Setting a path will read that file and load the filter for ARBFLT<x>. Access these filters using a Math with an expression of the form "ARBFlt<x>()".

Group Math

Syntax
```
MATHArbflt<x>:FILepath <QString>
MATHArbflt<x>:FILepath?
```

Arguments `<QString>` specifies the path to the file of filter coefficients.

Examples `MATHARBFLT8:FILEPATH "c:/myfilters/20mhz.flt"` reads filter coefficients from the file and loads the filter for ARBFLT8.

`MATHARBFLT1:FILEPATH?` might return `:MATHARBFLT1:FILEPATH "C:\Users\Public\Tektronix\TekScope\Math Arbitrary Filters\LowPass-Norm\lowpass_0.bw.flt"`, indicating that the path to the file of filter coefficients is set to "C:\Users\Public\Tektronix\TekScope\Math Arbitrary Filters\LowPass-Norm\lowpass_0.bw.flt".

MEASTABle:ADDNew (No Query Form)

This command adds a new measurement results view table to the scope application. The results view table can be named through the argument sent to the command.

Group Measurement

Syntax `MEASTABle:ADDNew <QString>`

Arguments `<QString>` specifies a new measurement results view table to the scope application.

Examples `MEASTABle:ADDNew "TABLE1"` adds a new table view by name TABLE1.

MEASTABle:DELETE (No Query Form)

This command removes the requested measurement results view table from the scope application.

Group Measurement

Syntax `MEASTABle:DELETE <QString>`

Arguments `<QString>` specifies a measurement results view table to remove.

Examples `MEASTABle:DELETE "TABLE1"` removes TABLE1 from the view.

MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MAXimum? (Query Only)

This query returns the maximum value of the measurement specified by the string, for all acquisitions.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MAXimum? <Qstring>`

Arguments `<Qstring>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS

|RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS
|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK
|RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK
|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY
|RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY
|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT
|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK
|RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MAXimum? "INPUT"` returns the maximum value of measurement `INPUT`, for all acquisitions.

MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MEAN? (Query Only)

This query returns the mean value of the measurement specified by the string, for all acquisitions.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MEAN? <Qstring>`

Arguments `<Qstring>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|
OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT
|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS
|RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS
|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK
|RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK
|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY
|RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY
|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT
|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK
|RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MEAN? "RAIL1DPMOVERSHOOT"` returns the mean value of measurement `RAIL1DPMOVERSHOOT`, for all acquisitions.

MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MINimum? (Query Only)

This query returns the minimum value of the measurement specified by the string, for all acquisitions.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MINimum? <Qstring>`

Arguments `<Qstring>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS |RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS |RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK |RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK |RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY |RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY |RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT |RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK |RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:MINimum? "OUTPUT7"` returns the minimum value of measurement `OUTPUT7`, for all acquisitions.

MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:PK2PK? (Query Only)

This query returns the peak-to-peak value of the measurement specified by the string, for all acquisitions.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:PK2PK? <Qstring>`

Arguments <Qstring> = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5| OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT |RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS |RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS |RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK |RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK |RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY |RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY |RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT |RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK |RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:PK2PK? "OUTPUT4"` returns the peak-to-peak value of measurement `OUTPUT4`, for all acquisitions.

MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:POPUlation? (Query Only)

This query returns the population value of the measurement specified by the string, for all acquisitions.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:POPUlation? <Qstring>`

Arguments <Qstring> = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5| OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT |RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS |RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS |RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK |RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK |RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY |RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY |RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT |RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK |RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:POPUlation? "RAIL1DPMUNDERSHOOT"` returns the population value of measurement `RAIL1DPMUNDERSHOOT`, for all acquisitions.

MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:STDDev? (Query Only)

This query returns the standard deviation value of the measurement specified by the string, for all acquisitions.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:STDDev? <Qstring>`

Arguments `<Qstring>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS |RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS |RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK |RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK |RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY |RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY |RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT |RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK |RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:STDDev? "RAIL7PK2PK"` returns the standard deviation value of measurement `RAIL7PK2PK`, for all acquisitions.

MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MAXimum? (Query Only)

This query returns the maximum value of the measurement specified by the string, for the current acquisition.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MAXimum? <Qstring>`

Arguments <Qstring> = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS |RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS |RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK |RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK |RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY |RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY |RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT |RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK |RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MAXimum? "INPUT"` returns the maximum value of measurement `INPUT`, for the current acquisition.

MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MEAN? (Query Only)

This query returns the mean value of the measurement specified by the string, for the current acquisition.

Conditions Requires option 5-DPM (5 Series MSO instruments) or option 5-IMDA (MSO58 and MSO56 instruments) or 6-DPM (6 Series MSO instrument)

Group DPM, Measurement, IMDA

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MEAN? <Qstring>`

Arguments <Qstring> = INPUT |OUTPUT1 |OUTPUT2 |OUTPUT3 |OUTPUT4 |OUTPUT5 |OUTPUT6 |OUTPUT7 |RAIL1DPMOVERSHOOT |RAIL1DPMUNDERSHOOT |RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS |RAIL1RMSFULL |RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS |RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK |RAIL3RMS |RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK |RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY |RAIL5PK2PK |RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY |RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT |RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK |RAIL7RMS |RAIL7RMSFULL |L2LPH1VRMS |L2LPH1VCFactor |L2LPH1TruePwr |L2LPH1RePwr |L2LPH1AppPwr |L2LPH1PwrFactor |L2LPH1PhaseDiff |L2LPH1VPhase |L2LPH2VRMS |L2LPH2VCFactor |L2LPH2TruePwr |L2LPH2RePwr |L2LPH2AppPwr |L2LPH2PwrFactor |L2LPH2PhaseDiff |L2LPH2VPhase |L2LPH3VRMS |L2LPH3VCFactor |L2LPH3TruePwr |L2LPH3RePwr |L2LPH3AppPwr |L2LPH3PwrFactor |L2LPH3PhaseDiff |L2LPH3VPhase

|L2NPH1VRMS |L2NPH1VCFactor |L2NPH1TruePwr |L2NPH1RePwr
|L2NPH1AppPwr |L2NPH1PwrFactor |L2NPH1PhaseDiff |L2NPH1VPhase
|L2NPH2VRMS |L2NPH2VCFactor |L2NPH2TruePwr |L2NPH2RePwr
|L2NPH2AppPwr |L2NPH2PwrFactor |L2NPH2PhaseDiff |L2NPH2VPhase
|L2NPH3VRMS |L2NPH3VCFactor |L2NPH3TruePwr |L2NPH3RePwr
|L2NPH3AppPwr |L2NPH3PwrFactor |L2NPH3PhaseDiff |L2NPH3VPhase
|PH1IRMS |PH1IPhase |PH1ICFactor |PH2IRMS |PH2IPhase |PH2ICFactor
|PH3IRMS |PH3IPhase |PH3ICFactor |Frequency |TotalPower
|L2LPH1F1Mag |L2LPH1F3Mag |L2LPH1THDF |L2LPH1THDR
|L2LPH1RMS |L2LPH1IRMS |L2LPH1Status |L2LPH1HarmonicsNumber
|L2LPH1Frequency |L2LPH1MagnitudeAbs |L2LPH1MagnitudePct
|L2LPH1Phase |L2LPH1Limits |L2LPH1PassFail |L2LPH1Margin
|L2LPH2F1Mag |L2LPH2F3Mag |L2LPH2THDF |L2LPH2THDR
|L2LPH2RMS |L2LPH2IRMS |L2LPH2Status |L2LPH2HarmonicsNumber
|L2LPH2Frequency |L2LPH2MagnitudeAbs |L2LPH2MagnitudePct
|L2LPH2Phase |L2LPH2Limits |L2LPH2PassFail |L2LPH2Margin
|L2LPH3F1Mag |L2LPH3F3Mag |L2LPH3THDF |L2LPH3THDR |L2LPH3RMS
|L2LPH3IRMS |L2LPH3Status |L2LPH3HarmonicsNumber |L2LPH3Frequency
|L2LPH3MagnitudeAbs |L2LPH3MagnitudePct |L2LPH3Phase |L2LPH3Limits
|L2LPH3PassFail |L2LPH3Margin |L2NPH1F1Mag |L2NPH1F3Mag
|L2NPH1THDF |L2NPH1THDR |L2NPH1RMS |L2NPH1IRMS
|L2NPH1Status |L2NPH1HarmonicsNumber |L2NPH1Frequency
|L2NPH1MagnitudeAbs |L2NPH1MagnitudePct |L2NPH1Phase|L2NPH1Limits
|L2NPH1PassFail|L2NPH1Margin |L2NPH2F1Mag|L2NPH2F3Mag
|L2NPH2THDF|L2NPH2THDR |L2NPH2RMS|L2NPH2IRMS
|L2NPH2Status |L2NPH2HarmonicsNumber |L2NPH2Frequency
|L2NPH2MagnitudeAbs |L2NPH2MagnitudePct |L2NPH2Phase|L2NPH2Limits
|L2NPH2PassFail|L2NPH2Margin |L2NPH3F1Mag |L2NPH3F3Mag
|L2NPH3THDF |L2NPH3THDR |L2NPH3RMS |L2NPH3IRMS |L2NPH3Status
|L2NPH3HarmonicsNumber |L2NPH3Frequency |L2NPH3MagnitudeAbs
|L2NPH3MagnitudePct |L2NPH3Phase |L2NPH3Limits |L2NPH3PassFail
|L2NPH3Margin |ORDER|PH1INPwr |PH1OUTPwr |PH1Efficiency |PH2INPwr
|PH2OUTPwr |PH2Efficiency |TotalEfficiency |PH1LRIPRMS |PH1LRIPPK2PK
|PH2LRIPRMS |PH2LRIPPK2PK |PH3LRIPRMS |PH3LRIPPK2PK
|PH1SWRIPRMS |PH1SWRIPPK2PK |PH2SWRIPRMS |PH2SWRIPPK2PK
|PH3SWRIPRMS |PH3SWRIPPK2PK

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MEAN? "RAIL1DPMOVERSHOOT"` returns the mean value of measurement `RAIL1DPMOVERSHOOT`, for the current acquisition.

MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MINimum? (Query Only)

This query returns the minimum value of the measurement specified by the string, for the current acquisition.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MINimum? <Qstring>`

Arguments <Qstring> = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS |RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS |RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK |RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK |RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY |RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY |RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT |RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK |RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:MINimum? "OUTPUT7"` returns the minimum value of measurement `OUTPUT7`, for the current acquisition.

MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:PK2PK? (Query Only)

This query returns the peak-to-peak value of the measurement specified by the string, for the current acquisition.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:PK2PK? <Qstring>`

Arguments <Qstring> = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS |RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS |RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK |RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK

|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY
|RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY
|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT
|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK
|RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:PK2PK? "OUTPUT4"` returns the peak-to-peak value of measurement `OUTPUT4`, for the current acquisition.

MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:POPUlation? (Query Only)

This query returns the population value of the measurement specified by the string, for the current acquisition.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax

```
MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:POPUlation?
<Qstring>
```

Arguments `<Qstring>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|
OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT
|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS
|RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS
|RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK
|RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK
|RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY
|RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY
|RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT
|RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK
|RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:POPUlation? "RAIL1DPMUNDERSHOOT` returns the population value of measurement `RAIL1DPMUNDERSHOOT`, for the current acquisition.

MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:STDDev? (Query Only)

This query returns the standard deviation value of the measurement specified by the string, for the current acquisition.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:STDDev? <Qstring>`

Arguments `<Qstring>` = INPUT|OUTPUT1|OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7| RAIL1DPMOVERSHOOT|RAIL1DPMUNDERSHOOT|RAIL1FREQUENCY |RAIL1PK2PK |RAIL1RMS |RAIL1RMSFULL|RAIL2FREQUENCY |RAIL2PK2PK |RAIL2RMS |RAIL2RMSFULL |RAIL3FREQUENCY |RAIL3PK2PK |RAIL3RMS|RAIL3RMSFULL |RAIL4FREQUENCY |RAIL4PK2PK |RAIL4RMS |RAIL4RMSFULL |RAIL5FREQUENCY |RAIL5PK2PK|RAIL5RMS |RAIL5RMSFULL |RAIL6FREQUENCY |RAIL6PK2PK |RAIL6RMS |RAIL6RMSFULL |RAIL7DPMOVERSHOOT |RAIL7DPMUNDERSHOOT |RAIL7FREQUENCY |RAIL7PK2PK |RAIL7RMS |RAIL7RMSFULL

Examples `MEASU:MEAS1:SUBGROUP:RESUlts:CURRentacq:STDDev? "RAIL7PK2PK"` returns the standard deviation value of measurement `RAIL7PK2PK`, for the current acquisition.

MEASUrement? (Query Only)

This query-only command returns all measurement parameters.

Group Measurement

Syntax `MEASUrement?`

Examples `MEASUREMENT?` might return `:MEASUREMENT:GATING OFF;IMMED:TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:IMMED:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2`

```
RISE;DIRECTION FORWARDS;:MEASUREMENT:IMMED:REFLEVEL:METHOD
PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2
0.0000;:MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 90.0000;LOW
10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:IMMED:METHOD
HISTOGRAM;NOISE HIGH;:MEASUREMENT:MEAS1:STATE
0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2
CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS1:SOURCE2:SIGTYPE
PULSE;:MEASUREMENT:MEAS1:DELAY:EDGE1 RISE;EDGE2
RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS1:REFLEVEL:METHOD
PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2
0.0000;:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 90.0000;LOW
10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:MEAS1:METHOD
HISTOGRAM;NOISE HIGH;:MEASUREMENT:MEAS2:STATE
0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2
CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS2:SOURCE2:SIGTYPE
PULSE;:MEASUREMENT:MEAS2:DELAY:EDGE1 RISE;EDGE2
RISE;DIRECTION.
```

MEASUrement:ADDMEAS (No Query Form)

This command adds a measurement.

Conditions Refer to the Arguments list for measurements are require the DJA option or are not available on 4 Series MSO instruments.

Group Measurement, IMDA

Syntax

```
MEASUrement:ADDMEAS {ACCOMMONMODE |ACRMS |AMPlITUDE
|AREA |BASE |BITAMPLITUDE |BITHIGH |BITLOW |BURSTWIDTH
|COMMONMODE |DATARATE |DCD |DDJ |DDRAOS |DDRAOSPERTCK
|DDRAOSPERUI |DDRAUS |DDRAUSPERTCK |DDRAUSPERUI |DDRHOLDDIFF
|DDRSETUPDIFF |DDRTCHABS |DDRTCHAVERAGE |DDRTCKAVERAGE
|DDRTCLABS |DDRTCLAVERAGE |DDRTERRMN |DDRTERRN |DDRTJITCC
|DDRTJITDUTY |DDRTJITPER |DDRTPST |DDRTRPRE |DDRTWPRE
|DDRVIXAC |DDRTDQSCK |DELAY |DJ |DJDIRAC |EYEHIGH |EYELOW
|FALLSLEWRATE |FALLTIME |FREQUENCY |F2 |F4 |F8 |HEIGHT
|HEIGHTBER |HIGH |HIGHTIME |HOLD |IMDAPOWERQUALITY
|IMDAHARMONICS |IMDAINPUTVOLTAGE |IMDAINPUTCURRENT
|IMDAINPUTPOWER |IMDAPHASORDIAGRAM |IMDAEFFICIENCY
|IMDALINERIPPLE |IMDASWITCHRIPPLE |JITTERSUMMARY |J2 |J9
|LOW |LOWTIME |MAXIMUM |MEAN |MINIMUM |NDUty |NOVERSHOOT
|NPERIOD |NPJ |NWIDTTH |PDUTY |PERIOD |PHASE |PHASENOISE |PJ
|PK2Pk |POVERSHOOT |PWIDTH |QFACTOR |RISESLEWRATE |RISETIME
|RJ |RJDIRAC |RMS |SETUP |SKEW |SRJ |SSCFREQDEV |SSCMODRATE
|TIE |TIMEOUTSIDELEVEL |TIMETOMAX |TIMETOMIN |TJBER
|TNTRATIO |TOP |UNITINTERVAL |VDIFFXOVR |WIDTH |WIDTHBER}
```

Arguments

ACCOMMONMODE AC Common Mode (Pk-Pk) is the peak-to-peak of the common mode voltage of two sources. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

ACRMS (AC RMS) is the true Root Mean Square of the data points, about the Mean. This measurement can be made across the entire record, or on each cycle in the record.

AMPLITUDE is the difference between the Top value and the Base value. This measurement can be made across the entire record, or on each cycle in the record.

Amplitude = High - Low

AREA is the area under the curve, calculated by integrating the data points. The area measured above ground is positive. The area measured below ground is negative. This measurement can be made across the entire record, or on each cycle in the record.

BASE is the most common data value below the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

BITAMPLITUDE (Bit Amplitude) is the difference between the amplitudes of the 1 bit and the 0 bit surrounding a transition. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each transition bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BITHIGH (Bit High) is the amplitude of a 1 bit. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BITLOW (Bit Low) is the amplitude of a 0 bit. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BURSTWIDTH (Burst Width) is the duration of a series of adjacent crossings of the Mid reference level (RM). Bursts are separated by a user-defined idle time (tI). This measurement is made on each burst in the record.

COMMONMODE (DC Common Mode) is the arithmetic mean of the common mode voltage of two sources. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DATARATE (Data Rate) is the reciprocal of Unit Interval. This measurement is made on each bit in the record.

DCD (duty cycle distortion) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the signal polarity. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DDJ (data dependent jitter) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the data pattern in the waveform. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DDRAOS (area above signal) is the total area of the signal above a specified reference level. This measurement is made across the entire record.

DDRAOSPERTCK (area over signal for tCK events) is the total area of the signal above a specified reference level calculated over consecutive tCK intervals. It is applicable to clock and address/command waveforms.

DDRAOSPERUI (area over signal for UI events) is the total area of the signal above a specified reference level calculated over consecutive unit intervals. It is applicable to data and data strobe waveforms.

DDRAUS (area under signal) is the total area of the signal below a specified reference level. This measurement is made across the entire record.

DDRAUSPERTCK (area under signal for tCK events) is the total area of the signal below a specified reference level calculated over consecutive tCK intervals. It is applicable to clock and address/command waveforms.

DDRAUSPERUI (area under signal for UI events) is the total area of the signal below a specified reference level calculated over consecutive unit intervals. It is applicable to data and data strobe waveforms.

DDRHOLDDIFF (hold difference) is the elapsed time between the specified edge of a single-ended clock waveform and the specified edge of a differential data waveform. The measurement uses the closest respective waveform edges that fall within the range limits.This measurement is made across the entire record.

DDRSETUPDIFF (setup difference) is the elapsed time between the specified edge of a single-ended clock waveform and when the specified edge of a differential data waveform crosses a specified level. The measurement uses the closest respective waveform edges that fall within the range limits. This measurement is made across the entire record.

DDRTCHABS (absolute high pulse width) is the absolute value of the high pulse width as measured from one rising edge to the next falling edge.

DDRTCHAVERAGE (average high pulse width) is the average value of the high pulse width as measured from one rising edge to the next falling edge, across 200 consecutive cycles. This measurement is made across the entire record.

DDRTCKAVERAGE (average clock period) is the average clock period calculated from rising edge to rising edge, across 200 consecutive cycles. This measurement is made across the entire record.

DDRTCLABS (absolute low pulse width) is the absolute value of the low pulse width as measured from a falling edge to the next rising edge. This measurement is made across the entire record.

DDRTCLAVERAGE (average low pulse width) is the average value of the low pulse width as measured from one falling edge to the next rising edge, across 200 consecutive cycles.

DDRTERRMN (cumulative error) is the cumulative error across multiple consecutive defined cycles from tCK(avg).

DDRTERRN (cumulative error) is the cumulative error across specified consecutive cycles from tCK(avg). In other words, this measures the time difference between the sum of the clock period from a 200 cycle window and n times tCK(avg).

DDRTJITCC (cycle to cycle jitter period) is the absolute difference in clock period between two consecutive clock cycles. This measurement is made across the entire record.

DDRTJITDUTY (half period jitter) is the largest elapsed time between tCH and tCH(avg), and tCL and tCL(avg), over 200 consecutive cycles.

DDRTJITPER (clock period jitter) is the largest deviation of any tCK signal from tCK(avg). This measurement is made across the entire record.

DDRTPST (read/write burst postamble) is the width of the Read or Write burst postamble, measured from the last falling edge of the mid reference level to the start of an undriven state. This measurement is made across the entire record.

DDRTRPRE (read burst preamble) is the width of the Read burst preamble, measured from exiting tristate levels to the first driving edge of the differential strobe. This measurement is made across the entire record.

DDRTWPRE is the width of the Write burst preamble, measured from exiting tristate levels to the first driving edge of the differential strobe. This measurement is made across the entire record.

DDRVIXAC is the differential input cross-point voltage measured from the true state transition (and it's compliment) to a specified reference level, measured on a single-ended signal.

DDRTDQSCK is the strobe output access time, measured between the rising edge of the clock and before or after the differential strobe Read preamble time. Signal edges are determined by the mid-ref threshold level settings.

DELay is the time between the specified Mid reference level (RM) crossing on one source to a specified Mid reference level (RM) crossing on a second source. This measurement is made on the first occurrence in the record.

DJ (deterministic jitter) is the peak-to-peak amplitude of all timing errors that exhibit deterministic behavior. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DJDIRAC (dual-dirac deterministic jitter) is deterministic jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

EYEHIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

EYELOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

FALLSLEWRATE (Falling Slew Rate) is the rate of change in voltage as an edge transitions from the Top reference level (RT) to the Bottom reference level (RB). This measurement is made on each cycle in the record.

FALLTIME (Fall Time) is the time required for an edge to fall from the Top reference level (RT) to the Base reference level (RB). This measurement is made on each cycle in the record.

FREQuency is the reciprocal of Period. This measurement is made on each cycle in the record.

F2 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 2. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

F4 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 4. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

F8 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 8. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record.

HEIGHT (Eye Height) is the minimum vertical eye opening at the center of the recovered unit interval. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HEIGHTBER (Eye Height@BER) is the predicted vertical eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HIGH

HIGHTIME (High Time) is the time the signal remains above the Top reference level (RT). This measurement is made on each cycle in the record.

HOLD (Hold Time) is the time between the specified Mid reference level crossing (RM) on the Clock source to the closest specified Mid reference level (RM) crossing on the Data source. This measurement is made on each specified Clock edge in the record.

IMDAPOWERQUALITY measures the operating Frequency, RMS values of the voltage and current, Crest Factors of the voltage and current, Real Power, Reactive Power, Apparent Power, Power Factor, and Phase Angle of the AC signal. Plots the Phasor diagram. This measurement requires the IMDA option.

IMDAHARMONICS measures the signal amplitudes at the fundamental line frequency and its harmonics. Measures the RMS amplitude and Total Harmonic Distortion of the signal. Plots the Harmonics Bar Graph. This measurement requires the IMDA option.

IMDAINPUTVOLTAGE measures the RMS value of the input voltage for all phases based on the configured wiring. This measurement requires the IMDA option.

IMDAINPUTCURRENT measures the RMS value of the input current for all phases based on the configured wiring. This measurement requires the IMDA option.

IMDAINPUTPOWER measures the RMS and peak-to-peak values of the AC signal. This measurement requires the IMDA option.

IMDAPHASORDIAGRAM plots Vrms and Irms and phase angles between voltage(s) and current(s) per winding pair. This measurement requires the IMDA option.

IMDAEFFICIENCY measures the ratio of output power to input power per phase for a AC input and Drive output. Measures the total efficiency of the system. This measurement requires the IMDA option.

IMDALINERIPPLE measures the RMS value of the input power for all phases based on the configured wiring. Measures the True / Apparent / Reactive Power for all the phases. This measurement requires the IMDA option.

IMDASWITCHRIPPLEJITTERSUMMARY measures the RMS and peak-to-peak values of the switching signal. This measurement requires the IMDA option.

JITTERSUMMARY (Jitter Summary) is a group consisting of the following measurements: TIE, TJ@BER, Eye Width@BER, Eye Height@BER, RJ-δδ, DJ-δδ, PJ, DDJ, DcD, F/2, F/4, F/8. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

J2 is the total jitter at a bit error rate of 2.5e-3 (TJ@2.5e-3). This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

J9 is the total jitter at a bit error rate of 2.5e-10 (TJ@2.5e-10). This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

LOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record.

LOWTIME (Low Time) is the time the signal remains below the Base reference level (RB). This measurement is made on each cycle in the record.

MAXimum is the maximum data point. This measurement can be made across the entire record, or on each cycle in the record.

MEAN is the arithmetic mean of the data points. This measurement can be made across the entire record, or on each cycle in the record.

MINImum is the minimum data point. This measurement can be made across the entire record, or on each cycle in the record.

NDUty (Negative Duty Cycle) is the ratio of the Negative Pulse Width to the Period. This measurement is made on each cycle in the record.

Negative Duty Cycle = (Negative Width) / Period × 100%

NPERIOD (Duration N-Periods) is the time required to complete N cycles. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

NPJ (non-periodic jitter) is the portion of the BUJ (bounded uncorrelated jitter) that is random. BUJ excludes DDJ, DCD and RJ. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

NOVershoot (Negative Overshoot) is the difference between Minimum and Base, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

Negative Overshoot = (Base - Minimum) / Amplitude × 100%)

NWIdth (Negative Pulse Width) is the time the signal remains below the Mid reference level (RM). This measurement is made on each cycle in the record.

PDUTY (Positive Duty Cycle) is the ratio of the Positive Pulse Width to the Period. This measurement is made on each cycle in the record.

Positive Duty Cycle = (Positive Width)/Period × 100%

PERIOD is the time required to complete a cycle. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

PHASE is the ratio of the Skew between two sources to the Period of the first source. This measurement is made on each cycle in the record.

PHASENOISE (Phase Noise) is the RMS magnitude of all integrated jitter falling within a user specified offset range of the fundamental clock frequency. This measurement is made across the entire record. This measurement is not available on a 4 Series MSO instrument.

PJ (periodic jitter) is the peak-to-peak amplitude of the uncorrelated sinusoidal components of the deterministic jitter. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

PK2Pk (Peak-to-peak) is the difference between Maximum and Minimum. This measurement can be made across the entire record, or on each cycle in the record.

POVERSHOOT (Positive Overshoot) is the difference between Maximum and Top, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

Positive Overshoot = (Maximum - Top) / Amplitude ×100%

PWIDTH (Positive Pulse Width) is the time the signal remains above the Mid reference level (RM). This measurement is made on each cycle in the record.

QFACTOR (Q-Factor) is the ratio of the vertical eye opening to RMS vertical noise measured at a user specified location within the recovered unit interval. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RISESLEWRATE (Rising Slew Rate) is the rate of change in voltage as an edge transitions from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RISETIME Rise Time is the time required for an edge to rise from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RJ (random jitter) is the RMS magnitude of all random timing errors following a Gaussian distribution. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RJDIRAC (dual-dirac random jitter) is random jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RMS is the true Root Mean Square of the data points. This measurement can be made across the entire record, or on each cycle in the record.

SRJ (sub-rate jitter) is the composite jitter due to periodic components at 1/2, 1/4 and 1/8 of the data rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SSCFREQDEV (SSC Frequency Deviation) is the spread spectrum clock frequency deviation. This measurement enables a time trend plot of the spread spectrum clock modulation profile. This measurement is made on each cycle in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SSCMODRATE (SSC Modulation Rate) is the modulating frequency of a spread spectrum clock. This measurement is made on each cycle in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SETUP (Setup Time) is the time between the specified Mid reference level (RM) crossing on the Data source to the closest specified Mid reference level (RM) crossing on the Clock source. This measurement is made on each specified Clock edge in the record.

SKEW Skew is the time between the specified Mid reference level (RM) crossing on one source to the following specified Mid reference level (RM) crossing on a second source. This measurement is made on each cycle in the record.

TIE (time interval error) is the difference, in time, between an edge in the source waveform and the corresponding edge in a recovered reference clock. This measurement is made on each edge in the waveform. This measurement is not available on a 4 Series MSO instrument.

TIMEOUTSIDELEVEL Time Outside Level is the time the signal remains above the Top reference level (RT) and/or below the Base reference level (RB). This measurement is made on each occurrence in the record.

TIMETOMAX Time to Max is the amount of time from the trigger point to the maximum data point. This measurement can be made across the entire record or on each cycle in the record.

TIMETOMIN Time to Min is the amount of time from the trigger point to the minimum data point. This measurement can be made across the entire record or on each cycle in the record.

TJBER (total jitter at a specified bit error rate) is the predicted peak-to-peak amplitude of jitter that will only be exceeded with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

TNTRATIO T/nT Ratio is the ratio of a non-transition bit voltage (2nd and subsequent bit voltage after a transition) to its nearest preceding transition bit voltage (1st bit voltage after the transition). Bit voltages are measured at the interpolated midpoint of the recovered unit interval. This measurement is made on

each non-transition bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

TOP is the most common data value above the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

UNITINTERVAL (Unit Interval) is the time difference between two successive bits. This measurement is made on each bit in the record.

VDIFFXOVR (Differential Crossover) is the voltage level of a differential signal pair at the crossover points. This measurement is made at each crossover point in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

WIDTH (Eye Width) is the minimum horizontal eye opening at the user specified reference level. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

WIDTHBER (Eye Width@BER) is the predicted horizontal eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

Examples `MEASUREMENT:ADDMEAS FREQUENCY` adds a frequency measurement.

MEASUrement:ADDNew (No Query Form)

This command adds the specified measurement.

Group Measurement

Syntax `MEASUrement:ADDNew "QString"`

Arguments `"QString"` is the measurement to add. The argument is of the form "MEAS<NR1>" where NR1 ≥ 1.

Examples `MEASUrement:ADDNew "MEAS11"` adds measurement 11.

MEASUrement:ANNOTate

This command sets or queries the annotation state for measurements.

Group Measurement

Syntax
```
MEASUrement:ANNOTate {OFF|AUTO}
MEASUrement:ANNOTate?
```

Arguments `OFF` turns off measurement annotations.

`AUTO` turns on visible measurement annotations.

Examples `MEASUREMENT:ANNOTATE OFF` turns off measurement annotations.

`MEASUREMENT:ANNOTATE?` might return `:MEASUREMENT:ANNOTATION:STATE OFF`, indicating that no measurement annotations are active.

MEASUrement:AUTOset (No Query Form)

This command performs an analysis jitter autoset.

Group Measurement, IMDA

Syntax
```
MEASUrement:AUTOset EXECute |THREEPHASEAUTOset
```

Examples `MEASUREMENT:AUTOSET EXECute` performs an analysis jitter autoset.

`MEASUREMENT:AUTOSET THREEPHASEAUTOset` performs an IMDA 3 phase autoset.

MEASUrement:AUTOset DPMAutoset (No Query Form)

This command performs a power rail autoset operation.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax
```
MEASUrement:AUTOset DPMAutoset
```

Examples `MEASUrement:AUTOset DPMAutoset` performs a power rail autoset.

MEASUrement:AUTOset DPMPReset (No Query Form)

This command performs a power rail preset operation.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax

```
MEASUrement:AUTOset DPMPReset
```

Examples `MEASUrement:AUTOset DPMPReset` performs a power rail preset.

MEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:FALLHigh?
```

Arguments `<NR3>` is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

Examples `MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLHIGH 1.5` sets the high reference level of the falling edge to 1.5 V.

`MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLHIGH?` might return `:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000` indicating that the high reference level of the falling edge is 1.0 V.

MEASUrement:CH<x>:REFLevels:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:ABSolute:FALLLow <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:FALLLow?
```

Arguments <NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

Examples `MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW 1.5` sets the low reference level of the falling edge to 1.5 V.

`MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW?` might return `:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLLOW 1.5000` indicating that the low reference level of the falling edge is 1.5 V.

MEASUrement:CH<x>:REFLevels:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:ABSolute:FALLMid <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:FALLMid?
```

Arguments <NR3> is the mid reference level used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.

Examples `MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLMID 0` sets the mid reference level of the falling edge to 0.0 V.

`MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLMID?` might return `:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0` indicating that the mid reference level of the falling edge is 0.0 V.

MEASUrement:CH<x>:REFLevels:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:ABSolute:HYSTeresis <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:HYSTeresis?
```

Arguments <NR3> is the hysteresis value used for autoset.

Examples `MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3` sets the reference hysteresis level to 30 mV.

`:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS?` might return `:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3` indicating that reference hysteresis level is set to 30 mV.

MEASUrement:CH<x>:REFLevels:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:ABSolute:RISEHigh <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:RISEHigh?
```

Arguments <NR3> is the high reference level of the rising edge when the source ref level method is set to absolute.

Examples `MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.5` sets the high reference level of the rising edge to 1.5 V.

`MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH?` might return `:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000` indicating that the high reference level of the rising edge is 1.0 V.

MEASUrement:CH<x>:REFLevels:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:ABSolute:RISELow <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:RISELow?
```

Arguments <NR3> is the low reference level of the rising edge when the source ref level method is set to absolute.

Examples MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW 1.5 sets the low reference level of the rising edge to 1.5 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW? might return :MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISELOW 1.0000 indicating that the low reference level of the rising edge is 1.0 V.

MEASUrement:CH<x>:REFLevels:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the source ref level method is set to absolute. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:ABSolute:RISEMid <NR3>
MEASUrement:CH<x>:REFLevels:ABSolute:RISEMid?
```

Arguments <NR3> is the mid reference level of the rising edge when the source ref level method is set to absolute.

Examples MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID 0 sets the mid reference level of the rising edge to 0.0 V.

MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID? might return :MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating that the mid reference level of the rising edge is 0.0 V.

MEASUrement:CH<x>:REFLevels:ABSolute:TYPE

This command sets or queries the reference level type for the source. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:ABSolute:TYPE {SAME|UNIQue}
MEASUrement:CH<x>:REFLevels:ABSolute:TYPE?
```

Arguments `SAME` specifies that the absolute reference levels for the specified measurement channel are the same.

`UNIQue` specifies that the absolute reference levels for the specified measurement channel are not the same.

Examples `MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:TYPE SAME` set the reference levels to be the same.

`MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:TYPE?` might return `:MEASUREMENT:CH2:REFLEVELS:ABSOLUTE:TYPE SAME` indicating that the reference levels for absolute measurements are the same.

MEASUrement:CH<x>:REFLevels:BASETop

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:BASETop
{AUTO|MINMax|MEANhistogram|MODEhistogram|EYEhistogram}
MEASUrement:CH<x>:REFLevels:BASETop?
```

Arguments `AUTO` automatically chooses a reference level method.

`MINMax` specifies that reference levels are relative to the measurement MIN and MAX.

`MEANhistogram` specifies that reference levels are relative to the histogram mean BASE and TOP.

`MODEhistogram` specifies that reference levels are relative to the histogram mode BASE and TOP.

`EYEhistogram` specifies that reverence levels are relative to the eye histogram BASE and TOP.

Examples `MEASUREMENT:CH2:REFLEVELS:BASETOP MINMAX` specifies that reference levels are relative to the measurement MIN and MAX.

`MEASUREMENT:CH2:REFLEVELS:BASETOP?` might return `:MEASUREMENT:CH2:REFLEVELS:BASETOP AUTO` indicating the reference levels are chosen automatically.

MEASUrement:CH<x>:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:CH<x>:REFLevels:METHod?
```

Arguments `PERCent` specifies percent reference level units.

`ABSolute` specifies absolute reference level units.

Examples `MEASUREMENT:CH2:REFLEVELS:METHOD PERCENT` sets reference levels to be calculated in percent.

`MEASUREMENT:CH2:REFLEVELS:METHOD?` might return `:MEASUREMENT:CH2:REFLEVELS:METHOD PERCENT` indicating the reference levels are calculated in percent.

MEASUrement:CH<x>:REFLevels:PERCent:FALLHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:PERCent:FALLHigh <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:FALLHigh?
```

Arguments `<NR3>` is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples `MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH 90.0000` sets the high reference level for the falling edge to 90%.

`MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH?` might return `:MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLHIGH 90.0000` indicating the high reference level for the falling edge is set to 90%.

MEASUrement:CH<x>:REFLevels:PERCent:FALLLow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:PERCent:FALLLow <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:FALLLow?
```

Arguments <NR3> is the percentage (where 100% is equal to TOP) used to calculate the low reference level when the measurement Ref level method is set to Percent.

Examples `MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLLOW 10.0000` sets the reference levels for the falling edge to 10%.

`MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLLOW?` might return `:MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLLOW 10.0000` indicating the reference levels for the falling edge is set to 10%.

MEASUrement:CH<x>:REFLevels:PERCent:FALLMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the source ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:PERCent:FALLMid <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:FALLMid?
```

Arguments <NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLMiD 50.0000 sets the MID reference level for the falling edge to 50%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLMID? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:FALLMID 50.0000 indicating the MID reference level for the falling edge is set to 50%.

MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:HYSTeresis?
```

Arguments <NR3> is the hysteresis value used for the autoset.

Examples MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000 sets the reference level hysteresis to 5.0 mV.

MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000 indicating the reference level hysteresis is set to 5.0 mV.

MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:RISEHigh?
```

Arguments

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples

`MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH 90.0000` sets the high reference level for the rising edge to 90%.

`MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH?` might return `:MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEHIGH 90.0000` indicating the high reference level for the rising edge is set to 90%.

MEASUrement:CH<x>:REFLevels:PERCent:RISELow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:CH<x>:REFLevels:PERCent:RISELow <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:RISELow?
```

Arguments

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples

`MEASUREMENT:CH2:REFLEVELS:PERCENT:RISELOW 10.0000` sets the reference levels for the rising edge to 10%.

`MEASUREMENT:CH2:REFLEVELS:PERCENT:RISELOW?` might return `:MEASUREMENT:CH2:REFLEVELS:PERCENT:RISELOW 10.0000` indicating the reference levels for the rising edge is set to 10%.

MEASUrement:CH<x>:REFLevels:PERCent:RISEMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement ref level method is set to percent. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:PERCent:RISEMid <NR3>
MEASUrement:CH<x>:REFLevels:PERCent:RISEMid?
```

Arguments <NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID 50.0000 sets the MID reference level for the rising edge to 50%.

MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:RISEMID 50.0000 indicating the MID reference level for the rising edge is set to 50%.

MEASUrement:CH<x>:REFLevels:PERCent:TYPE

This command specifies or queries the reference level percent type for the measurement. The channel number is specified by x.

Group Measurement

Syntax
```
MEASUrement:CH<x>:REFLevels:PERCent:TYPE
{TENNinety|TWENtyeighty|CUSTom}
MEASUrement:CH<x>:REFLevels:PERCent:TYPE?
```

Arguments TENNinety specifies reference levels at the 10 and 90% levels.

TWENtyeighty specifies reference levels at the 20 and 80% levels.

CUSTom specifies custom reference levels.

Examples MEASUREMENT:CH2:REFLEVELS:PERCENT:TYPE TENNINETY sets the reference levels to the 10 and 90% levels.

MEASUREMENT:CH2:REFLEVELS:PERCENT:TYPE? might return :MEASUREMENT:CH2:REFLEVELS:PERCENT:TYPE TWENTYEIGHTY indicating the reference levels are set to the 20 and 80% levels.

MEASUrement:CLOCKRecovery:ADVanced:METHod

This command sets or queries the global advanced clock recovery method. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax
```
MEASUrement:CLOCKRecovery:ADVanced:METHod
{NONE|NOMinal|PATTern}
MEASUrement:CLOCKRecovery:ADVanced:METHod?
```

Arguments `NONE` sets to use no advanced CRD method.

`NOMinal` sets the advanced CRD method to Nominal Data Rate.

`PATTern` sets the advanced CDR method to use a Known Data Pattern.

Examples `MEASUREMENT:CLOCKRECOVERY:ADVANCED:METHOD NOMINAL` sets the CDR method to use a Known Data Pattern.

`MEASUREMENT:CLOCKRECOVERY:ADVANCED:METHOD?` might return `:MEASUREMENT:CLOCKRECOVERY:ADVANCED:METHOD NONE` indicating that no advanced CRD method will be used.

MEASUrement:CLOCKRecovery:CLOCKFrequency

This command sets or queries the global clock frequency used when fixed constant clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax
```
MEASUrement:CLOCKRecovery:CLOCKFrequency <NR3>
MEASUrement:CLOCKRecovery:CLOCKFrequency?
```

Arguments `<NR3>` is the global clock frequency used with Constant Clock - Fixed clock recovery method.

Examples `MEASUREMENT:CLOCKRECOVERY:CLOCKFREQUENCY 2.0E+9` sets the clock frequency to 2.0 GHz.

:MEASUREMENT:CLOCKRECOVERY:CLOCKFREQUENCY? might return :MEASUREMENT:CLOCKRECOVERY:CLOCKFREQUENCY 2.5000E+9 indicating the clock frequency is set to 2.5 GHz.

MEASUrement:CLOCKRecovery:CLOCKMultiplier

This command sets or queries the global clock multiplier used when explicit clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax

```
MEASUrement:CLOCKRecovery:CLOCKMultiplier <NR3>
MEASUrement:CLOCKRecovery:CLOCKMultiplier?
```

Arguments <NR3> is the global clock multiplier.

Examples MEASUREMENT:CLOCKRECOVERY:CLOCKMULTIPLIER 1.000 sets the clock multiplier to 1.000.

MEASUREMENT:CLOCKRECOVERY:CLOCKMULTIPLIER? might return :MEASUREMENT:CLOCKRECOVERY:CLOCKMULTIPLIER 1.0000 indicating the clock multiplier is set to 1.0000.

MEASUrement:CLOCKRecovery:CONSTCLOCKMODe

This command sets or queries the global constant clock mode used when constant clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax

```
MEASUrement:CLOCKRecovery:CONSTCLOCKMODe {MEAN|MEDian|FIXed}
MEASUrement:CLOCKRecovery:CONSTCLOCKMODe?
```

Arguments MEAN sets the constant clock mode to MEAN.

MEDian sets the constant clock mode to MEDian.

FIXed sets the constant clock mode to FIXed.

Examples `MEASUREMENT:CLOCKRECOVERY:CONSTCLOCKMODE MEAN` sets the constant clock mode to MEAN.

`MEASUREMENT:CLOCKRECOVERY:CONSTCLOCKMODE?` might return `:MEASUREMENT:CLOCKRECOVERY:CONSTCLOCKMODE MEAN` indicating the constant clock mode is set to MEAN.

MEASUrement:CLOCKRecovery:DAMPing

This command sets or queries the global damping value used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax
```
MEASUrement:CLOCKRecovery:DAMPing <NR3>
MEASUrement:CLOCKRecovery:DAMPing?
```

Arguments `<NR3>` is the global clock recovery damping value.

Examples `MEASUREMENT:CLOCKRECOVERY:DAMPING`

`:MEASUREMENT:CLOCKRECOVERY:DAMPING?` might return `:MEASUREMENT:CLOCKRECOVERY:DAMPING 700.0000E-3` indicating the damping value is set to

MEASUrement:CLOCKRecovery:DATAPath

This command sets or queries the global file containing the data pattern used when known data pattern clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax
```
MEASUrement:CLOCKRecovery:DATAPath <QString>
MEASUrement:CLOCKRecovery:DATAPath?
```

Arguments `<QString>` is the path and filename, in quotes, of the file containing the data pattern.

Examples MEASUREMENT:CLOCKRECOVERY:DATAPATH "C:/E:" sets the data path to C:/E:.

MEASUREMENT:CLOCKRECOVERY:DATAPATH? might return :MEASUREMENT:CLOCKRECOVERY:DATAPATH "C:/E".

MEASUrement:CLOCKRecovery:DATARate

This command sets or queries the global nominal data bit rate used when nominal data rate clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax
```
MEASUrement:CLOCKRecovery:DATARate <NR3>
MEASUrement:CLOCKRecovery:DATARate?
```

Arguments <NR3> is the global value for the Nominal data rate.

Examples MEASUREMENT:CLOCKRECOVERY:DATARATE 2.0e+9 sets the data rate for clock recovery to 2.0 GHz.

MEASUREMENT:CLOCKRECOVERY:DATARATE? might return :MEASUREMENT:CLOCKRECOVERY:DATARATE 2.5000E+9 indicating the data rate is 2.5 GHz.

MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODe

This command sets or queries the global explicit clock mode used when explicit clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax
```
MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODe {EDGE|PLL}
MEASUrement:CLOCKRecovery:EXPLICITCLOCKMODe?
```

Arguments EDGE sets the clock mode to clock edge.

PLL sets the clock mode to phase locked loop.

Examples `MEASUREMENT:CLOCKRECOVERY:EXPLICITCLOCKMODE EDGE` sets the clock mode to edge.

`MEASUREMENT:CLOCKRECOVERY:EXPLICITCLOCKMODE?` might return `:MEASUREMENT:CLOCKRECOVERY:EXPLICITCLOCKMODE EDGE` indicating the clock mode is edge.

MEASUrement:CLOCKRecovery:JTFBandwidth

This command sets or queries the global JTF bandwidth used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax
```
MEASUrement:CLOCKRecovery:JTFBandwidth <NR3>
MEASUrement:CLOCKRecovery:JTFBandwidth?
```

Arguments `<NR3>` is the global clock recovery JTF bandwidth.

Examples `MEASUREMENT:CLOCKRECOVERY:JTFBANDWIDTH 2.0e6` sets the bandwidth to 2.0 MHz.

`MEASUREMENT:CLOCKRECOVERY:JTFBANDWIDTH?` might return `:MEASUREMENT:CLOCKRECOVERY:JTFBANDWIDTH 1.0000E+6` indicating the bandwidth is 1.0 MHz.

MEASUrement:CLOCKRecovery:LOOPBandwidth

This command sets or queries the global loop bandwidth used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax
```
MEASUrement:CLOCKRecovery:LOOPBandwidth <NR3>
MEASUrement:CLOCKRecovery:LOOPBandwidth?
```

Arguments `<NR3>` is the global loop bandwidth.

Examples MEASUREMENT:CLOCKRECOVERY:LOOPBANDWIDTH 2.0e6 sets the loop bandwidth to 2.0 MHz.

MEASUREMENT:CLOCKRECOVERY:LOOPBANDWIDTH? might return :MEASUREMENT:CLOCKRECOVERY:LOOPBANDWIDTH 1.0000E+6 indicating the loop bandwidth is 1.0 MHz.

MEASUrement:CLOCKRecovery:MEANAUTOCalculate

This command sets or queries how often the clock is calculated when constant clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax
```
MEASUrement:CLOCKRecovery:MEANAUTOCalculate {FIRST|EVERY}
MEASUrement:CLOCKRecovery:MEANAUTOCalculate?
```

Arguments FIRST calculates the clock on the first acquisition.

EVERY calculates the clock on every acquisition.

Examples MEASUREMENT:CLOCKRECOVERY:MEANAUTOCALCULATE EVERY calculates the clock on every acquisition.

MEASUREMENT:CLOCKRECOVERY:MEANAUTOCALCULATE? might return :MEASUREMENT:CLOCKRECOVERY:MEANAUTOCALCULATE FIRST indicating the clock is calculated on the first acquisition.

MEASUrement:CLOCKRecovery:METHod

This command sets or queries the global clock recovery method for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group Measurement

Syntax
```
MEASUrement:CLOCKRecovery:METHod
{PLL|CONSTANTCLOCK|EXPLICITCLOCK}
MEASUrement:CLOCKRecovery:METHod?
```

Arguments

PLL specifies using the phase locked loop.

CONSTANTCLOCK specifies using a constant clock.

EXPLICITCLOCK specifies using an explicit clock.

Examples

MEASUREMENT:CLOCKRECOVERY:METHOD EXPLICITCLOCK specifies using an explicit clock.

MEASUREMENT:CLOCKRECOVERY:METHOD? might return :MEASUREMENT:CLOCKRECOVERY:METHOD PLL indicating the method is set to PLL.

MEASUrement:CLOCKRecovery:MODel

This command sets or queries the global phase locked loop (PLL) clock recovery model used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:MODel {TYPE1|TYPE2}
MEASUrement:CLOCKRecovery:MODel?
```

Arguments

TYPE1 PLL clock recovery uses PLL model type I.

TYPE2 PLL clock recovery uses PLL model type II.

Examples

MEASUREMENT:CLOCKRECOVERY:MODEL TYPE2 sets the model to type 2.

MEASUREMENT:CLOCKRECOVERY:MODEL? might return :MEASUREMENT:CLOCKRECOVERY:MODEL TYPE1 indicating the recovery model is type 1.

MEASUrement:CLOCKRecovery:NOMINALOFFset

This command sets or queries the global offset value used when explicit clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:NOMINALOFFset <NR3>
MEASUrement:CLOCKRecovery:NOMINALOFFset?
```

Arguments

<NR3> is the global clock offset.

Examples

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET 1.0 sets the offset to 1.0.

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET? might return :MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET 0.0E+0 indicating the offset is set to 0.0.

MEASUrement:CLOCKRecovery:NOMINALOFFset:SELECTIONtype

This command sets or queries the global offset type used when explicit clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:NOMINALOFFset:SELECTIONtype
{AUTO|MANUAL}
MEASUrement:CLOCKRecovery:NOMINALOFFset:SELECTIONtype?
```

Arguments

AUTO sets the selection type to AUTO.

MANUAL sets the selection type to MANUAL.

Examples

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET:SELECTIONTYPE AUTO sets the selection type to auto.

MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET:SELECTIONTYPE? might return :MEASUREMENT:CLOCKRECOVERY:NOMINALOFFSET:SELECTIONTYPE MANUAL indicating the selection type is manual.

MEASUrement:CLOCKRecovery:STAndard

This command sets or queries the global communications standard used when PLL clock recovery is used for the measurement. This will affect measurements whose :MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal flag is set to 1.

Group

Measurement

Syntax

```
MEASUrement:CLOCKRecovery:STAndard
{CUSTom|ENET100|FW1394BS400B|FW1394BS800B|
FW1394BS1600B|FBD1|FBD2|FBD3|FC133|FC266|
FC531|FC1063|FC2125|FC4250|FC8500|ENET1000|
IBA2500|IBA_GEN2|OC1|OC3|OC12|OC48|PCIE_GEN1|
PCIE_GEN2|PCIE_GEN3|RIO125|RIO250|RIO3125|
SAS15_NOSSC|SAS3_NOSSC|SAS6_NOSSC|SAS12_NOSSC|
SAS15_SSC|SAS3_SSC|SAS6_SSC|SAS12_SSC|SATA_GEN1|
SATA_GEN2|SATA_GEN3|USB3|XAUI|XAUI_GEN2}
MEASUrement:CLOCKRecovery:STAndard?
```

Arguments Arguments are the global clock recovery standards.

Examples `MEASUREMENT:CLOCKRECOVERY:STANDARD PCIE_GEN3` sets the standard to PCIE_GEN3.

`MEASUREMENT:CLOCKRECOVERY:STANDARD?` might return `:MEASUREMENT:CLOCKRECOVERY:STANDARD PCIE_GEN1` indicating the standard is set to PCIE_GEN1.

MEASUrement:DELete (No Query Form)

The command deletes the specified measurement.

Group Measurement

Syntax

```
MEASUrement:DELete <QString>
```

Arguments `<QString>` is the measurement to delete. Argument is of the form "`MEAS<NR1>`" where `<NR1>` is ≥1.

Examples `MEASUREMENT:DELETE "MEAS4"` deletes measurement 4.

MEASUrement:DELETEALL (No Query Form)

This command deletes all the active instances of measurements defined in the scope application.

Group Measurement

Syntax `MEASUrement:DELETEALL`

Examples `MEASUREMENT:DELETEALL` deletes all the active instances of measurements.

MEASUrement:DIRacmodel

This command sets or queries the dirac model used to separate random from deterministic jitter for jitter measurements.

Group Measurement

Syntax
```
MEASUrement:DIRacmodel {PCIExpress|FIBREchannel}
MEASUrement:DIRacmodel?
```

Arguments `PCIExpress` specifies the PCIExpress dirac model.

`FIBREchannel` specifies the FIBREchannel dirac model.

Examples `MEASUREMENT:DIRACMODEL FIBRECHANNEL` sets the dirac model to FIBRECHANNEL.

`MEASUREMENT:DIRACMODEL?` might return `:MEASUREMENT:DIRACMODEL PCIEXPRESS` indicating the dirac model is set to PCIECPRESS.

MEASUrement:DISPLAYUnits

This command sets or queries the display units used for jitter summary measurements.

Group Measurement

Syntax `MEASUrement:DISPLAYUnits {SEConds|UNITINtervals}`

Arguments `SEConds` set the display units to seconds.

`UNITINtervals` set the display units to unit intervals.

Examples `MEASUREMENT:DISPLAYUNITS UNITINTERVALS` sets the display units to unit intervals.

MEASUREMENT:DISPLAYUNITS? might return :MEASUREMENT:DISPLAYUNITS SECONDS indicating the display units are seconds.

MEASUrement:EDGE<x>

This command sets or queries the type of the edge for the measurement.

Group Measurement

Syntax

```
MEASUrement:EDGE<x> {RISE|FALL|BOTH}
```

Arguments

FALL specifies the falling edge of the waveform.

RISE specifies the rising edge of the waveform.

BOTH specifies both a rising and falling edge of the waveform.

Examples

MEASUrement:EDGE2

MEASUrement:EDGE2? might return :MEASUREMENT:EDGE2 BOTH indicating

MEASUrement:EYERENDER

This command sets or queries the state of high-performance eye rendering for an eye diagram.

Group Measurement

Syntax

```
MEASUrement:EYERENDER {<NR1>|OFF|ON}
MEASUrement:EYERENDER?
```

Arguments

<NR1> = 0 disables high-performance eye rendering for an eye diagram; any other value turns this feature on.

OFF disables high-performance eye rendering for an eye diagram.

ON enables high-performance eye rendering for an eye diagram.

Examples `MEASUREMENT:EYERENDER`

`MEASUREMENT:EYERENDER?` might return `:MEASUREMENT:EYERENDER 1` indicating high-performance eye rendering is on.

MEASUrement:FILTers:BLANKingtime

This command sets or queries the global filter blanking time.

Group Measurement

Syntax
```
MEASUrement:FILTers:BLANKingtime <NR3>
MEASUrement:FILTers:BLANKingtime?
```

Arguments `<NR3>` is the current filter blanking time.

Examples `MEASUREMENT:FILTERS:BLANKINGTIME` 3.0 sets the blanking time to 3.0.

`MEASUREMENT:FILTERS:BLANKINGTIME?` might return `:MEASUREMENT:FILTERS:BLANKINGTIME 4.0000` indicating the blanking time is 4.0.

MEASUrement:FILTers:HIGHPass:FREQ

This command sets or queries the global high pass filter frequency for the measurement.

Group Measurement

Syntax
```
MEASUrement:FILTers:HIGHPass:FREQ <NR3>
MEASUrement:FILTers:HIGHPass:FREQ?
```

Arguments `<NR3>` is the current high pass filter frequency.

Examples `MEASUREMENT:FILTERS:HIGHPASS:FREQ 20.0e6` sets the frequency to 20 MHz.

`MEASUREMENT:FILTERS:HIGHPASS:FREQ?` might return `:MEASUREMENT:FILTERS:HIGHPASS:FREQ 10.0000E+6` indicating the filter frequency is set to 10.0 MHz.

MEASUrement:FILTers:HIGHPass:SPEC

This command sets or queries the global high pass filter order for the measurement.

Group Measurement

Syntax

```
MEASUrement:FILTers:HIGHPass:SPEC {NONE|FIRST|SECOND|THIRD}
MEASUrement:FILTers:HIGHPass:SPEC?
```

Arguments

`NONE` specifies no high pass filter.

`FIRST` specifies a first order high pass filter.

`SECOND` specifies a second order high pass filter.

`THIRD` specifies a third order high pass filter.

Examples

`MEASUREMENT:FILTERS:HIGHPASS:SPEC NONE` specifies no high pass filter.

`MEASUREMENT:FILTERS:HIGHPASS:SPEC?` might return `:MEASUREMENT:FILTERS:HIGHPASS:SPEC SECOND` indicating a second order high pass filter.

MEASUrement:FILTers:LOWPass:FREQ

This command sets or queries the global low pass filter cutoff frequency for the measurement.

Group Measurement

Syntax

```
MEASUrement:FILTers:LOWPass:FREQ <NR3>
MEASUrement:FILTers:LOWPass:FREQ?
```

Arguments

`<NR3>` is the current low pass filter frequency.

Examples

`MEASUREMENT:FILTERS:LOWPASS:FREQ 20.0e6` sets the low pass frequency to 20 MHz.

`MEASUREMENT:FILTERS:LOWPASS:FREQ?` might return `:MEASUREMENT:FILTERS:LOWPASS:FREQ 10.0000E+6` indicating the low pass frequency is 10.0 MHz.

MEASUrement:FILTers:LOWPass:SPEC

This command sets or queries the global low pass filter order for the measurement.

Group Measurement

Syntax

```
MEASUrement:FILTers:LOWPass:SPEC {NONE|FIRST|SECOND|THIRD}
MEASUrement:FILTers:LOWPass:SPEC?
```

Arguments

`NONE` specifies no low pass filter.

`FIRST` specifies a first order low pass filter.

`SECOND` specifies a second order low pass filter.

`THIRD` specifies a third order low pass filter.

Examples

`MEASUREMENT:FILTERS:LOWPASS:SPEC SECOND` specifies a second order low pass filter.

`MEASUREMENT:FILTERS:LOWPASS:SPEC?` might return `:MEASUREMENT:FILTERS:LOWPASS:SPEC NONE` indicating no low pass filter.

MEASUrement:FILTers:RAMPtime

This command sets or queries the global filter ramp time for the measurement.

Group Measurement

Syntax

```
MEASUrement:FILTers:RAMPtime <NR3>
MEASUrement:FILTers:RAMPtime?
```

Arguments

`<NR3>` is the current filter ramp time.

Examples

`MEASUREMENT:FILTERS:RAMPTIME 3.0` sets the ramp time to 3.0.

`MEASUREMENT:FILTERS:RAMPTIME?` might return `:MEASUREMENT:FILTERS:RAMPTIME 2.0000` indicating the ramp time is 2.0.

MEASUrement:GATing

This command sets or queries the global gating type for the measurement.

Group Measurement

Syntax

```
MEASUrement:GATing {NONE|SCREEN|CURSor|LOGic|SEARch|TIMe}
MEASUrement:GATing?
```

Related Commands MEASUrement:GATing:ENDtime

MEASUrement:GATing:STARTtime

Arguments `NONE` turns off measurement gating.

`SCREen` turns on gating, using the left and right edges of the screen.

`CURSor` limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.

`LOGic` specifies that measurements are taken only on the portion of the waveform where the logic source is in the active state.

`SEARCH` specifies that measurements are taken based on search criteria.

`TIMe` limits measurements to the portion of the waveform between the Start and End gate times

Examples `MEASUREMENT:GATING SCREEN` turns on measurement gating, using the left and right edges of the screen.

`MEASUREMENT:GATING?` might return `:MEASUREMENT:GATING CURSOR`, indicating that measurements are limited to the portion of the waveform between the vertical bar cursors.

MEASUrement:GATing:ACTive

This command sets or queries the global gating active level used for logic gating.

Group Measurement

Syntax

```
MEASUrement:GATing:ACTive {HIGH|LOW}
MEASUrement:GATing:ACTive?
```

Arguments `HIGH` specifies the gate is HIGH.

`LOW` specifies the gate is LOW.

Examples `MEASUREMENT:GATING:ACTIVE LOW` specifies a low gate.

`MEASUREMENT:GATING:ACTIVE?` might return `:MEASUREMENT:GATING:ACTIVE HIGH` indicating the gate is high.

MEASUrement:GATing:ENDtime

Sets or queries the end gate time for all measurements that use Global gating.

Group Measurement

Syntax
```
MEASUrement:GATing:ENDtime <NR3>
MEASUrement:GATing:ENDtime?
```

Related Commands MEASUrement:GATing

Arguments `<NR3>` is the time gating end gate time in seconds. The valid range is -10000 s to 10000 s.

Examples `MEASUrement:GATing:ENDtime 5.31e-2` sets the global end gate time to 53.1 ms.

`MEASUrement:GATing:ENDtime?` might return `1.05` indicating that the global end gate time is 1.05 s

MEASUrement:GATing:HYSTeresis

This command sets or queries the global gating hysteresis value used for logic gating.

Group Measurement

Syntax
```
MEASUrement:GATing:HYSTeresis <NR3>
MEASUrement:GATing:HYSTeresis?
```

Arguments `<NR3>` is the gating hysteresis.

Examples `MEASUREMENT:GATING:HYSTERESIS 40.0e-3` sets the hysteresis to 40 mV.

`MEASUREMENT:GATING:HYSTERESIS?` might return `:MEASUREMENT:GATING:HYSTERESIS 30.0000E-3` indicating the hysteresis is 30 mV.

MEASUrement:GATing:LOGICSource

This command sets or queries the gating data source used for logic gating.

Group Measurement

Syntax
```
MEASUrement:GATing:LOGICSource {CH<x>|MATH<x>|REF<x>}
MEASUrement:GATing:LOGICSource?
```

Arguments Arguments are the logic gating source.

Examples `MEASUREMENT:GATING:LOGICSOURCE MATH1` sets the logic source to MATH 1.

`MEASUREMENT:GATING:LOGICSOURCE?` might return `:MEASUREMENT:GATING:LOGICSOURCE CH6` indicating the logic source is channel 6.

MEASUrement:GATing:MIDRef

This command sets or queries the global gating mid ref value used for logic gating.

Group Measurement

Syntax
```
MEASUrement:GATing:MIDRef <NR3>
MEASUrement:GATing:MIDRef?
```

Arguments `<NR3>` is the mid ref value for gating.

Examples `MEASUREMENT:GATING:MIDREF 2.0` set the gating mid ref to 2.0 V.

`MEASUREMENT:GATING:MIDREF?` might return `:MEASUREMENT:GATING:MIDREF 1.5000` indicating the gating mid ref is 1.5 V.

MEASUrement:GATing:SEARCHSource

This command sets or queries the global gating search source when the gating type is search.

Group Measurement

Syntax
```
MEASUrement:GATing:SEARCHSource SEARCH1
MEASUrement:GATing:SEARCHSource?
```

Arguments Argument is the search source.

Examples `MEASUREMENT:GATING:SEARCHSOURCE SEARCH1` set the search source to search 1.

`MEASUREMENT:GATING:SEARCHSOURCE?` might return `:MEASUREMENT:GATING:SEARCHSOURCE UNDEFINED` indicating the search source is not defined.

MEASUrement:GATing:STARTtime

Sets or queries the start gate time for all measurements that use Global gating.

Group Measurement

Syntax
```
MEASUrement:GATing:STARTtime <NR3>
MEASUrement:GATing:STARTtime?
```

Related Commands MEASUrement:GATing

Arguments `<NR3>` is the time gating start gate time in seconds. The valid range is -10000 s to 10000 s.

Examples `MEASUrement:GATing:STARTtime -1.5e-3` sets the global start gate time to -1.5 ms.

`MEASUrement:GATing:STARTtime?` might return `3.0e-9`, indicating that the global start gate time is 3 ns.

MEASUrement:HIGHLEVel:CONNECTion

This command sets or queries the connection type for the measurement.

Group Measurement, IMDA

Syntax

```
MEASUrement:HIGHLEVel:CONNECTion {L2L|L2N}
MEASUrement:HIGHLEVel:CONNECTion?
```

Arguments

`L2L` specifies L2L connection type.

`L2N` specifies L2N connection type.

Examples

`MEASUREMENT:HIGHLEVEL:CONNECTION L2L` sets the connection type as L2L.

`MEASUREMENT:HIGHLEVEL:CONNECTION?` might return `:MEASUrement:HIGHLEVel:CONNECTion L2N`, indicating the connection type is set to L2N.

MEASUrement:HIGHLEVel:L2LTOL2N

This command enables the Line-to-Line to Line-to-Neutral Conversion.

Group Measurement, IMDA

Syntax

```
MEASUrement:HIGHLEVel:L2LTOL2N {0|1}
MEASUrement:HIGHLEVel:L2LTOL2N?
```

Examples

`MEASUREMENT:HIGHLEVEL:L2LTOL2N 1` enables Line-to-Line to Line-to-Neutral Conversion.

`MEASUREMENT:HIGHLEVEL:L2LTOL2N?` might return `:MEASUrement:HIGHLEVel:L2LTOL2N 0`, indicating Line-to-Line to Line-to-Neutral Conversion is disabled.

MEASUrement:HIGHLEVel:WIRing

This command sets or queries the wiring configuration for the measurement.

Group Measurement, IMDA

Syntax

```
MEASUrement:HIGHLEVel:WIRing {P3W3V2I2|P3W3V3I3|P3W4}
MEASUrement:HIGHLEVel:WIRing?
```

Arguments

`P3W3V2I2` specifies 3 phase- 3 wire (with 2 voltage and 2 current sources) wiring configuration.

`P3W3V3I3` specifies 3 phase- 3 wire (with 3 voltage and 3 current sources) wiring configuration.

`P3W4` specifies 3 Phase and 4 Wire, the 4th wire being Neutral.

Examples

`MEASUREMENT:HIGHLEVEL:WIRING P3W4` sets the wiring configuration to P3W4.

`MEASUREMENT:HIGHLEVEL:WIRING?` might return `:MEASUrement:HIGHLEVel:WIRing P3W3V3I3`, indicating the wiring configuration is P3W3V3I3.

MEASUrement:INTERp

This command sets or queries the interpolation mode used to locate edge crossings.

Group Measurement

Syntax

```
MEASUrement:INTERp {AUTO|SINX|LINear}
MEASUrement:INTERp?
```

Arguments

`AUTO` automatically selects the interpolation mode.

`SINX` specifies sin(x)/x interpolation, where acquired points are fit to a curve.

`LINear` specifies linear interpolation, where acquired points are connected with straight lines.

Examples

`MEASUREMENT:INTERP SINX` set the interpolation mode to sin(x)/x.

`MEASUREMENT:INTERP?` might return `:MEASUREMENT:INTERP AUTO` indicating the interpolation mode is auto.

MEASUrement:JITTermodel

This command sets or queries the model used to separate random from deterministic jitter for jitter measurements.

Group Measurement

Syntax
```
MEASUrement:JITTermodel {SPECTRAL|SPECTRALBUJ}
MEASUrement:JITTermodel?
```

Arguments `SPECTRAL` sets the jitter separation model to spectral only.

`SPECTRALBUJ` sets the jitter separation model to spectral plus BUJ.

Examples `MEASUREMENT:JITTERMODEL SPECTRAL` set the jitter model to spectral.

`MEASUREMENT:JITTERMODEL?` might return `:MEASUREMENT:JITTERMODEL SPECTRAL` indicating the jitter model is set to spectral.

MEASUrement:LIST? (Query Only)

This query returns a comma separated list of all currently defined measurements.

Group Measurement

Syntax
```
MEASUrement:LIST?
```

Returns Returns a list of all currently defined measurements.

Examples `MEASUREMENT:LIST?` might return `:MEASUREMENT:LIST MEAS1,MEAS2,MEAS3,MEAS4,MEAS5,MEAS6` indicating 6 measurements are defined.

MEASUrement:LOCKRJ

This command sets or queries the state of RJ locking.

Group Measurement

Syntax
```
MEASUrement:LOCKRJ {OFF|ON|<NR1>}
MEASUrement:LOCKRJ?
```

Arguments

MEAS<x> specifies the measurement number.

ON indicates that RJ locking is active.

OFF indicates that RJ locking is off.

<NR1> a 0 turns off RJ locking; any other value activates RJ locking.

Examples

MEASUREMENT:LOCKRJ ON activates RJ locking

MEASUrement:LOCKRJValue

This command sets or queries the RJ lock value.

Group

Measurement

Syntax

```
MEASUrement:LOCKRJValue <NR3>
MEASUrement:LOCKRJValue?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> default value = 1e-12; minimum value = 1e-15; maximum value = 1.

Examples

MEASUREMENT:LOCKRJVALUE 1 sets the lock value to the maximum value of 1.

MEASUREMENT:LOCKRJVALUE? might return :MEASUREMENT:LOCKRJVALUE 1.0000E-12, indicating the lock value is set to the default value of 1e-12.

MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLHigh?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

Examples MEASUrement:MATH1:REFLevels:ABSolute:FALLHigh 1.5 sets the reference level to 1.5 V.

MEASUrement:MATH2:REFLevels:ABSolute:FALLHigh? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating the reference level is 1.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the measuement ref level method is set to absolute. The math number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLLow?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

Examples MEASUrement:MATH2:REFLevels:ABSolute:FALLLow -1.5 sets the reference level to -1.5 V.

MEASUrement:MATH2:REFLevels:ABSolute:FALLLow? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLLOW -1.0000 indicating the reference level is -1.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:FALLMid?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.

Examples

`MEASUrement:MATH2:REFLevels:ABSolute:FALLMid 0.0` sets the reference level to 0.0 V.

`MEASUrement:MATH2:REFLevels:ABSolute:FALLMid?` might return `:MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0` indicating the reference level is 0.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the measurement ref level method is set to absolute. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:HYSTeresis?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the hysteresis value used for the autoset.

Examples

`MEASUrement:MATH2:REFLevels:ABSolute:HYSTeresis 20.0E-3` sets the hysteresis to 20.0 mV.

`MEASUrement:MATH2:REFLevels:ABSolute:HYSTeresis?` might return `:MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3` indicating the hysteresis value is 30.0 mV.

MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEHigh?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

Examples MEASUrement:MATH2:REFLevels:ABSolute:RISEHigh 1.5 sets the reference level to 1.5 V.

MEASUrement:MATH2:REFLevels:ABSolute:RISEHigh? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000 indicating the reference level is 1.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:RISELow?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the high reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHod is set to Absolute.

Examples MEASUrement:MATH2:REFLevels:ABSolute:RISELow -1.5 sets the reference level to -1.5 V.

MEASUrement:MATH2:REFLevels:ABSolute:RISELow? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISELOW -1.0000 indicating the reference level is -1.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the measurement ref level method is set to absolute. The math number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid <NR3>
MEASUrement:MATH<x>:REFLevels:ABSolute:RISEMid?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Absolute.

Examples MEASUrement:MATH2:REFLevels:ABSolute:RISEMid 30.0E-3 sets the reference level to 30 mV.

MEASUrement:MATH2:REFLevels:ABSolute:RISEMid? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating the reference level is 0.0 V.

MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE

This command sets or queries the reference level type for the measurement. The math number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE {SAME|UNIQue}
MEASUrement:MATH<x>:REFLevels:ABSolute:TYPE?
```

Arguments MEAS<x> specifies the measurement number.

SAME specifies that the absolute levels are set the same.

UNIQue specifies that the absolute levels can be set independently.

Examples MEASUrement:MATH2:REFLevels:ABSolute:TYPE UNIQUE specifies that the absolute levels can be set independently.

MEASUrement:MATH2:REFLevels:ABSolute:TYPE? might return :MEASUREMENT:MATH2:REFLEVELS:ABSOLUTE:TYPE SAME indicating that the absolute levels are set the same.

MEASUrement:MATH<x>:REFLevels:BASETop

This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement. The math number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:BASETop
{AUTO|MINMax|MEANhistogram| MODEhistogram|EYEhistogram}
MEASUrement:MATH<x>:REFLevels:BASETop?
```

Arguments

MEAS<x> specifies the measurement number.

AUTO automatically chooses a reference level method.

MINMax specifies that reference levels are relative to the measurement MIN and MAX.

MEANhistogram specifies that reference levels are relative to the histogram mean BASE and TOP.

MODEhistogram specifies that reference levels are relative to the histogram mode BASE and TOP.

EYEhistogram specifies that reverence levels are relative to the eye histogram BASE and TOP.

Examples

MEASUrement:MATH1:REFLevels:BASETop MINMax specifies that reference levels are relative to the measurement MIN and MAX.

MEASUrement:MATH1:REFLevels:BASETop? might return :MEASUREMENT:MATH2:REFLEVELS:BASETOP AUTO indicating the instrument automatically chooses a reference level method.

MEASUrement:MATH<x>:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement. The math number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MATH<x>:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:MATH<x>:REFLevels:METHod?
```

Arguments `MEAS<x>` specifies the measurement number.

`PERCent` specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the `MEASUrement:MATH<x>:REFLevel:PERCent` commands.

`ABSolute` specifies that the reference levels are set explicitly using the `MEASUrement:MATH<x>:REFLevel:ABSolute` commands. This method is useful when precise values are required.

Examples `MEASUrement:MATH2:REFLevels:METHod ABSOLUTE` specifies that the reference levels are set explicitly

`MEASUrement:MATH2:REFLevels:METHod?` might return `:MEASUREMENT:MATH2:REFLEVELS:METHOD PERCENT` indicating reference levels are in percent relative to HIGH and LOW.

MEASUrement:MATH<x>:REFLevels:PERCent:FALLHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MATH<x>:REFLevels:PERCent:FALLHigh <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:FALLHigh?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the percentage (where 100% is equal to HIGH) used to calculate the high reference level when the measurement Ref level method is set to Percent.

Examples `MEASUrement:MATH2:REFLevels:PERCent:FALLHigh 95` sets the reference level to 95% of TOP.

MEASUrement:MATH2:REFLevels:PERCent:FALLHigh? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLHIGH 90.0000 indicating the reference level is set to 90% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:FALLLow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:FALLLow <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:FALLLow?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to HIGH) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples MEASUrement:MATH2:REFLevels:PERCent:FALLLow 5 sets the reference level to 5% of TOP.

MEASUrement:MATH2:REFLevels:PERCent:FALLLow? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLLOW 10.0000 indicating the reference level is set to 10% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement ref level method is set to percent. The math number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:FALLMid?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples MEASUrement:MATH2:REFLevels:PERCent:FALLMid 50 sets the reference level to 50% of TOP.

MEASUrement:MATH2:REFLevels:PERCent:FALLMid? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:FALLMID 50.0000 indicating the reference level is set to 50% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement ref level method is set to percent. The math number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:HYSTeresis?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the hysteresis value used for the autoset.

Examples MEASUrement:MATH2:REFLevels:PERCent:HYSTeresis 2.5 sets the hysteresis to 2.5% of MAX

MEASUrement:MATH2:REFLevels:PERCent:HYSTeresis? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:HYSTERESIS 5.0000 indicating the hysteresis is set to 5% of MAX.

MEASUrement:MATH<x>:REFLevels:PERCent:RISEHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:RISEHigh <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:RISEHigh?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples

`MEASUrement:MATH2:REFLevels:PERCent:RISEHigh 95` sets the reference level to 95% of TOP.

`MEASUrement:MATH2:REFLevels:PERCent:RISEHigh?` might return `:MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEHIGH 90.0000` indicating the reference level is set to 90% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:RISELow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement ref level method is set to percent. The math number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MATH<x>:REFLevels:PERCent:RISELow <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:RISELow?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples

`MEASUrement:MATH2REFLevels:PERCent:RISELow 5` sets the reference level to 5% of TOP.

`MEASUrement:MATH2REFLevels:PERCent:RISELow?` might return `:MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISELOW 10.0000` indicating the reference level is set to 10% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge

when the measurement ref level method is set to percent. The math number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid <NR3>
MEASUrement:MATH<x>:REFLevels:PERCent:RISEMid?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples MEASUrement:MATH2:REFLevels:PERCent:RISEMid 50 sets the reference level to 50% of TOP.

MEASUrement:MATH2:REFLevels:PERCent:RISEMid? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:RISEMID 50.0000 indicating the reference level is set to 50% of TOP.

MEASUrement:MATH<x>:REFLevels:PERCent:TYPE

This command specifies or queries the reference level percent type for the measurement. The math number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MATH<x>:REFLevels:PERCent:TYPE
{TENNinety|TWENtyeighty|CUSTom}
MEASUrement:MATH<x>:REFLevels:PERCent:TYPE?
```

Arguments MEAS<x> specifies the measurement number.

TENNinety sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

TWENtyeighty sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

CUSTom allows setting other reference level percents.

Examples MEASUrement:MATH2:REFLevels:PERCent:TYPE TWENTYEIGHTY sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

MEASUrement:MATH2:REFLevels:PERCent:TYPE? might return :MEASUREMENT:MATH2:REFLEVELS:PERCENT:TYPE TENNINETY indicating the reference levels are set to 10%, 50% and 90%.

MEASUrement:MEASRange:MAX

This command sets or queries the global range maximum value.

Group Measurement

Syntax
```
MEASUrement:MEASRange:MAX <NR3>
MEASUrement:MEASRange:MAX?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the maximum measurement range limit value.

Examples MEASUREMENT:MEASRANGE:MAX 1.5 sets the maximum range limit to 1.5 V.

MEASUREMENT:MEASRANGE:MAX? might return :MEASUREMENT:MEASRANGE:MAX 1.0000 indicating the maximum range limit is 1.0 V.

MEASUrement:MEASRange:MIN

This command sets or queries the global range minimum value.

Group Measurement

Syntax
```
MEASUrement:MEASRange:MIN <NR3>
MEASUrement:MEASRange:MIN?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the minimum measurement range limit value.

Examples MEASUREMENT:MEASRANGE:MIN 30.0E-3 sets the minimum measurement range limit to 30.0 mV.

MEASUREMENT:MEASRANGE:MIN? might return :MEASUREMENT:MEASRANGE:MIN 0.0E+0 indicating minimum range limit is 0.0 V.

MEASUrement:MEASRange:STATE

This command sets or queries the global range state.

Group Measurement

Syntax

```
MEASUrement:MEASRange:STATE {OFF|ON|0|1}
MEASUrement:MEASRange:STATE?
```

Arguments

MEAS<x> specifies the measurement number.

OFF specifies that the measurement results are not limited.

ON specifies that the measurement results are limited to results with values between the range minimum and maximum.

0 specifies that the measurement results are not limited.

1 specifies that the measurement results are limited to results with values between the range minimum and maximum.

Examples

MEASUREMENT:MEASRANGE:STATE 1 specifies that the measurement results are limited to results with values between the range minimum and maximum.

MEASUREMENT:MEASRANGE:STATE? might return :MEASUREMENT:MEASRANGE:STATE 0 indicating that the measurement results are not limited.

MEASUrement:MEAS<x>:BER

This command sets or queries BER value for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:BER <NR3>
MEASUrement:MEAS<x>:BER?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is BER value for the measurement.

Examples `MEASUrement:MEAS2:BER 10.0` sets the BER for the measurement to 10.0.

`MEASUrement:MEAS2:BER?` might return `:MEASUREMENT:MEAS2:BER 12.0000` indicating the measurement BER is 12.0.

MEASUrement:MEAS<x>:BER:TARGETBER

This command sets or queries the target BER value for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:BER:TARGETBER <NR3>
MEASUrement:MEAS<x>:BER:TARGETBER?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the target BER value.

Examples `MEASUrement:MEAS1:BER:TARGETBER 14.0` sets the target BER to 14.0.

`MEASUrement:MEAS1:BER:TARGETBER?` might return `:MEASUREMENT:MEAS1:BER:TARGETBER 12.0000` indicating the target BER is 12.0.

MEASUrement:MEAS<x>:BIN

This command sets or queries the bin count for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:BIN <NR3>
MEASUrement:MEAS<x>:BIN?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the bin count.

Examples `MEASUrement:MEAS1:BIN 2` sets the bin count to 2.

`MEASUrement:MEAS1:BIN?` might return `:MEASUREMENT:MEAS1:BIN 1` indicating the bin count is 1.

MEASUrement:MEAS<x>:BITCfgmode

This command sets or queries whether the measurement returns the mean or mode statistic result when the measurement type is bit amplitude/high/low. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:BITCfgmode {MEAN|MODE}
MEASUrement:MEAS<x>:BITCfgmode?
```

Arguments `MEAS<x>` specifies the measurement number.

`MEAN` specifies that the measurement returns results for each bit.

`MODE` specifies that the measurement returns the mode of its results.

Examples `MEASUrement:MEAS1:BITCfgmode MODE` specifies that the measurement returns the mode of its results.

`MEASUrement:MEAS1:BITCfgmode?` might return `:MEASUREMENT:MEAS1:BITCFGMODE MEAN` indicating the configuration mode is set to MEAN.

MEASUrement:MEAS<x>:BITEnd

This command sets or queries the bit end as a percentage of the unit interval. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:BITEnd <NR3>
MEASUrement:MEAS<x>:BITEnd?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the bit end.

Examples `MEASUrement:MEAS2:BITEnd 60` sets the bit end to 60% of the unit interval.

`MEASUrement:MEAS2:BITEnd?` might return `:MEASUREMENT:MEAS2:BITEND 50.0000` indicating the bit end is 50%.

MEASUrement:MEAS<x>:BITPcnt

This command sets or queries the bit center as a percentage of the unit interval. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:BITPcnt <NR3>
MEASUrement:MEAS<x>:BITPcnt?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the bit center percentage value to be measured for the bit type selected.

Examples `MEASUrement:MEAS2:BITPcnt 55` sets the bit center to 55% of the unit interval.

`MEASUrement:MEAS2:BITPcnt?` might return `:MEASUREMENT:MEAS2:BITPCNT 50.0000` indicating bit center is 50% of the unit interval.

MEASUrement:MEAS<x>:BITSTart

This command sets or queries the bit start as a percentage of the unit interval. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:BITSTart <NR3>
MEASUrement:MEAS<x>:BITSTart?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the bit start.

Examples `MEASUrement:MEAS2:BITSTart 45` sets the bit start to 45% of the unit interval.

`MEASUrement:MEAS2:BITSTart?` might return `:MEASUREMENT:MEAS2:BITSTART 50.0000` indicating the bit start is 50% of the unit interval.

MEASUrement:MEAS<x>:BITType

This command sets or queries the bit type for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:BITType
{ALLBits|TRANSition|NONTRANsition}
MEASUrement:MEAS<x>:BITType?
```

Arguments `MEAS<x>` specifies the measurement number.

`ALLBits` specifies that the measurement returns results for all bits.

`TRANSition` specifies that the measurement returns results for transitions bit only.

`NONTRANsition` specifies that the measurement returns results for non-transition bits only.

Examples `MEASUrement:MEAS1:BITType TRANSITION` specifies that the measurement returns results for transition bits only.

`MEASUrement:MEAS1:BITType?` might return `:MEASUREMENT:MEAS1:BITTYPE ALLBITS` indicating that measurements return results for all bits.

MEASUrement:MEAS<x>:BURSTEDGTYPe

This command sets or queries the burst edge type for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:BURSTEDGTYPe {RISE|FALL}
MEASUrement:MEAS<x>:BURSTEDGTYPe?
```

Arguments

`MEAS<x>` specifies the measurement number.

`RISE` specifies a burst with a rising edge.

`FALL` specifies a burst with a falling edge.

Examples

`MEASUrement:MEAS2:BURSTEDGTYPe FALL` specifies a burst with a falling edge.

`MEASUrement:MEAS2:BURSTEDGTYPe?` might return `:MEASUREMENT:MEAS2:BURSTEDGTYPE RISE` indicating a burst with a rising edge.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MAXimum? (Query Only)

This query-only command returns the maximum cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MAXimum?`

Returns

The maximum cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples

`MEASUrement:MEAS2:CCRESUlts:ALLAcqs:MAXimum?` might return `:MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MAXIMUM 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MEAN? (Query Only)

This query-only command returns the mean cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group

Measurement

Syntax

`MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MEAN?`

Returns

The mean cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples `MEASUrement:MEAS2CCRESUlts:ALLAcqs:MEAN?` might return `:MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MEAN 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MINimum? (Query Only)

This query-only command returns the minimum cycle-cycle value for the specified measurement for all acquisitions. Measurements are specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:MINimum?`

Returns The minimum cycle-cycle statistic value for the specified measurement for all acquisitions.

Examples `MEASUrement:MEAS2:CCRESUlts:ALLAcqs:MINimum?` might return `:MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:MINIMUM 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:PK2PK? (Query Only)

This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for all acquisitions. Measurements are specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:PK2PK?`

Returns The peak to peak cycle-cycle statistic value for the specified measurement all acquisitions.

Examples `MEASUrement:MEAS2:CCRESUlts:ALLAcqs:PK2PK?` might return `:MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:PK2PK 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:POPUlation? (Query Only)

This query-only command returns the population of all cycle-cycle statistics for the specified measurement for all acquisitions accumulated since statistics were last reset. Measurements are specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:POPUlation?`

Returns The population of all cycle-cycle statistics for the specified measurement accumulated over all acquisitions since statistics were last reset.

Examples `MEASUrement:MEAS2:CCRESUlts:ALLAcqs:POPUlation?` might return `:MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:POPULATION 0` indicating no statistics for the measurement.

MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:STDDev? (Query Only)

This query-only command returns the standard deviation cycle-cycle for the specified measurement for all acquisitions. Measurements are specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:STDDev?`

Returns The standard deviation cycle-cycle statistic value for the specified measurement all acquisitions.

Examples `MEASUrement:MEAS2:CCRESUlts:ALLAcqs:STDDev?` might return `:MEASUREMENT:MEAS2:CCRESULTS:ALLACQS:STDDEV 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MAXimum? (Query Only)

This query-only command returns the maximum cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MAXimum?`

Returns The maximum cycle-cycle statistic value for the specified measurement for the current acquisition.

Examples `MEASUrement:MEAS2:CCRESUlts:CURRentacq:MAXimum?` might return `:MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:MAXIMUM 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MEAN? (Query Only)

This query-only command returns the mean cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MEAN?`

Returns The mean cycle-cycle statistic value for the specified measurement for the current acquisition.

Examples `MEASUrement:MEAS1:CCRESUlts:CURRentacq:MEAN?` might return `:MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:MEAN 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MINimum? (Query Only)

This query-only command returns the minimum cycle-cycle value for the specified measurement for the current acquisition. Measurements are specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:MINimum?`

Returns The minimum cycle-cycle statistic value for the specified measurement for the current acquisition.

Examples `MEASUrement:MEAS2:CCRESUlts:CURRentacq:MINimum?` might return `:MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:MINIMUM 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:PK2PK? (Query Only)

This query-only command returns the peak to peak cycle-cycle statistic for the specified measurement for the current acquisition. Measurements are specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:PK2PK?`

Returns The peak to peak cycle-cycle statistic value for the specified measurement for the current acquisition.

Examples `MEASUrement:MEAS1:CCRESUlts:CURRentacq:PK2PK?` might return `:MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:PK2PK 9.91E+37`.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:POPUlation? (Query Only)

This query-only command returns the population of the cycle-cycle statistics for the specified measurement for the current acquisition. Measurements are specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:POPUlation?`

Returns The population of the cycle-cycle statistics for the specified measurement for the current acquisition.

Examples `MEASUrement:MEAS1:CCRESUlts:CURRentacq:POPUlation?` might return `:MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:POPULATION 0` indicating the population is 0.

MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:STDDev? (Query Only)

This query-only command returns the standard deviation cycle-cycle for the specified measurement for the current acquisition. Measurements are specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:CCRESUlts:CURRentacq:STDDev?`

Returns The standard deviation cycle-cycle statistic value for the specified measurement for the current acquisition.

Examples `MEASUrement:MEAS1:CCRESUlts:CURRentacq:STDDev?` might return `:MEASUREMENT:MEAS1:CCRESULTS:CURRENTACQ:STDDEV 9.91E+37`.

MEASUrement:MEAS<x>:CLOCKRecovery:ADVanced:METHod

This command sets or queries the advanced clock recovery method when advanced clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:ADVanced:METHod
{NONE|NOMinal|PATTern}
MEASUrement:MEAS<x>:CLOCKRecovery:ADVanced:METHod?
```

Arguments `MEAS<x>` specifies the measurement number.

`NOMinal` sets the advanced CRD method to Nominal Data Rate.

`PATTern` sets the advanced CDR method to use a Known Data Pattern.

`NONE` sets no advanced CRD method.

Examples `MEASUrement:MEAS1:CLOCKRecovery:ADVanced:METHod PATTERN` sets the advanced CDR method to use a Known Data Pattern.

`MEASUrement:MEAS1:CLOCKRecovery:ADVanced:METHod?` might return `:MEASUREMENT:MEAS1:CLOCKRECOVERY:ADVANCED:METHOD NONE` indicating no advanced CRD method.

MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKFrequency

This command sets or queries the clock frequency used when fixed constant clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKFrequency <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKFrequency?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the clock frequency.

Examples MEASUrement:MEAS1:CLOCKRecovery:CLOCKFrequency 2.0E+9 sets the frequency to 2.0 GHz.

MEASUrement:MEAS1:CLOCKRecovery:CLOCKFrequency? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:CLOCKFREQUENCY 2.5000E+9 indicating the frequency is 2.5 GHz.

MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier

This command sets or queries the clock multiplier used when explicit clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:CLOCKMultiplier?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the clock multiplier.

Examples MEASUrement:MEAS1:CLOCKRecovery:CLOCKMultiplier 1.5 sets the clock multiplier to 1.5.

MEASUrement:MEAS1:CLOCKRecovery:CLOCKMultiplier? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:CLOCKMULTIPLIER 1.0000 indicating the clock multiplier is 1.0.

MEASUrement:MEAS<x>:CLOCKRecovery:CONSTCLOCKMODe

This command sets or queries the constant clock mode used when constant clock recovery is used for the measurement. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:CONSTCLOCKMODe
{MEAN|MEDian|FIXed}
```

Arguments

MEAS<x> specifies the measurement number.

MEAN specifies that clock recovery uses the mean of the clock signal as the clock frequency.

MEDian specifies that clock recovery uses the mode of the clock signal as the clock frequency.

FIXed specifies that clock recovery uses the value set by the user as the clock frequency.

Examples

MEASUrement:MEAS1:CLOCKRecovery:CONSTCLOCKMODe FIXED specifies that clock recovery uses the value set by the user as the clock frequency.

MEASUrement:MEAS1:CLOCKRecovery:CONSTCLOCKMODe? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:CONSTCLOCKMODE MEAN indicating that clock recovery uses the mean of the clock signal as the clock frequency.

MEASUrement:MEAS<x>:CLOCKRecovery:DAMPing

This command sets or queries the damping value used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:DAMPing <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:DAMPing?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the clock recovery damping value.

Examples

MEASUrement:MEAS1:CLOCKRecovery:DAMPing 700.0E-3 sets the damping value to 0.70.

MEASUrement:MEAS1:CLOCKRecovery:DAMPing? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:DAMPING 700.0000E-3 indicating the damping value is 0.70.

MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath

This command sets or queries the file containing the data pattern used when known data pattern clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath <QString>
MEASUrement:MEAS<x>:CLOCKRecovery:DATAPath?
```

Arguments MEAS<x> specifies the measurement number.

<QString> is the file containing the data pattern to be used for known data pattern clock recovery.

Examples MEASUrement:MEAS1:CLOCKRecovery:DATAPath "TEST125.txt specifies the file containing the data pattern is TEST125.txt.

MEASUrement:MEAS1:CLOCKRecovery:DATAPath? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:DATAPATH "PRBS127.txt" indicating the file containing the data pattern is PRBS127.txt.

MEASUrement:MEAS<x>:CLOCKRecovery:DATARate

This command sets or queries the nominal data bit rate when nominal data rate clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:DATARate <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:DATARate?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the value for the Nominal data rate.

Examples MEASUrement:MEAS1:CLOCKRecovery:DATARate 2.0000E+9 sets the data rate to 2.0 Gb/s.

MEASUrement:MEAS1:CLOCKRecovery:DATARate? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:DATARATE 2.5000E+9 indicating the data rate is 2.5 Gb/s.

MEASUrement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODe

This command sets or queries the explicit clock mode used when explicit clock recovery is used for the measurement. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODe
{EDGE|PLL}
MEASUrement:MEAS<x>:CLOCKRecovery:EXPLICITCLOCKMODe?
```

Arguments `MEAS<x>` specifies the measurement number.

`EDGE` specifies the clock edge.

`PLL` specifies the phase locked loop.

Examples `MEASUrement:MEAS1:CLOCKRecovery:EXPLICITCLOCKMODe EDGE` sets the clock mode to edge.

`MEASUrement:MEAS1:CLOCKRecovery:EXPLICITCLOCKMODe?` might return `:MEASUREMENT:MEAS1:CLOCKRECOVERY:EXPLICITCLOCKMODE EDGE` indicating the explicit clock mode is edge.

MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal

This command sets or queries the clock recovery settings global flag for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal {OFF|ON|0|1}
MEASUrement:MEAS<x>:CLOCKRecovery:GLOBal?
```

Arguments `MEAS<x>` specifies the measurement number.

`OFF` clock recovery settings are changed independently for each individual measurement.

`ON` applies global clock recovery settings to all the measurements' clock recovery settings.

`0` clock recovery settings are changed independently for each individual measurement.

`1` applies global clock recovery settings to all the measurements' clock recovery settings.

Examples `MEASUrement:MEAS1:CLOCKRecovery:GLOBal 1` applies global clock recovery settings to all the measurements' clock recovery settings.

MEASUrement:MEAS1:CLOCKRecovery:GLOBal? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:GLOBAL 0 indicating clock recovery settings are changed independently for each individual measurement.

MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth

This command sets or queries the JTF bandwidth used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:JTFBandwidth?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the clock recovery JTF bandwidth.

Examples MEASUrement:MEAS1:CLOCKRecovery:JTFBandwidth 1.10E+6 sets the bandwidth to 1.1 MHz.

MEASUrement:MEAS1:CLOCKRecovery:JTFBandwidth? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:JTFBANDWIDTH 1.0000E+6 indicating the bandwidth is 1.0 MHz.

MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth

This command sets or queries the loop bandwidth used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:LOOPBandwidth?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the clock recovery loop bandwidth.

Examples MEASUrement:MEAS1:CLOCKRecovery:LOOPBandwidth 1.10E+6 sets the bandwidth to 1.1 Mhz.

MEASUrement:MEAS1:CLOCKRecovery:LOOPBandwidth? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:LOOPBANDWIDTH 1.0000E+6 indicating the bandwidth is 1.0 MHz.

MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate

This command sets or queries how often the clock is calculated when constant clock recovery is used for the measurement. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate
{FIRST|EVERY}
MEASUrement:MEAS<x>:CLOCKRecovery:MEANAUTOCalculate?
```

Arguments

MEAS<x> specifies the measurement number.

FIRST calculates the clock on the first acquisition.

EVERY calculates the clock on every acquisition.

Examples

MEASUrement:MEAS1:CLOCKRecovery:MEANAUTOCalculate EVERY causes the clock to be calculated on every acquisition.

MEASUrement:MEAS1:CLOCKRecovery:MEANAUTOCalculate? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:MEANAUTOCALCULATE FIRST indicating the clock is calculated on the first acquisition.

MEASUrement:MEAS<x>:CLOCKRecovery:METHod

This command sets or queries the clock recovery method for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:METHod
{PLL|CONSTANTCLOCK|EXPLICITCLOCK}
MEASUrement:MEAS<x>:CLOCKRecovery:METHod?
```

Arguments MEAS<x> specifies the measurement number.

PLL specifies a phase locked loop clock recovery method.

CONSTANTCLOCK specifies using a constant clock.

EXPLICITCLOCK specifies using an explicit clock.

Examples MEASUrement:MEAS1:CLOCKRecovery:METHod PLL specifies a phase locked loop clock recovery method.

MEASUrement:MEAS1:CLOCKRecovery:METHod? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:METHOD CONSTANTCLOCK indicating a constant clock is in use.

MEASUrement:MEAS<x>:CLOCKRecovery:MODel

This command sets or queries the PLL clock recovery model used when PLL clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:MODel {TYPE1|TYPE2}
MEASUrement:MEAS<x>:CLOCKRecovery:MODel?
```

Arguments MEAS<x> specifies the measurement number.

Arguments are the clock recovery model type.

Examples MEASUrement:MEAS1:CLOCKRecovery:MODel TYPE1 set the clock recovery model lto type 1.

MEASUrement:MEAS1:CLOCKRecovery:MODel? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:MODEL TYPE1 indicating the clock recovery model is type 1.

MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset

This command sets or queries the offset value used when explicit clock recovery is used for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset <NR3>
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the clock offset.

Examples

MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFset 1.0e-9 sets the clock offset to 1 ns.

MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFset? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:NOMINALOFFSET 0.0E+0 indicating the offset is 0.0 ns.

MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset:SELECTIONtype

This command sets or queries the offset type used when explicit clock recovery is used for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset:
SELECTIONtype {AUTO|MANUAL}
MEASUrement:MEAS<x>:CLOCKRecovery:NOMINALOFFset:
SELECTIONtype?
```

Arguments

MEAS<x> specifies the measurement number.

AUTO automatically calculates the offset.

MANUAL allows the user to set the offset.

Examples

MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFset:SELECTIONtype AUTO sets the offset selection type to auto.

MEASUrement:MEAS1:CLOCKRecovery:NOMINALOFFset:SELECTIONtype? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:NOMINALOFFSET:SELECTIONTYPE MANUAL indicating the selection type is manual.

MEASUrement:MEAS<x>:CLOCKRecovery:STAndard

This command sets or queries the communications standard when PLL clock recovery is used for the measurement. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CLOCKRecovery:STAndard {CUSTom|
ENET100|FW1394BS400B|FW1394BS800B|FW1394BS1600B|FBD1|
FBD2|FBD3|FC133|FC266|FC531|FC1063|FC2125|FC4250|
FC8500|ENET1000|IBA2500|IBA_GEN2|OC1|OC3|OC12|OC48|
PCIE_GEN1|PCIE_GEN2|PCIE_GEN3|RIO125|RIO250|RIO3125|
SAS15_NOSSC|SAS3_NOSSC|SAS6_NOSSC|SAS12_NOSSC|SAS15_SSC|
SAS3_SSC|SAS6_SSC|SAS12_SSC|SATA_GEN1|SATA_GEN2|
SATA_GEN3|USB3|XAUI|XAUI_GEN2}
```

Arguments MEAS<x> specifies the measurement number.

Arguments are the clock recovery standards.

Examples MEASUrement:MEAS1:CLOCKRecovery:STAndard PCIE_GEN2 sets the standard to PCIE_GEN2.

MEASUrement:MEAS1:CLOCKRecovery:STAndard? might return :MEASUREMENT:MEAS1:CLOCKRECOVERY:STANDARD PCIE_GEN1 indicating the standard is PCIE_GEN1.

MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE

This command sets or queries whether a filter is used for the measurement when the measurement type is AC common mode. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE {OFF|ON|0|1}
MEASUrement:MEAS<x>:COMMONMode:FILTers:STATE?
```

Arguments MEAS<x> specifies the measurement number.

OFF turns the filter off.

ON turns the filter on.

0 turns the filter off.

1 turns the filter on.

Examples `MEASUrement:MEAS1:COMMONMode:FILTers:STATE ON` sets the filter to on.

`MEASUrement:MEAS1:COMMONMode:FILTers:STATE?` might return `:MEASUREMENT:MEAS1:COMMONMODE:FILTERS:STATE 0` indicating the filter is off.

MEASUrement:MEAS<x>:COMMONMode:SOURCEs

This command sets or queries the number of sources for the measurement when the measurement type is AC common mode. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:COMMONMode:SOURCEs {SINGLE|DOUBLE}
MEASUrement:MEAS<x>:COMMONMode:SOURCEs?
```

Arguments `MEAS<x>` specifies the measurement number.

`SINGLE` specifies a single source.

`DOUBLE` specifies double sources.

Examples `MEASUrement:MEAS1:COMMONMode:SOURCEs SINGLE` specifies a single source.

`MEASUrement:MEAS1:COMMONMode:SOURCEs?` might return `:MEASUREMENT:MEAS1:COMMONMODE:SOURCES DOUBLE` indicating sources are set to double.

MEASUrement:MEAS<x>:CONFIGuration

This command sets or queries the Input or Output Configuration of the measurement. Measurements are specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:CONFIGuration {INput|OUTput}
MEASUrement:MEAS<x>:CONFIGuration?
```

Arguments `MEAS<x>` specifies the measurement number.

`INput` specifies input as the configuration of the measurement.

OUTput specifies output as the configuration of the measurement.

Examples MEASUrement:MEAS1:CONFIGuration INput sets configuration of measurement 1 to input.

MEASUrement:MEAS1:CONFIGuration? might return :MEASUrement:MEAS1:CONFIGuration OUTput, indicating the configuration of measurement 1 is output.

MEASUrement:MEAS<x>:CUSTOMLIMITSFile

This command sets or queries the custom limits file path for custom harmonics standard. Measurements are specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:CUSTOMLIMITSFile <Qstring>
MEASUrement:MEAS<x>:CUSTOMLIMITSFile?
```

Arguments MEAS<x> specifies the measurement number.

<Qstring> specifies the location of limits file.

Examples MEASUrement:MEAS1:CUSTOMLIMITSFile <C:\MyDir\Limits.csv> sets the location of the limits file as C:\MyDir\.

MEASUrement:MEAS1:CUSTOMLIMITSFile? might return :MEASUrement:MEAS1:CUSTOMLIMITSFile <C:\MyDir\Limits.csv>, indicating the location of the Limits.csv file.

MEASUrement:MEAS<x>:CYCLemode

This command sets or queries the cycle mode for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:CYCLemode {RECORD|CYCLE}
MEASUrement:MEAS<x>:CYCLemode?
```

Arguments

MEAS<x> specifies the measurement number.

RECORD specifies that the measurement is taken over the whole record.

CYCLE specifies that measurements are taken on each cycle of the source.

Examples

MEASUrement:MEAS1:CYCLemode CYCLE sets the measurement to be taken over each cycle of the source.

MEASUrement:MEAS1:CYCLemode? might return :MEASUREMENT:MEAS1:CYCLEMODE RECORD indicating the measurement is taken over the whole record.

MEASUrement:MEAS<x>:DELay:EDGE<x>

This command sets or queries the 'to edge' type when EDGE<x> is EDGE1 and the 'from edge' type when EDGE<x> is EDG2, for the measurement when the measurement type is DELAY. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:DELay:EDGE<x>
{FALL|RISe|BOTH|SAMEas|OPPositeas}
MEASUrement:MEAS<x>:DELay:EDGE<x>?
```

Related Commands

MEASUrement:MEAS<x>:TOEdge

Arguments

MEAS<x> specifies the measurement number.

FALL specifies the falling edge of the waveform.

RISE specifies the rising edge of the waveform.

BOTH specifies both a rising and falling edge of the waveform.

SAMEas specifies that both edges of the waveform are the same.

OPPositeas specifies that the edges of the waveform are not the same.

Examples

MEASUREMENT:MEAS3:DELAY:EDGE2 RISE specifies that the rising edge of the waveform be used for measurement 3.

MEASUREMENT:MEAS1:DELAY:EDGE2? might return :MEASUREMENT:MEAS1:DELAY:EDGE2 FALL, indicating that the falling edge of the waveform is being used for measurement 1.

MEASUrement:MEAS<x>:DISPlaystat:ENABle

This command turns on and off display of statistics in measurement badges in the user interface. This command affects only the display of statistics, basic-statistics are computed regardless of the state of this command. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:DISPlaystat:ENABle {OFF|ON|<NR1>}
MEASUrement:MEAS<x>:DISPlaystat:ENABle?
```

Arguments

MEAS<x> specifies the measurement number.

OFF turns off the display of statistics in measurement badges.

ON turns on the display of statistics in measurement badges.

<NR1> = 0 turns off the display of statistics in the measurement badge, any other value turns on the display of statistics.

Examples

MEASUrement:MEAS<x>:DISPlaystat:ENABle 0 turns off the display of statistics in the measurement badge.

MEASUrement:MEAS<x>:DISPlaystat:ENABle? might return :MEASUREMENT:MEAS1:DISPLAYSTAT:ENABLE 1 indicating that the display of statistics in the measurement badge is on.

MEASUrement:MEAS<x>:EDGEIncre

This command sets or queries the edge increment value for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGEIncre <NR3>
MEASUrement:MEAS<x>:EDGEIncre?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the measurements edge increment value.

Examples `:MEASUREMENT:MEAS2:EDGEINCRE?` might return `:MEASUREMENT:MEAS2:EDGEINCRE 1.0000` indicating the edge increment value is set to 1.0000.

MEASUrement:MEAS<x>:EDGEQUALifier

This command sets or queries the edge qualifier source.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM, IMDA

Syntax
```
MEASUrement:MEAS<x>:EDGEQUALifier
{CH1|CH2|CH3|CH4|CH5|CH6|CH7|CH8|MATH<x>|REF<x>}
MEASUrement:MEAS<x>:EDGEQUALifier?
```

Arguments `MEAS<x>` is the measurement number.

`CH1–CH8` is the edge qualifier source channel.

`MATH<x>` is the edge qualifier source math waveform; <x> is ≥1.

`REF<x>` is the edge qualifier source reference waveform; <x> is ≥1.

Examples `MEASUrement:MEAS1:EDGEQUALifier CH4` sets the edge qualifier source for measurement 1 to Channel 4.

`MEASUrement:MEAS3:EDGEQUALifier?` might return `UNDEFINED`, indicating that there is no edge qualifier source set for measurement 3.

MEASUrement:MEAS<x>:EDGEQUALONE

This command set or queries the measurement 1 input edge qualifier one source for Efficiency measurement.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:EDGEQUALONE <Qstring>
MEASUrement:MEAS<x>:EDGEQUALONE?
```

Arguments MEAS<x> is the measurement number.

<Qstring> specifies the edge qualifier source at input configuration.

Examples MEASUrement:MEAS1:EDGEQUALONE Vac sets the edge qualifier source for measurement 1 to Vac at input configuration.

MEASUrement:MEAS1:EDGEQUALONE? might return Vac, indicating that the edge qualifier source for measurement 1 is Vac at input configuration.

MEASUrement:MEAS<x>:EDGEQUALTWO

This command sets or queries the measurement 1 output edge qualifier two source for Efficiency measurement.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:EDGEQUALTWO <Qstring>
MEASUrement:MEAS<x>:EDGEQUALTWO?
```

Arguments MEAS<x> is the measurement number.

<Qstring> specifies the edge qualifier source at output configuration.

Examples MEASUrement:MEAS1:EDGEQUALTWO Vxz sets the edge qualifier source for measurement 1 to Vxz at output configuration.

MEASUrement:MEAS1:EDGEQUALTWO? might return Vxz, indicating that the edge qualifier source for measurement 1 is Vxz at output configuration.

MEASUrement:MEAS<x>:EDGES:FROMLevel

This command sets or queries the 'from level' edge for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:EDGES:FROMLevel {MID|LOW|HIGH}
MEASUrement:MEAS<x>:EDGES:FROMLevel?
```

Arguments

MEAS<x> specifies the measurement number.

MID specifies the MID level.

HIGH specifies the HIGH level.

LOW specifies the LOW level.

Examples

MEASUrement:MEAS1:EDGES:FROMLevel HIGH set the from level to the high level

MEASUrement:MEAS1:EDGES:FROMLevel? might return :MEASUREMENT:MEAS1:EDGES:FROMLEVEL LOW indicating the from level is the low level.

MEASUrement:MEAS<x>:EDGES:LEVel

This sets or queries the level type for the 'time outside level' measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGES:LEVel {HIGH|LOW|BOTH}
MEASUrement:MEAS<x>:EDGES:LEVel?
```

Arguments

MEAS<x> specifies the measurement number.

HIGH specifies the HIGH level.

LOW specifies the LOW level.

BOTH specifies both the HIGH and LOW level.

Examples

MEASUrement:MEAS1:EDGES:LEVel LOW sets the level to the low level.

MEASUrement:MEAS1:EDGES:LEVel? might return :MEASUREMENT:MEAS1:EDGES:LEVEL HIGH indicating the level is set to the high level.

MEASUrement:MEAS<x>:EDGES:LOWERFREQuency

This command sets or queries the lower frequency for the measurement when the measurement type is phase noise. Lower frequencies are ignored. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:EDGES:LOWERFREQuency <NR3>
MEASUrement:MEAS<x>:EDGES:LOWERFREQuency?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the lower frequency of the edge.

Examples `MEASUrement:MEAS1:EDGES:LOWERFREQuency 1.0e3` sets the lower frequency to 1 kHz.

`MEASUrement:MEAS1:EDGES:LOWERFREQuency?` might return `:MEASUREMENT:MEAS1:EDGES:LOWERFREQUENCY 0.0E+0` indicating the lower frequency is set to 0.0 Hertz.

MEASUrement:MEAS<x>:EDGES:N

The command sets or queries the number of accumulation cycles for the measurement when the measurement type is nperiod. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:EDGES:N <NR3>
MEASUrement:MEAS<x>:EDGES:N?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the maximum number of edges used by the measurement.

Examples `measurement:meas1:edges:n 2` sets the number of edges to 2.

`measurement:meas1:edges:n?` might return `:MEASUREMENT:MEAS1:EDGES:N 6.0000`.

MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod

This command sets or queries the slew rate method for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod {NOMinal|DDR}
MEASUrement:MEAS<x>:EDGES:SLEWRATEMethod?
```

Arguments `MEAS<x>` specifies the measurement number.

`NOMinal` specifies the nominal slew rate method.

`DDR` specifies the DDR slew rate method.

Examples `measurement:meas1:edges:slewratemethod NOMINAL` specifies the nominal slew rate method.

`measurement:meas1:edges:slewratemethod?` might return `:MEASUREMENT:MEAS1:EDGES:SLEWRATEMETHOD NOMINAL` indicating the slew rate method is set to NOMINAL.

MEASUrement:MEAS<x>:EDGES:TOLevel

This command sets or queries the 'to level' edge for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:EDGES:TOLevel {HIGH|MID|LOW}
MEASUrement:MEAS<x>:EDGES:TOLevel?
```

Arguments `MEAS<x>` specifies the measurement number.

`HIGH` specifies the HIGH level.

`MID` specifies the MID level.

`LOW` specifies the LOW level.

Examples `MEASUrement:MEAS1:EDGES:TOLevel HIGH` specifies the High to level.

`MEASUrement:MEAS1:EDGES:TOLevel?` might return `:MEASUREMENT:MEAS1:EDGES:TOLEVEL LOW` indicating the to level edge is set to the Low level.

MEASUrement:MEAS<x>:EDGES:UPPERFREQuency

This command sets or queries the upper frequency for the measurement when the measurement type is phase noise. Higher frequencies are ignored. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGES:UPPERFREQuency <NR3>
MEASUrement:MEAS<x>:EDGES:UPPERFREQuency?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the upper frequency of the edge.

Examples

`MEASUrement:MEAS1:EDGES:UPPERFREQuency 1.5` sets the upper frequency to 1.5 MHz.

`MEASUrement:MEAS1:EDGES:UPPERFREQuency?` might return `:MEASUREMENT:MEAS1:EDGES:UPPERFREQUENCY 1.0000E+6` indicating the upper frequency is 1.0 MHz.

MEASUrement:MEAS<x>:EDGE<x>

This command sets or queries the type of the specified edge, rise or fall, for the measurement. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:EDGE<x> {RISE|FALL|BOTH}
```

Arguments

`MEAS<x>` specifies the measurement number.

`RISE` specifies the rising edge.

`FALL` specifies the falling edge.

`BOTH` specifies either the rising or falling edge.

Examples

`MEASUREMENT:MEAS2:EDGE1?` might return `:MEASUREMENT:MEAS2:EDGE1 RISE` indicating that edge 1 of measurement 2 is the rising edge.

MEASUrement:MEAS<x>:FAILCount? (Query Only)

This command returns the number of measurement failures, if applicable, for the selected measure The measurement number is specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:FAILCount?`

Related Commands ACTONEVent:ENable

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE

ACTONEVent:MEASUrement:ACTION:SRQ:STATE

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE

Returns A running sum of the number of violations of the user specified measurement limit.

Examples `MEASUrement:MEAS2:FAILCount?` might return `:MEASUrement:MEAS2:FAILCount 2` indicating that measurement 2 has two measurement failures.

MEASUrement:MEAS<x>:FILTERRANGEFROM

This command sets or queries harmonics filter from order. Measurements are specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:FILTERRANGEFROM <NR3>
MEASUrement:MEAS<x>:FILTERRANGEFROM?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` specifies the filter range from value. The default value is 1.

Examples `MEASUrement:MEAS1:FILTERRANGEFROM 4` sets filter range from value for measurement 1 to 4.

MEASUrement:MEAS1:FILTERRANGEFROM? might return :MEASUrement:MEAS1:FILTERRANGEFROM 3, indicating the filter range from value is 3.

MEASUrement:MEAS<x>:FILTERRANGETO

This command sets or queries harmonics filter to order. Measurements are specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:FILTERRANGETO <NR3>
MEASUrement:MEAS<x>:FILTERRANGETO?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> specifies the filter range to value. The default value is 50.

Examples MEASUrement:MEAS1:FILTERRANGETO 50 sets filter range to value for measurement 1 to 50.

MEASUrement:MEAS1:FILTERRANGETO? might return :MEASUrement:MEAS1:FILTERRANGETO 50, indicating the filter range to value is 50.

MEASUrement:MEAS<x>:FILTers:BLANKingtime

This command sets or queries the filter blanking time for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:FILTers:BLANKingtime <NR3>
MEASUrement:MEAS<x>:FILTers:BLANKingtime?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the current filter blanking time.

Examples MEASUrement:MEAS1:FILTers:BLANKingtime 3.5 sets the blanking time to 3.5.

MEASUrement:MEAS1:FILTers:BLANKingtime? might return :MEASUREMENT:MEAS1:FILTERS:BLANKINGTIME 4.0000 indicating the filter blanking time is 4.0.

MEASUrement:MEAS<x>:FILTers:GLOBal

This command sets or queries the global flag for filter settings for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTers:GLOBal {OFF|ON|0|1}
MEASUrement:MEAS<x>:FILTers:GLOBal?
```

Arguments

MEAS<x> specifies the measurement number.

OFF causes filter settings to be changed independently for each individual measurement.

ON applies global filter settings to all the measurements' filter settings.

0 causes filter settings to be changed independently for each individual measurement.

1 applies global filter settings to all the measurements' filter settings.

Examples

MEASUrement:MEAS<x>:FILTers:GLOBal OFF causes filter settings to be changed independently for each individual measurement.

MEASUrement:MEAS<x>:FILTers:GLOBal? might return :MEASUREMENT:MEAS1:FILTERS:GLOBAL 1 indicating filter settings are for all measurements.

MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ

This command sets or queries the high pass filter frequency for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ <NR3>
MEASUrement:MEAS<x>:FILTers:HIGHPass:FREQ?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the current high pass filter frequency.

Examples

MEASUrement:MEAS1:FILTers:HIGHPass:FREQ 15.0E+6 sets the high pass frequency to 15.0 MHz.

MEASUrement:MEAS1:FILTers:HIGHPass:FREQ? might return :MEASUREMENT:MEAS1:FILTERS:HIGHPASS:FREQ 10.0000E+6 indicating the high pass frequency is set to 10.0 MHz.

MEASUrement:MEAS<x>:FILTers:HIGHPass:SPEC

This command sets or queries the high pass filter order for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FILTers:HIGHPass:SPEC
{NONE|FIRST|SECOND|THIRD}
MEASUrement:MEAS<x>:FILTers:HIGHPass:SPEC?
```

Arguments

MEAS<x> specifies the measurement number.

NONE specifies no filter.

FIRST specifies a first-order filter.

SECOND specifies a second-order filter.

THIRD specifies a third-order filter.

Examples

MEASUrement:MEAS1:FILTers:HIGHPass:SPEC FIRST specifies a first-order filter.

MEASUrement:MEAS1:FILTers:HIGHPass:SPEC? might return :MEASUREMENT:MEAS1:FILTERS:HIGHPASS:SPEC NONE indicating no high pass filter is used.

MEASUrement:MEAS<x>:FILTers:LOWPass:FREQ

This command sets or queries the low pass filter cutoff frequency for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:FILTers:LOWPass:FREQ <NR3>
MEASUrement:MEAS<x>:FILTers:LOWPass:FREQ?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the current low pass filter frequency.

Examples `MEASUrement:MEAS1:FILTers:LOWPass:FREQ 5.0E+6` sets the low pass frequency to 5.0 MHz.

`MEASUrement:MEAS1:FILTers:LOWPass:FREQ?` might return `:MEASUREMENT:MEAS1:FILTERS:LOWPASS:FREQ 10.0000E+6` indicating the low pass frequency is 10.0 MHz.

MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC

This command sets or queries the low pass filter order for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC
{NONE|FIRST|SECOND|THIRD}
MEASUrement:MEAS<x>:FILTers:LOWPass:SPEC?
```

Arguments `MEAS<x>` specifies the measurement number.

`NONE` specifies no filter.

`FIRST` specifies a first-order filter.

`SECOND` specifies a second-order filter.

`THIRD` specifies a third-order filter.

Examples `MEASUrement:MEAS1:FILTers:LOWPass:SPEC FIRST` specifies a first-order filter.

`MEASUrement:MEAS1:FILTers:LOWPass:SPEC?` might return `:MEASUREMENT:MEAS1:FILTERS:LOWPASS:SPEC NONE` indicating no low pass filter is used.

MEASUrement:MEAS<x>:FILTers:RAMPtime

This command sets or queries the filter ramp time for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:FILTers:RAMPtime <NR3>
MEASUrement:MEAS<x>:FILTers:RAMPtime?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the current filter ramp time.

Examples `MEASUrement:MEAS1:FILTers:RAMPtime 1.5` sets the ramp time to 1.5.

`MEASUrement:MEAS1:FILTers:RAMPtime?` might return `:MEASUREMENT:MEAS1:FILTERS:RAMPTIME 2.0000` indicating the ramp time is 2.0.

MEASUrement:MEAS<x>:FREQ

This command sets or queries switching frequency for Switching Ripple measurement. Measurements are specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:FREQ <NR3>
MEASUrement:MEAS<x>:FREQ?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` specifies the switching frequency. The default is 100k Hz, The minimum is 50 Hz, and the maximum is 1 GHz

Examples `MEASUrement:MEAS1:FREQ 50k` sets the switching frequency of measurement 1 to 50k Hz.

`MEASUrement:MEAS1:FREQ?` might return `:MEASUrement:MEAS1:FREQ 100k`, indicating the switching frequency is 100k Hz.

MEASUrement:MEAS<x>:FROMedge

This command sets or queries the from edge type for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMedge {RISe|FALL|BOTH}
MEASUrement:MEAS<x>:FROMedge?
```

Arguments

`MEAS<x>` specifies the measurement number.

`FALL` specifies the falling edge of the waveform.

`RISE` specifies the rising edge of the waveform.

`BOTH` specifies both the rising and falling edges of the waveform.

Examples

`MEASUrement:MEAS1:FROMedge RISE` specifies the rising edge of the waveform.

`MEASUrement:MEAS1:FROMedge?` might return `:MEASUREMENT:MEAS1:FROMEDGE BOTH` indicating both the rising and falling edges of the waveform.

MEASUrement:MEAS<x>:FROMEDGESEARCHDIRect

This command sets or queries the from edge search direction for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMEDGESEARCHDIRect {FORWard|BACKWard}
MEASUrement:MEAS<x>:FROMEDGESEARCHDIRect?
```

Arguments

`MEAS<x>` specifies the measurement number.

`FORWard` specifies a forward search from the edge.

`BACKWard` specifies a backward search from the edge.

Examples

`MEASUrement:MEAS1:FROMEDGESEARCHDIRect BACKWard` specifies a backward search from the edge.

MEASUrement:MEAS1:FROMEDGESEARCHDIRect? might return MEASUREMENT:MEAS1:FROMEDGESEARCHDIRECT FORWARD indicating a forward search from the edge.

MEASUrement:MEAS<x>:FROMSymbol:LOGIC2SOUrce

This command sets or queries the DDR digital measurement logic 1 source From symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMSymbol:LOGIC2SOUrce {H|L|X}
MEASUrement:MEAS<x>:FROMSymbol:LOGIC2SOUrce?
```

Related Commands MEASUrement:ADDMEAS

Arguments MEAS<x> specifies the measurement number.

H sets the logic source 1 From Symbol to H (High).

L sets the logic source 1 From Symbol to L (Low).

X sets the logic source 1 From Symbol to X (Don't care).

Examples MEASUrement:MEAS1:FROMSymbol:LOGIC2SOUrce H sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT) logic 1 source From Symbol to High.

MEASUrement:MEAS1:FROMSymbol:LOGIC2SOUrce? might return MEASUrement:MEAS1:FROMSymbol:LOGIC2SOUrce H, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT) From symbol is set to H.

MEASUrement:MEAS<x>:FROMSymbol:LOGIC3SOUrce

This command sets or queries the DDR digital measurement logic 2 source From symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:FROMSymbol:LOGIC3SOUrce {H|L|X}
MEASUrement:MEAS<x>:FROMSymbol:LOGIC3SOUrce?
```

Related Commands MEASUrement:ADDMEAS

Arguments `MEAS<x>` specifies the measurement number.

`H` sets the logic source 2 From Symbol to H (High).

`L` sets the logic source 2 From Symbol to L (Low).

`X` sets the logic source 2 From Symbol to X (Don't care).

Examples `MEASUrement:MEAS1:FROMSymbol:LOGIC3SOUrce H` sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT) logic 1 source From Symbol to High.

`MEASUrement:MEAS1:FROMSymbol:LOGIC3SOUrce?` might return `MEASUrement:MEAS1:FROMSymbol:LOGIC3SOUrce H`, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT) From symbol is set to H.

MEASUrement:MEAS<x>:FROMSymbol:LOGIC4SOUrce

This command sets or queries the DDR digital measurement logic 3 source From symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:FROMSymbol:LOGIC4SOUrce {H|L|X}
MEASUrement:MEAS<x>:FROMSymbol:LOGIC4SOUrce?
```

Related Commands MEASUrement:ADDMEAS

Arguments MEAS<x> specifies the measurement number.

H sets the logic source 3 From Symbol to H (High).

L sets the logic source 3 From Symbol to L (Low).

X sets the logic source 3 From Symbol to X (Don't care).

Examples MEASUrement:MEAS1:FROMSymbol:LOGIC4SOUrce H sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT) logic 1 source From Symbol to High.

MEASUrement:MEAS1:FROMSymbol:LOGIC4SOUrce? might return MEASUrement:MEAS1:FROMSymbol:LOGIC4SOUrce H, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT) From symbol is set to H.

MEASUrement:MEAS<x>:FROMSymbol:LOGIC5SOUrce

This command sets or queries the DDR digital measurement logic 4 source From symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMSymbol:LOGIC5SOUrce {H|L|X}
MEASUrement:MEAS<x>:FROMSymbol:LOGIC5SOUrce?
```

Related Commands MEASUrement:ADDMEAS

Arguments MEAS<x> specifies the measurement number.

H sets the logic source 4 From Symbol to H (High).

L sets the logic source 4 From Symbol to L (Low).

X sets the logic source 4 From Symbol to X (Don't care).

Examples

`MEASUrement:MEAS1:FROMSymbol:LOGIC5SOUrce H` sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT) logic 1 source From Symbol to High.

`MEASUrement:MEAS1:FROMSymbol:LOGIC5SOUrce?` might return `MEASUrement:MEAS1:FROMSymbol:LOGIC5SOUrce H`, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT) From symbol is set to H.

MEASUrement:MEAS<x>:FROMSymbol:MEASUREAT

This command sets or queries the DDR digital measurement From Symbol MeasureAT value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:FROMSymbol:MEASUREAT
{Start|Stop|ClockEdge}
MEASUrement:MEAS<x>:FROMSymbol:MEASUREAT?
```

Related Commands

MEASUrement:ADDMEAS

Arguments

`MEAS<x>` specifies the measurement number.

`Start` sets the FromSymbol MeasureAT to Start.

`Stop` sets the FromSymbol MeasureAT to Stop.

`ClockEdge` sets the FromSymbol MeasureAT to ClockEdge.

Examples

`MEASUrement:MEAS1:FROMSymbol:MEASUREAT Start` sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) FromSymbol MeasureAT value to Start.

`MEASUrement:MEAS1:FROMSymbol:MEASUREAT?` might return the `MEASUrement:MEAS1:FROMSymbol:MEASUREAT Stop`, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) FromSymbol MeasureAT value is set to Stop.

MEASUrement:MEAS<x>:FUNDCURRent

This command sets or queries the fundamental current value for IEC-Class C type harmonics standard. Measurements are specified by x.

Group Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:FUNDCURRent <NR3>
MEASUrement:MEAS<x>:FUNDCURRent?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> specifies the value for the fundamental current for class C harmonics standard.

\

Examples MEASUrement:MEAS1:FUNDCURRent 16.0000 specifies the fundamental current as 16.

MEASUrement:MEAS1:FUNDCURRent? might return :MEASUrement:MEAS1:FUNDCURRent 16.0000 indicating the fundamental current is 16.

MEASUrement:MEAS<x>:GATing

This command sets or queries the gating type for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing
{NONE|SCREEN|CURSor|LOGic|SEARch|TIMe}
MEASUrement:MEAS<x>:GATing?
```

Related Commands MEASUrement:MEAS<x>:GATing:ENDtime

MEASUrement:MEAS<x>:GATing:STARTtime

Arguments MEAS<x> is the measurement number for which to return a value.

NONE specifies measurements are taken across the entire record.

SCREEN turns on gating, using the left and right edges of the screen.

CURSor limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.

LOGic specifies that measurements are taken only when the logical state of other waveforms is true.

SEARCh specifies that measurements are taken only where the results of a user specified search are found.

TIMe limits measurements to the portion of the waveform between the Start and End gate times.

Examples

MEASUrement:MEAS1:GATing CURSOR limits measurements to the portion of the waveform between the vertical bar cursor.

MEASUrement:MEAS1:GATing? might return :MEASUREMENT:MEAS1:GATING NONE indicating measurements are taken across the entire record.

MEASUrement:MEAS<x>:GATing:ACTive

This command sets or queries the gating active level when the gating type is logic. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:ACTive {HIGH|LOW}
MEASUrement:MEAS<x>:GATing:ACTive?
```

Arguments

MEAS<x> specifies the measurement number.

HIGH takes a measurement when logic gating is High.

LOW takes a measurement when logic gating Low.

Examples

MEASUrement:MEAS1:GATing:ACTive LOW specifies taking measurements when logic gating is Low.

MEASUrement:MEAS1:GATing:ACTive? might return :MEASUREMENT:MEAS1:GATING:ACTIVE HIGH indicating measurements are taken when logic gating is High.

MEASUrement:MEAS<x>:GATing:ENDtime

Sets or queries the end gate time for the measurement when using Local gating.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:GATing:ENDtime <NR3>
MEASUrement:MEAS<x>:GATing:ENDtime?
```

Related Commands MEASUrement:MEAS<x>:GATing

Arguments MEAS<x> specifies the measurement number.

<NR3> is the time gating end gate time in seconds. The valid range is -10000 s to 10000 s.

Examples MEASUrement:MEAS4:GATing:ENDtime 5.31e-2 sets the local end gate time for Meas 4 to 53.1 ms.

MEASUrement:MEAS6:GATing:ENDtime? might return 1.05, indicating that the local end gate time for Meas 6 is 1.05 s.

MEASUrement:MEAS<x>:GATing:GLOBal

This command sets or queries the gating settings global flag. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:GATing:GLOBal {OFF|ON|0|1}
MEASUrement:MEAS<x>:GATing:GLOBal?
```

Arguments MEAS<x> specifies the measurement number.

OFF specifies gate settings can be changed independently for each individual measurement.

ON applies global gate settings to all the measurements' gate settings.

0 specifies gate settings can be changed independently for each individual measurement.

1 applies global gate settings to all the measurements' gate settings.

Examples MEASUrement:MEAS1:GATing:GLOBal OFF specifies gate settings can be changed independently for each individual measurement.

`MEASUrement:MEAS1:GATing:GLOBal?` might return `:MEASUREMENT:MEAS1:GATING:GLOBAL 1` indicating that gating settings apply to all measurements.

MEASUrement:MEAS<x>:GATing:HYSTeresis

This command sets or queries the gating hysteresis value when the gating type is logic. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:HYSTeresis <NR3>
MEASUrement:MEAS<x>:GATing:HYSTeresis?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the gating hysteresis.

Examples `MEASUrement:MEAS1:GATing:HYSTeresis 25.0E-3` sets the hysteresis to 25 mV.

`MEASUrement:MEAS1:GATing:HYSTeresis?` might return `:MEASUREMENT:MEAS1:GATING:HYSTERESIS 30.0000E-3` indicating the hysteresis is set to 30.0 mV.

MEASUrement:MEAS<x>:GATing:LOGICSource

This command sets or queries the gating data source when the gating type is logic. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:LOGICSource
{CH<x>|MATH<x>|REF<x>}
```

Arguments `MEAS<x>` specifies the measurement number.

Arguments are the sources for logic gating.

Examples `MEASUrement:MEAS1:GATing:LOGICSource CH3` sets the gating logic source to channel 3.

MEASUrement:MEAS1:GATing:LOGICSource? might return :MEASUREMENT:MEAS1:GATING:LOGICSOURCE CH2 indicating the gating logic source is channel 2.

MEASUrement:MEAS<x>:GATing:MIDRef

This command sets or queries the gating mid ref value when the gating type is logic. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:GATing:MIDRef <NR3>
MEASUrement:MEAS<x>:GATing:MIDRef?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the mid ref value for gating.

Examples MEASUrement:MEAS1:GATing:MIDRef 1.0E+0 sets the gating midref to 1.0.

MEASUrement:MEAS1:GATing:MIDRef? might return :MEASUREMENT:MEAS1:GATING:MIDREF 0.0E+0 indicating the midref value is set to 0.0.

MEASUrement:MEAS<x>:GATing:SEARCHSource

This command sets or queries the gating search source when the gating type is search. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:GATing:SEARCHSource SEARCH1
```

Arguments MEAS<x> specifies the measurement number.

SEARCH1 is the gating source for search gating.

Examples MEASUrement:MEAS1:GATing:SEARCHSource SEARCH1 sets the gating search source to SEARCH1.

MEASUrement:MEAS1:GATing:SEARCHSource? might return :MEASUREMENT:MEAS1:GATING:SEARCHSOURCE SEARCH1 indicating the gating search source is SEARCH1.

MEASUrement:MEAS<x>:GATing:STARTtime

Sets or queries the start gate time for the measurement when using Local gating.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:GATing:STARTtime <NR3>
MEASUrement:MEAS<x>:GATing:STARTtime?
```

Related Commands MEASUrement:MEAS<x>:GATing

Arguments MEAS<x> specifies the measurement number.

<NR3> is the time gating start gate time in seconds. The valid range is -10000 s to 10000 s.

Examples MEASUrement:MEAS1:GATing:STARTtime -1.5e-3 sets the local start gate time for Meas 1 to -1.5 ms.

MEASUrement:MEAS1:GATing:STARTtime? might return 3.0e-9, indicating that the local start gate time for Meas 1 is 3 ns.

MEASUrement:MEAS<x>:GLOBalref

This command sets or queries the reference levels global flag for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:GLOBalref {OFF|ON|0|1}
MEASUrement:MEAS<x>:GLOBalref?
```

Arguments MEAS<x> specifies the measurement number.

OFF allows ref levels to be set separately for each measurement.

ON applies the same ref levels to all measurements.

0 allows ref levels to be set separately for each measurement.

1 applies the same ref levels to all measurements.

Examples MEASUrement:MEAS1:GLOBalref 0 allows ref levels to be set separately for each measurement.

MEASUrement:MEAS1:GLOBalref? might return :MEASUREMENT:MEAS1:GLOBALREF 1 indicating the same ref levels apply to all measurements.

MEASUrement:MEAS<x>:HARMONICSCLass

This command sets or queries the class type for the harmonics measurement. Measurements are specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:HARMONICSCLass
{CLASSA|CLASSB|CLASSC|CLASSD}
MEASUrement:MEAS<x>:HARMONICSCLass?
```

Arguments MEAS<x> specifies the measurement number.

CLASSA-D specifies the class type.

Examples MEASUrement:MEAS1:HARMONICSCLass CLASSC sets the class type for measurement 1 as class C.

MEASUrement:MEAS1:HARMONICSCLass? might return :MEASUrement:MEAS1:HARMONICSCLass A, indicating the class type is set to class A.

MEASUrement:MEAS<x>:HARMONICSSOURce

This command sets or queries harmonics source. Measurements are specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:HARMONICSSOURce {CURRENT|VOLTAGE}
MEASUrement:MEAS<x>:HARMONICSSOURce?
```

Arguments MEAS<x> specifies the measurement number.

CURRENT specifies current as the harmonics source.

VOLTAGE specifies voltage as the harmonics source.

Examples MEASUrement:MEAS1:HARMONICSSOURce CURRENT sets the harmonics source for measurement 1 as current.

MEASUrement:MEAS1:HARMONICSSOURce? might return :MEASUrement:MEAS1:HARMONICSSOURce VOLTAGE, indicating the harmonics source as voltage.

MEASUrement:MEAS<x>:HARMONICSStd

This command sets or queries harmonics standard. Measurements are specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:HARMONICSStd {NONE|IEC|IEEE519|CUSTOM}
MEASUrement:MEAS<x>:HARMONICSStd?
```

Arguments MEAS<x> specifies the measurement number.

NONE specifies none as the harmonics test standard.

IEC specifies IEC as the harmonics test standard.

IEEE519 specifies IEEE519 as the harmonics test standard.

CUSTOM specifies that there is a custom harmonics test standard.

Examples MEASUrement:MEAS1:HARMONICSStd IEC sets the harmonics test standard for measurement 1 as IEC.

MEASUrement:MEAS1:HARMONICSStd? might return :MEASUrement:MEAS1:HARMONICSStd NONE, indicating no harmonics test standard has been set.

MEASUrement:MEAS<x>:HARMONICSUNits

This command sets or queries harmonics units. Measurements are specified by x.

Group Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:HARMONICSUNits {LOGARITHM|LINEAR}
MEASUrement:MEAS<x>:HARMONICSUNits?
```

Arguments

MEAS<x> specifies the measurement number.

LOGARITHM specifies logarithmic as the harmonics units.

LINEAR specifies linear as the harmonics units.

Examples

MEASUrement:MEAS1:HARMONICSUNits LINEAR sets the harmonics units standard for measurement 1 as linear.

MEASUrement:MEAS1:HARMONICSUNits? might return :MEASUrement:MEAS1:HARMONICSUNits LINEAR, indicating the harmonics units are linear.

MEASUrement:MEAS<x>:HIGHREFVoltage

This command sets or queries the high reference voltage value for the 'time outside level' measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:HIGHREFVoltage <NR3>
MEASUrement:MEAS<x>:HIGHREFVoltage?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the high reference voltage value for the selected configuration.

Examples

MEASUrement:MEAS1:HIGHREFVoltage 1.5 sets the high reference voltage to 1.5 V.

MEASUrement:MEAS1:HIGHREFVoltage? might return :MEASUREMENT:MEAS1:HIGHREFVOLTAGE 1.0000 indicating the high reference voltage is set to 1.0 V.

MEASUrement:MEAS<x>:IDLETime

This command sets or queries the idle time for the measurement when the measurement type is burst width. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:IDLETime <NR3>
MEASUrement:MEAS<x>:IDLETime?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the idle time.

Examples

MEASUrement:MEAS1:IDLETime 40.0E-6 sets the idle time to 40.0 µs.

MEASUrement:MEAS1:IDLETime? might return :MEASUREMENT:MEAS1:IDLETIME 50.0000E-6 indicating the idle time is 50.0 µs.

MEASUrement:MEAS<x>:INFILTers:LOWPass:FREQ

This command sets or queries the measurement input low pass filter frequency to the input filter cutoff frequency value on the scope. Measurements are specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:INFILTers:LOWPass:FREQ <NR2>
MEASUrement:MEAS<x>:INFILTers:LOWPass:FREQ?
```

Arguments

MEAS<x> specifies the measurement number.

<NR2> specifies the value for the input filter cutoff frequency.

Examples

MEASUrement:MEAS1:INFILTers:LOWPass:FREQ 2.000E+6 sets the input filter cutoff frequency for measurement 1 to 2.000E+6.

MEASUrement:MEAS1:INFILTers:LOWPass:FREQ? might return :MEASUrement:MEAS1:INFILTers:LOWPass:FREQ 2.000E+6, indicating the input filter cutoff frequency is 2.000E+6.

MEASUrement:MEAS<x>:INFILTers:LOWPass:SPEC

This command sets or queries the measurement input low pass filter specification filter order on the scope. Measurements are specified by x.

Group

Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:INFILTers:LOWPass:SPEC
{NONE|FIRST|SECOND|THIRD}
MEASUrement:MEAS<x>:INFILTers:LOWPass:SPEC?
```

Arguments

MEAS<x> specifies the measurement number.

NONE specifies the filter order none.

FIRST specifies the filter order first.

SECOND specifies the filter order second.

THIRD specifies the filter order third.

Examples

MEASUrement:MEAS1:INFILTers:LOWPass:SPEC SECOND sets the filter order of measurement 1 to second.

MEASUrement:MEAS1:INFILTers:LOWPass:SPEC? might return :MEASUrement:MEAS1:INFILTers:LOWPass:SPEC SECOND, indicating the filter order is set to second.

MEASUrement:MEAS<x>:INPUTLEVel

Sets or queries the input trigger voltage level.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:INPUTLEVel <NR2>
MEASUrement:MEAS<x>:INPUTLEVel?
```

Arguments

MEAS<x> is the measurement number.

<NR2> is the trigger voltage, in the range of -500 V to +500 V.

Examples

MEASUrement:MEAS1:INPUTLEVel 60 sets the input trigger level of measurement 1 to +60 volts.

MEASUrement:MEAS3:INPUTLEVel? might return 1.2500, indicating that the input trigger level of measurement 3 is +1.25 volts.

MEASUrement:MEAS<x>:INPUTPOwer

This command sets or queries the input power value for IEC-Class C and Class D harmonics standard .

Group Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:INPUTPOwer <NR3>
MEASUrement:MEAS<x>:INPUTPOwer?
```

Arguments MEAS<x> is the measurement number.

<NR3> specifies the input power value for class C harmonics standard.

Examples MEASUrement:MEAS1:INPUTPOwer 100W sets the input power value for class C harmonics standard to 100W.

MEASUrement:MEAS1:INPUTPOwer? might return :MEASUrement:MEAS1:INPUTPOwer 100W, indicating that the input power value for class C harmonics standard is 100W.

MEASUrement:MEAS<x>:JITTERSummary:DCD

This command sets or queries whether DCD is included in the jitter summary for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:DCD {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DCD?
```

Arguments MEAS<x> specifies the measurement number.

1 adds the DCD measurement as part of jitter summary.

0 do not add the DCD measurement as part of jitter summary.

Examples MEASUrement:MEAS1:JITTERSummary:DCD 0 specifies that the DCD measurement is not part of the jitter summary.

MEASUrement:MEAS1:JITTERSummary:DCD? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:DCD 1 indicating the DCD measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:DDJ

This command sets or queries whether DDJ is included in the jitter summary for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:DDJ {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DDJ?
```

Arguments

MEAS<x> specifies the measurement number.

1 adds the DDJ measurement as part of jitter summary.

0 do not add the DDJ measurement as part of jitter summary.

Examples

MEASUrement:MEAS4:JITTERSummary:DDJ 0 specifies that the DDJ measurement is not part of the jitter summary.

MEASUrement:MEAS4:JITTERSummary:DDJ? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:DDJ 1 indicating the DDJ measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:DJDD

This command sets or queries whether DJ-dd is included in the jitter summary for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:DJDD {0|1}
MEASUrement:MEAS<x>:JITTERSummary:DJDD?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the DJDD measurement as part of jitter summary.

0 do not add the DJDD measurement as part of jitter summary.

Examples

MEASUrement:MEAS4:JITTERSummary:DJDD 0 specifies the DJDD measurement is not part of the jitter summary.

MEASUrement:MEAS4:JITTERSummary:DJDD? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:DJDD 1 indicating the DJDD measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER

This command sets or queries whether EyeWidth@BER is included in the jitter summary for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER {0|1}
MEASUrement:MEAS<x>:JITTERSummary:EYEWIDTHBER?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the EyeWidth@BER measurement as part of jitter summary.

0 do not add the EyeWidth@BER measurement as part of jitter summary.

Examples

MEASUrement:MEAS4:JITTERSummary:EYEWIDTHBER 0 specifies that the EyeWidth@BER measurement is not part of the jitter summary.

MEASUrement:MEAS4:JITTERSummary:EYEWIDTHBER? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:EYEWIDTHBER 1 indicating EyeWidth@BER is part of the jitter summary

MEASUrement:MEAS<x>:JITTERSummary:NPJ

This command sets or queries whether NPJ is included in the jitter summary for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:JITTERSummary:NPJ {0|1}
MEASUrement:MEAS<x>:JITTERSummary:NPJ?
```

Arguments

MEAS<x> specifies the measurement number.

1 add the NPJ measurement as part of jitter summary.

0 do not add the NPJ measurement as part of jitter summary.

Examples `MEASUrement:MEAS4:JITTERSummary:NPJ 0` specifies that the NPJ measurement is not part of jitter summary.

`MEASUrement:MEAS4:JITTERSummary:NPJ?` might return `:MEASUREMENT:MEAS4:JITTERSUMMARY:NPJ 1` indicating the NPJ measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:PJ

This command sets or queries whether PJ is included in the jitter summary for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:JITTERSummary:PJ {0|1}
MEASUrement:MEAS<x>:JITTERSummary:PJ?
```

Arguments `MEAS<x>` specifies the measurement number.

`1` add the PJ measurement as part of jitter summary.

`0` do not add the PJ measurement as part of jitter summary.

Examples `MEASUrement:MEAS4:JITTERSummary:PJ 0` specifies the PJ measurement is not part of jitter summary.

`MEASUrement:MEAS4:JITTERSummary:PJ?` might return `:MEASUREMENT:MEAS4:JITTERSUMMARY:PJ 1` indicating the PJ measurement is part of jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:RJDD

This command sets or queries whether RJ-dd is included in the jitter summary for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:JITTERSummary:RJDD {0|1}
MEASUrement:MEAS<x>:JITTERSummary:RJDD?
```

Arguments `MEAS<x>` specifies the measurement number.

`1` add the RJ-dd measurement as part of jitter summary.

0 do not add the RJ-dd measurement as part of jitter summary.

Examples `MEASUrement:MEAS4:JITTERSummary:RJDD 0` specifies the RJ-dd measurement is not part of the jitter summary.

`MEASUrement:MEAS4:JITTERSummary:RJDD?` might return `:MEASUREMENT:MEAS4:JITTERSUMMARY:RJDD 1` indicating the RJ-dd measurement is part of the jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:TIE

This command sets or queries whether TIE is included in the jitter summary for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:JITTERSummary:TIE {0|1}
MEASUrement:MEAS<x>:JITTERSummary:TIE?
```

Arguments `MEAS<x>` specifies the measurement number.

1 add the TIE measurement as part of jitter summary.

0 do not add the TIE measurement as part of jitter summary.

Examples `MEASUrement:MEAS4:JITTERSummary:TIE 0` specifies the TIE measurement is not part of jitter summary.

`MEASUrement:MEAS4:JITTERSummary:TIE?` might return `:MEASUREMENT:MEAS4:JITTERSUMMARY:TIE 1` indicating the TIE measurement is part of jitter summary.

MEASUrement:MEAS<x>:JITTERSummary:TJBER

This command sets or queries whether TJ@BER is included in the jitter summary for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:JITTERSummary:TJBER {0|1}
MEASUrement:MEAS<x>:JITTERSummary:TJBER?
```

Arguments MEAS<x> specifies the measurement number.

1 add the TJ@BER measurement as part of jitter summary.

0 do not add the TJ@BER measurement as part of jitter summary.

Examples MEASUrement:MEAS4:JITTERSummary:TJBER 0 specifies that the TJ@BER measurement is not part of jitter summary.

MEASUrement:MEAS4:JITTERSummary:TJBER? might return :MEASUREMENT:MEAS4:JITTERSUMMARY:TJBER 1 indicating the TJ@BER measurement is part of jitter summary.

MEASUrement:MEAS<x>:LABel

This command sets or queries the label for the measurement. As the label can contain non 7-bit ASCII text, it is stored in Percent Encoding format. The measurement number is specified by x.

Group Measurement

Syntax MEASUrement:MEAS<x>:LABel <QString>

Arguments MEAS<x> specifies the measurement number.

<QString> is the measurement label.

Examples MEASUrement:MEAS1:LABel "Delay" sets the label to Delay.

MEASUrement:MEAS1:LABel? might return :MEASUREMENT:MEAS1:LABEL "Peak-to-Peak" indicating that the measurement 1 label is Peak-to-peak.

MEASUrement:MEAS<x>:LINESelected

This command sets or queries selected lines for the measurement. The measurement number is specified by x.

Group Measurement, IMDA

Syntax MEASUrement:MEAS<x>:LINESelected {ABCB|ACBC|BACA|XYZY|XZYZ|YXZX}

Arguments MEAS<x> specifies the measurement number.

The arguments specify the selected lines.

Examples MEASUrement:MEAS1:LINESelected ABCB sets the selected lines for measurement 1.

MEASUrement:MEAS1:LINESelected? might return :MEASUrement:MEAS1:LINESelected ABCB, indicating that selected lines for measurement 1 are ABCB.

MEASUrement:MEAS<x>:LOWREFVoltage

This command sets or queries the low reference voltage value for the 'time outside level' measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:LOWREFVoltage <NR3>
MEASUrement:MEAS<x>:LOWREFVoltage?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the low reference voltage value for the selected configuration.

Examples MEASUrement:MEAS1:LOWREFVoltage -1.30 sets the low reference voltage to -1.3 V.

MEASUrement:MEAS1:LOWREFVoltage? might return :MEASUREMENT:MEAS1:LOWREFVOLTAGE -1.0000 indicating the low ref voltage is -1.0 V.

MEASurement:MEAS<x>:MAXCycle

This command sets or queries the maximum cycle value for the DDRTERRN and DDRTERRMN measurements.

Conditions Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Group Measurement

Syntax

```
MEASurement:MEAS<x>:MAXCycle <NR1>
MEASurement:MEAS<x>:MAXCycle?
```

Arguments

MEAS<x> specifies the measurement number.

<NR1> is the maximum cycle range limit value in the range or 2 to 50.

Examples

MEASurement:MEAS2:MAXCycle 45 sets the maximum cycle value for measurement 2 to 45.

MEASurement:MEAS1:MAXCycle? might return MEASurement:MEAS1:MAXCycle 17, indicating the maximum cycle value is set to 17 for measurement 1.

MEASUrement:MEAS<x>:MAXVoltage

Sets or queries the maximum input voltage.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:MAXVoltage <NR2>
MEASUrement:MEAS<x>:MAXVoltage?
```

Arguments

MEAS<x> is the measurement number.

<NR2> is the maximum input voltage, in the range of -61 V to +61 V.

Examples

MEASUrement:MEAS2:MAXVoltage 40 sets the maximum input voltage of measurement 2 to +40 volts.

MEASUrement:MEAS4:MAXVoltage? might return 2.5000, indicating that the maximum input voltage of measurement 4 is +2.5 volts.

MEASUrement:MEAS<x>:MEASRange:GLOBal

This command sets or queries the range settings global flag for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:MEASRange:GLOBal {OFF|ON|0|1}
MEASUrement:MEAS<x>:MEASRange:GLOBal?
```

Arguments MEAS<x> specifies the measurement number.

OFF specifies that range settings can be set independently for each individual measurement.

ON applies global measurement range settings to all the measurements' range settings.

0 specifies that range settings can be set independently for each individual measurement.

1 applies global measurement range settings to all the measurements' range settings.

Examples MEASUrement:MEAS1:MEASRange:GLOBal 1 applies global measurement range settings to all the measurements' range settings.

MEASUrement:MEAS1:MEASRange:GLOBal? might return :MEASUREMENT:MEAS1:MEASRANGE:GLOBAL 0 indicating that range settings can be set independently for each individual measurement.

MEASUrement:MEAS<x>:MEASRange:MAX

This command sets or queries the range maximum value for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:MEASRange:MAX <NR3>
MEASUrement:MEAS<x>:MEASRange:MAX?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the maximum measurement range limit value.

Examples MEASUrement:MEAS1:MEASRange:MAX 2.50 sets the maximum range to 2.5 V.

MEASUrement:MEAS1:MEASRange:MAX? might return :MEASUREMENT:MEAS1:MEASRANGE:MAX 1.0000 indicating the maximum range is 1.0 V.

MEASUrement:MEAS<x>:MEASRange:MIN

This command sets or queries the range minimum value for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:MEASRange:MIN <NR3>
MEASUrement:MEAS<x>:MEASRange:MIN?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the minimum measurement range limit value.

Examples

`MEASUrement:MEAS1:MEASRange:MIN -1.0` sets the minimum measurement range to -1.0 V.

`MEASUrement:MEAS1:MEASRange:MIN?` might return `:MEASUREMENT:MEAS1:MEASRANGE:MIN 0.0E+0` indicating the minimum range is 0.0 V.

MEASUrement:MEAS<x>:MEASRange:STATE

This command sets or queries the range state for the measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:MEASRange:STATE {OFF|ON|0|1}
MEASUrement:MEAS<x>:MEASRange:STATE?
```

Arguments

`MEAS<x>` specifies the measurement number.

`OFF` turns off the measurement range limits.

`ON` turns on the measurement range limits.

`0` turns off the measurement range limits.

`1` turns on the measurement range limits.

Examples

`MEASUrement:MEAS1:MEASRange:STATE 0` turns off the measurement range limits.

MEASUrement:MEAS1:MEASRange:STATE? might return :MEASUREMENT:MEAS1:MEASRANGE:STATE 1 indicating the measurement range limits are on.

MEASurement:MEAS<x>:MINCycle

This command sets or queries the minimum cycle value for the DDRTERRN and DDRTERRMN measurements.

Conditions Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Group Measurement

Syntax

```
MEASurement:MEAS<x>:MINCycle <NR1>
MEASurement:MEAS<x>:MINCycle?
```

Arguments MEAS<x> specifies the measurement number.

<NR1> is the minimum cycle range limit value in the range or 2 to 50.

Examples MEASurement:MEAS2:MINCycle 20 sets the minimum cycle value for measurement 2 to 20.

MEASurement:MEAS1:MINCycle? might return MEASurement:MEAS1:MINCycle 17, indicating the minimum cycle value is set to 17 for measurement 1.

MEASUrement:MEAS<x>:ODDEVen

This command sets or queries harmonics oddeven.

Group Measurement, IMDA

Syntax

```
MEASUrement:MEAS<x>:ODDEVen {ALL|EVEN|ODD}
MEASUrement:MEAS<x>:ODDEVen?
```

Arguments MEAS<x> specifies the measurement number.

ALL displays all the results.

EVEN displays the even results.

ODD displays the odd results.

Examples MEASurement:MEAS1:ODDEVen ALL displays all the harmonics results for measurement 1.

MEASurement:MEAS1:ODDEVen? might return :MEASurement:MEAS1:ODDEVen EVEN, indicating the even results are displayed.

MEASUrement:MEAS<x>:OUTFILTers:LOWPass:FREQ

This command sets or queries the measurement output low pass filter frequency cutoff frequency on the scope.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:FREQ <NR2>
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:FREQ?
```

Arguments MEAS<x> specifies the measurement number.

<NR2> specifies the values for the output filter cutoff frequency.

Examples MEASurement:MEAS1:OUTFILTers:LOWPass:FREQ 2.000E+6 sets the value for the output filter cutoff frequency for measurement 1 as 2.000E+6.

MEASurement:MEAS1:OUTFILTers:LOWPass:FREQ? might return :MEASurement:MEAS1:OUTFILTers:LOWPass:FREQ 2.000E+6, indicating the value for the output filter cutoff frequency for is 2.000E+6.

MEASUrement:MEAS<x>:OUTFILTers:LOWPass:SPEC

This command sets or queries the measurement output low pass filter specification filter order on the scope. Measurements are specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:SPEC
{NONE|FIRST|SECOND|THIRD}
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:SPEC?
```

Arguments

MEAS<x> specifies the measurement number.

NONE specifies the filter order none.

FIRST specifies the filter order first.

SECOND specifies the filter order second.

THIRD specifies the filter order third.

Examples

MEASUrement:MEAS1:OUTFILTers:LOWPass:SPEC SECOND sets the filter order of measurement 1 to second.

MEASUrement:MEAS1:OUTFILTers:LOWPass:SPEC? might return :MEASUrement:MEAS1:OUTFILTers:LOWPass:SPEC SECOND, indicating the filter order is set to second.

MEASUrement:MEAS<x>:OUTPUT<n>VOLTage

Sets or queries the voltage level for the specified output voltage source n.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:OUTPUT<n>VOLTage <NR2>
MEASUrement:MEAS<x>:OUTPUT<n>VOLTage?
```

Arguments

MEAS<x> specifies the measurement number.

<NR2> is the output voltage for source n, in the range of -61 V to +61 V.

Examples

MEASUrement:MEAS1:OUTPUT3VOLTage –24 sets the output voltage for source 3 of measurement 1 to –24 volts.

MEASUrement:MEAS1:OUTPUT3VOLTage? might return 2.500, indicating that the output voltage for source 3 of measurement 1 is 2.5 volts.

MEASUrement:MEAS<x>:PASSFAILENabled

This command returns or sets the pass/fail test enable status. If enabled, this will turn on pass fail testing for the specified measurement. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILENabled <NR1>
MEASUrement:MEAS<x>:PASSFAILENabled?
```

Related Commands ACTONEVent:ENable

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE

ACTONEVent:MEASUrement:ACTION:SRQ:STATE

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE

Arguments <NR1> enables or disables pass fail testing for the specified measurement. A value of 1 enables and a value of 0 disables.

Examples `MEASUrement:MEAS1:PASSFAILENabled 1` enables pass fail testing for the specified measurement.

`MEASUrement:MEAS1:PASSFAILENabled?` might return `:MEASUrement:MEAS1:PASSFAILENabled 0` indicating that pass fail testing for the specified measurement is disabled.

MEASUrement:MEAS<x>:PASSFAILHIGHlimit

This command returns or sets the high limit for a measurement test. Used as the test value when the "fail when" criteria is set to "less than" or "greater than". Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILHIGHlimit <NR2>
MEASUrement:MEAS<x>:PASSFAILHIGHlimit?
```

Related Commands ACTONEVent:ENable

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE

ACTONEVent:MEASUrement:ACTION:SRQ:STATE

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE

Arguments <NR2> sets the high limit for a measurement test. The high limit is a number which a measurement result will be tested against.

Examples MEASUrement:MEAS1:PASSFAILHIGHlimit 710 specifies 710 as the high limit for measurement 1.

MEASUrement:MEAS1:PASSFAILHIGHlimit? might return :MEASUrement:MEAS1:PASSFAILHIGHlimit 710 indicating that 710 is the high limit for measurement 1.

MEASUrement:MEAS<x>:PASSFAILLIMit

This command returns or sets the limit for a measurement test. Used as the test value when the "fail when" criteria is set to "less than" or "greater than". Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:PASSFAILLIMit <NR2>
MEASUrement:MEAS<x>:PASSFAILLIMit?
```

Related Commands ACTONEVent:ENable

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE

ACTONEVent:MEASUrement:ACTION:SRQ:STATE

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE

Arguments <NR2> sets the limit for a measurement test. The limit is a number which a measurement result will be tested against.

Examples MEASUrement:MEAS1:PASSFAILLIMit 710 specifies 710 as the limit for measurement 1.

MEASUrement:MEAS1:PASSFAILLIMit? might return :MEASUrement:MEAS1:PASSFAILLIMit 710 indicating that 710 is the limit for measurement 1.

MEASUrement:MEAS<x>:PASSFAILLOWlimit

This command returns or sets the low limit for a measurement test. Used as the test value when the "fail when" criteria is set to "less than" or "greater than". Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILLOWlimit <NR2>
MEASUrement:MEAS<x>:PASSFAILLOWlimit?
```

Related Commands ACTONEVent:ENable

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE

ACTONEVent:MEASUrement:ACTION:SRQ:STATE

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE

Arguments <NR2> sets the low limit for a measurement test. The limit is a number which a measurement result will be tested against.

Examples `MEASUrement:MEAS1:PASSFAILLOWlimit 690` specifies 690 as the limit for measurement 1.

`MEASUrement:MEAS1:PASSFAILLOWlimit?` might return `:MEASUrement:MEAS1:PASSFAILLOWlimit 690` indicating that 690 is the low limit for measurement 1.

MEASUrement:MEAS<x>:PASSFAILMARgin

This command returns or sets the allowed margin for limit comparisons for all pass/fail checks. This is given as a percentage with a default value of 0.05 representing 5%. Measurements are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILMARgin <NR2>
MEASUrement:MEAS<x>:PASSFAILMARgin?
```

Related Commands

ACTONEVent:ENable

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE

ACTONEVent:MEASUrement:ACTION:SRQ:STATE

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE

Arguments

<NR2> sets the allowed margin for limit comparisons for all pass/fail checks. The margin as a percentage of the limit.

Examples

`MEASUrement:MEAS1:PASSFAILMARgin 0.10` specifies 10% as the allowed margin for limit comparisons for all pass/fail checks.

`MEASUrement:MEAS1:PASSFAILMARgin?` might return `:MEASUrement:MEAS1:PASSFAILMARgin 0.10` indicating that 10% is the allowed margin for limit comparisons for all pass/fail checks on measurement 1.

MEASUrement:MEAS<x>:PASSFAILWHEN

This command sets or returns the condition on which a measurement test fails. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PASSFAILWHEN {LESSthan| GREATERthan|
Equals| NOTEQuals| INSIDErange| OUTSIDErange}
MEASUrement:MEAS<x>:PASSFAILWHEN?
```

Related Commands

ACTONEVent:ENable

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE

ACTONEVent:MEASUrement:ACTION:SRQ:STATE

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE

Arguments

`LESSthan` sets the condition for measurement test failure as less than the given limit. This is the default value.

`GREATERthan` sets the condition for measurement test failure as greater than the given limit.

Equals sets the condition for measurement test failure as equals the given limit.

NOTEQuals sets the condition for measurement test failure as not equal to the given limit.

INSIDErange sets the condition for measurement test failure as inside the limit range.

OUTSIDErange sets the condition for measurement test failure as outside the limit range.

Examples

MEASUrement:MEAS1:PASSFAILWHEN LESSthan specifies the condition for measurement 1 test failure as less that the given limit.

MEASUrement:MEAS1:PASSFAILWHEN? might return :MEASUrement:MEAS1:PASSFAILWHEN LESSthan indicating that the condition for measurement 1 test failure as less that the given limit.

MEASUrement:MEAS<x>:PATTERNDETECTION

This command sets or queries the pattern detection type for the measurement. Measurements are specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:PATTERNDETECTION {AUTO|MANUAL}
MEASUrement:MEAS<x>:PATTERNDETECTION?
```

Arguments

MEAS<x> specifies the measurement number.

AUTO automatically detects the pattern.

MANUAL requires manually detecting the pattern.

Examples

MEASUrement:MEAS1:PATTERNDETECTION MANUAL specifies manually detecting the pattern.

MEASUrement:MEAS1:PATTERNDETECTION? might return :MEASUREMENT:MEAS1:PATTERNDETECTION AUTO indicating patterns are automatically detected.

MEASUrement:MEAS<x>:PATTERNLENgth

This command sets or queries the pattern length for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:PATTERNLENgth <NR3>
MEASUrement:MEAS<x>:PATTERNLENgth?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the pattern length.

Examples `MEASUrement:MEAS1:PATTERNLENgth 3.0` sets the pattern length to 3.0.

`MEASUrement:MEAS1:PATTERNLENgth?` might return `:MEASUREMENT:MEAS1:PATTERNLENGTH 2.0000` indicating the patternlength is 2.0.

MEASUrement:MEAS<x>:PATTERNTYPe

This command sets or queries the pattern type for the measurement. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:PATTERNTYPe {REPeating|ARBitrary}
MEASUrement:MEAS<x>:PATTERNTYPe?
```

Arguments `MEAS<x>` specifies the measurement number.

`REPeating` specifies a repeating pattern.

`ARBitrary` specifies an arbitrary pattern.

Examples `MEASUrement:MEAS1:PATTERNTYPe ARBITRARY` specifies an arbitrary pattern.

`MEASUrement:MEAS1:PATTERNTYPe?` might return `:MEASUREMENT:MEAS1:PATTERNTYPE REPEATING` indicating a repeating pattern.

MEASUrement:MEAS<x>:PERFREQ:EDGE

This command sets or queries the edge type of a Period/Frequency measurement. The measurement number is specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:PERFREQ:EDGE {FIRST|RISE|FALL}`

Arguments `MEAS<x>` specifies the measurement number.

`FIRST` computes the measurement between Rising edges if the first edge is Rising. Computes the measurement between Falling edges if the first edge is Falling.

`RISE` computes the measurement between Rising edges.

`FALL` computes the measurement between Falling edges.

Examples `MEASUrement:MEAS1:PERFREQ:EDGE RISE` specifies computing the measurement between Rising edges.

`measurement:meas1:perfreq:edge?` might return `:MEASUREMENT:MEAS1:PERFREQ:EDGE FIRST` if the measurement is computed between edges of the first type edge found.

MEASUrement:MEAS<x>:POLarity

This command sets or queries the polarity for the measurement when the measurement type is burst width. Measurements are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:POLarity {NORMal|INVerted}
MEASUrement:MEAS<x>:POLarity?
```

Arguments `MEAS<x>` specifies the measurement number.

`NORMal` specifies normal polarity.

`INVerted` specifies inverted polarity.

Examples `MEASUrement:MEAS1:POLarity INVERTED` specifies inverted polarity.

`MEASUrement:MEAS1:POLarity?` might return `:MEASUREMENT:MEAS1:POLARITY NORMAL` indicating normal polarity.

MEASUrement:MEAS<x>:POPUlation:GLOBal

This command sets or queries the population settings global flag. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:POPUlation:GLOBal {OFF|ON|0|1}
MEASUrement:MEAS<x>:POPUlation:GLOBal?
```

Arguments

MEAS<x> specifies the measurement number.

OFF specifies that population settings can be changed independently for each individual measurement.

ON applies the global population settings to all the measurements' population settings.

0 specifies that population settings can be changed independently for each individual measurement.

1 applies the global population settings to all the measurements' population settings.

Examples

MEASUrement:MEAS1:POPUlation:GLOBal 1 applies the global population settings to all the measurements' population settings.

MEASUrement:MEAS1:POPUlation:GLOBal? might return :MEASUREMENT:MEAS1:POPULATION:GLOBAL 0 indicating that population settings can be changed independently for each individual measurement.

MEASUrement:MEAS<x>:POPUlation:LIMIT:STATE

This command sets or queries the population limit state for the measurement. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:POPUlation:LIMIT:STATE {OFF|ON|0|1}
MEASUrement:MEAS<x>:POPUlation:LIMIT:STATE?
```

Arguments

MEAS<x> specifies the measurement number.

OFF turns off the population limit.

ON turns on the population limit.

0 turns off the population limit.

1 turns on the population limit.

Examples MEASUrement:MEAS1:POPUlation:LIMIT:STATE 0 turns off the population limit.

MEASUrement:MEAS1:POPUlation:LIMIT:STATE? might return :MEASUREMENT:MEAS1:POPULATION:LIMIT:STATE 1 indicating the population limit is on.

MEASUrement:MEAS<x>:POPUlation:LIMIT:VALue

This command sets or queries the population limit value for the measurement. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:POPUlation:LIMIT:VALue <NR3>
MEASUrement:MEAS<x>:POPUlation:LIMIT:VALue?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> the current limit value.

Examples MEASUrement:MEAS1:POPUlation:LIMIT:VALue 2000 sets the population limit to 2000.

MEASUrement:MEAS1:POPUlation:LIMIT:VALue? might return :MEASUREMENT:MEAS1:POPULATION:LIMIT:VALUE 1000 indicating the limit is set to 1000.

MEASUrement:MEAS<x>:POWERFACtor

This command sets or queries the power factor value for IEC-Class C type harmonics standard. The measurement number is specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:POWERFACtor <NR2>
MEASUrement:MEAS<x>:POWERFACtor?
```

Arguments MEAS<x> specifies the measurement number.

<NR2> specifies the power factor value for class C harmonics standard.

Examples MEASUrement:MEAS1:POWERFACtor sets the .

MEASUrement:MEAS1:POWERFACtor? might return :MEASUREMENT:MEAS1:POWERFACtor indicating the .

MEASUrement:MEAS<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

NOTE. *This command affects the results of rise and fall measurements*

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:REFLevels:ABSolute:FALLHigh <NR3>
MEASUrement:MEAS<x>:REFLevels:ABSolute:FALLHigh?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the high reference level in volts. The default is 0.0 V.

Examples MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH 1.71 sets the high reference level to 1.71 V.

MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH? might return :MEASUREMENT:MEAS1:REFLEVELS:ABSOLUTE:FALLHIGH 1.7100E+00, indicating that the absolute high reference level is set to 1.71 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

NOTE. *This command affects the results of rise and fall measurements.*

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLLow?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the low reference level in volts. The default is 0.0 V.

Examples MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW 0.0 V sets the low reference level to 0.0 V.

MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLLOW 0.0000E+00, indicating that the absolute low reference level is set to 0.0 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute. Measurements are specified by x.

NOTE. *This command affects the results of period, frequency, delay, and all cyclic measurements.*

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:FALLMid?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the mid reference level in volts. The default is 0.0 V.

Examples MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID 0.5 sets the mid reference level for the delay waveform to 0.5 V.

MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:FALLMID 0.5000E+00, indicating that the absolute mid reference level is set to 0.5 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSTeresis <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:HYSTeresis?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the hysteresis value used for autoset.

Examples `MEASUrement:MEAS1:REFLevels1:ABSolute:HYSTeresis 25.0E-3` sets the hysteresis to 25 mV.

`MEASUrement:MEAS1:REFLevels1:ABSolute:HYSTeresis?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:HYSTERESIS 30.0000E-3` indicating the hysteresis is set to 30 mV.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEHigh?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the high reference level, and is the zero percent level when the measurement's Ref level method is set to Absolute.

Examples `MEASUrement:MEAS1:REFLevels1:ABSolute:RISEHigh 1.50` sets the high reference level to 1.5 V.

MEASUrement:MEAS1:REFLevels1:ABSolute:RISEHigh? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEHIGH 1.0000 indicating the high reference level is set to 1.0 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISELow <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISELow?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the low reference level, and is the zero percent level when the measurement's Ref level method is set to Absolute.

Examples MEASUrement:MEAS1:REFLevels1:ABSolute:RISELow -1.50 sets the low reference level to -1.5 V.

MEASUrement:MEAS1:REFLevels1:ABSolute:RISELow? might return :MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISELOW -1.0000 indicating the low reference level is -1.0 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMid <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:RISEMid?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the mid reference level (where 50% is equal to MID) used to calculate the mid reference level when the measurement's Ref level method is set to Absolute.

Examples `MEASUrement:MEAS1:REFLevels1:ABSolute:RISEMid 30.0E-3` sets the mid reference level to 30 mV.

`MEASUrement:MEAS1:REFLevels1:ABSolute:RISEMid?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:RISEMID 0.0E+0` indicating the mid reference level is 0.0 V.

MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:TYPE

This command sets or queries the reference level type for the measurement. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:TYPE {SAME|UNIQue}
MEASUrement:MEAS<x>:REFLevels<x>:ABSolute:TYPE?
```

Arguments `MEAS<x>` specifies the measurement number.

`SAME` specifies that the absolute levels are set the same.

`UNIQue` specifies that the absolute levels can be set independently.

Examples `MEASUrement:MEAS1:REFLevels1:ABSolute:TYPE UNIQUE` specifies that the absolute levels can be set independently.

`MEASUrement:MEAS1:REFLevels1:ABSolute:TYPE?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:ABSOLUTE:TYPE SAME` indicating the absolute levels are set the same.

MEASUrement:MEAS<x>:REFLevels<x>:BASETop

This command sets or queries the method used to calculate the TOP and BASE used to calculate reference levels for the measurement. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:REFLevels<x>:BASETop
{AUTO|MINMax|MEANhistogram| MODEhistogram|EYEhistogram}
MEASUrement:MEAS<x>:REFLevels<x>:BASETop?
```

Arguments

MEAS<x> specifies the measurement number.

AUTO automatically chooses a reference level method.

MINMax specifies that reference levels are relative to the measurement MIN and MAX.

MEANhistogram specifies that reference levels are relative to the histogram mean BASE and TOP.

MODEhistogram specifies that reference levels are relative to the histogram mode BASE and TOP.

EYEhistogram specifies that reverence levels are relative to the eye histogram BASE and TOP.

Examples

MEASUrement:MEAS1:REFLevels1:BASETop MINMAX specifies that reference levels are relative to the measurement MIN and MAX.

MEASUrement:MEAS1:REFLevels1:BASETop? might return :MEASUREMENT:MEAS1:REFLEVELS1:BASETOP AUTO indicating the instrument automatically chooses a reference level method.

MEASUrement:MEAS<x>:REFLevels<x>:METHod

This command sets or queries the method used to calculate reference levels for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:METHod {PERCent|ABSolute}
MEASUrement:MEAS<x>:REFLevels<x>:METHod?
```

Arguments

MEAS<x> specifies the measurement number.

PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:MEAS<x>:REFLevel:PERCent commands.

ABSolute specifies that the reference levels are set explicitly using the MEASUrement:MEAS<x>:REFLevel:ABSolute commands. This method is useful when precise values are required.

Examples

MEASUREMENT:MEAS1:REFLEVELS1:METHOD ABSOLUTE specifies that explicit user-defined values are used for the reference levels.

MEASUREMENT:MEAS1:REFLEVELS1:METHOD? might return :MEASUREMENT:MEAS1:REFLEVELS1:METHOD PERCENT, indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLHigh <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLHigh?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to HIGH) used to calculate the high reference level.

Examples MEASUrement:MEAS1:REFLevels1:PERCent:FALLHigh 95.0 sets the high reference level of the falling edge to 95%.

MEASUrement:MEAS1:REFLevels1:PERCent:FALLHigh? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLHIGH 90.0000 indicating the high reference level is set to 90%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLLow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLLow <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLLow?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 100% is equal to HIGH) used to calculate the mid reference level.

Examples MEASUrement:MEAS1:REFLevels1:PERCent:FALLLow 5.0 sets the low reference level of the falling edge is 5%.

MEASUrement:MEAS1:REFLevels1:PERCent:FALLLow? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLLOW 10.0000 indicating the low reference level of the falling edge is 10%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLMid <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:FALLMid?
```

Arguments MEAS<x> specifies the measurement number.

<NR3> is the percentage (where 50% is equal to MID) used to calculate the mid reference level.

Examples MEASUrement:MEAS1:REFLevels1:PERCent:FALLMid 50.0 sets the mid reference level of the falling edge to 50%.

MEASUrement:MEAS1:REFLevels1:PERCent:FALLMid? might return :MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:FALLMID 50.0000 indicating the mid reference level of the falling edge is set to 50%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:HYSTeresis?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the hysteresis value used for the autoset.

Examples `MEASUrement:MEAS1:REFLevels1:PERCent:HYSTeresis 5.0` sets the hysteresis level to 5%.

`MEASUrement:MEAS1:REFLevels1:PERCent:HYSTeresis?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:HYSTERESIS 5.0000` indicating the hysteresis is 5.0%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEHigh?
```

Arguments `MEAS<x>` specifies the measurement number.

`<NR3>` is the percentage (where 100% is equal to TOP) used to calculate the high reference level when the measurement's Ref level method is set to Percent.

Examples `MEASUrement:MEAS1:REFLevels1:PERCent:RISEHigh 95.0` sets the high reference level of the rising edge is set to 95%.

`MEASUrement:MEAS1:REFLevels1:PERCent:RISEHigh?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEHIGH 90.0000` indicating the high reference level of the rising edge is set to 90%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISELow?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` is the percentage (where 100% is equal to TOP) used to calculate the mid reference level when the measurement's Ref level method is set to Percent.

Examples

`MEASUrement:MEAS1:REFLevels1:PERCent:RISELow 5.0` sets the low reference level is set to 5.0%.

`MEASUrement:MEAS1:REFLevels1:PERCent:RISELow?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISELOW 10.0000` indicating the low reference level is set to 10.0%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid <NR3>
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:RISEMid?
```

Arguments

`MEAS<x>` specifies the measurement number.

`<NR3>` the percentage (where 50% is equal to MID) used to calculate the mid reference level when the measurement Ref level method is set to Percent.

Examples

`MEASUrement:MEAS1:REFLevels1:PERCent:RISEMid 50.0000` sets the mid reference level of the rising edge is set to 50.0%.

`MEASUrement:MEAS1:REFLevels1:PERCent:RISEMid?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:RISEMID 50.0000` indicating the mid reference level of the rising edge is set to 50.0%.

MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE

This command specifies or queries the reference level percent type for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE
{TENNinety|TWENtyeighty|CUSTom}
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:TYPE?
```

Arguments

`MEAS<x>` specifies the measurement number.

`TENNinety` sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

`TWENtyeighty` sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

`CUSTom` allows setting other reference level percents.

Examples

`MEASUrement:MEAS1:REFLevels1:PERCent:TYPE TWENtyeighty` sets the reference levels percent to 20%, 50% and 80%.

`MEASUrement:MEAS1:REFLevels1:PERCent:TYPE?` might return `:MEASUREMENT:MEAS1:REFLEVELS1:PERCENT:TYPE TENNINETY` indicating the reference levels percent type is 10%, 50% and 90% respectively

MEASUrement:MEAS<x>:REFMode

This command sets or queries the reference level mode for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFMode {AUTO|MANual}
MEASUrement:MEAS<x>:REFMode?
```

Arguments

MEAS<x> specifies the measurement number.

AUTO sets the reference level for the measurement automatically.

MANual allows the user to set the reference level for the measurement.

Examples

MEASUrement:MEAS1:REFMode Manual allows the user to set the reference level for the measurement.

MEASUrement:MEAS1:REFMode? might return :MEASUREMENT:MEAS1:REFMODE AUTO indicating the reference levels for the measurement are set automatically.

MEASUrement:MEAS<x>:REFVoltage

This command sets or queries the reference voltage value for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:REFVoltage <NR3>
MEASUrement:MEAS<x>:REFVoltage?
```

Arguments

<NR3> is the reference voltage value for the selected configuration.

Examples

MEASUrement:MEAS1:REFVoltage 30.E-3 sets the reference voltage value for the measurement to 30 mV.

MEASUrement:MEAS1:REFVoltage? might return :MEASUREMENT:MEAS1:REFVOLTAGE 0.0E+0 indicating the reference voltage value for the measurement is 0.0 V.

MEASUrement:MEAS<x>:REFVOLTAGE<n>Val

Sets or queries the reference voltage per rail of Overshoot and Undershoot measurements.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax
```
MEASUrement:MEAS<x>:REFVOLTAGE<n>Val <NR2>
MEASUrement:MEAS<x>:REFVOLTAGE<n>Val?
```

Arguments MEAS<x> is the measurement number.

<n> is the rail number.

<NR2> is the reference voltage, in the range of -500 V to +500 V.

Examples MEASUrement:MEAS1:REFVOLTAGE3Val −125 sets the reference voltage for rail 3 of measurement 1 to –125 volts.

MEASUrement:MEAS3:REFVOLTAGE1val? might return 55.200, indicating that the reference voltage of rail 1 on measurement 3 is set to +55.2 volts.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MAXimum? (Query Only)

This query-only command returns the maximum value for all accumulated measurement acquisitions of the specified measurement. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MAXimum?
```

Returns The maximum value for all accumulated measurement acquisitions of the specified measurement.

Examples MEASUrement:MEAS1:RESUlts:ALLAcqs:MAXimum? might return :MEASUREMENT:MEAS1:RESULTS:ALLACQS:MAXIMUM 2.420 indicating the maximum measurement value is 2.420 V.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MEAN? (Query Only)

This query-only command returns the mean value for all accumulated measurement acquisitions for measurement <x>.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MEAN?`

Returns The mean value for all accumulated measurement acquisitions for measurement <x>.

Examples `MEASUrement:MEAS1:RESUlts:ALLAcqs:MEAN?` might return `:MEASUREMENT:MEAS1:RESULTS:ALLACQS:MEAN 2.2807617754647` indicating the measurement mean is 2.28 V.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MINimum? (Query Only)

This query-only command returns the minimum value for all accumulated measurement acquisitions for measurement <x>.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:ALLAcqs:MINimum?`

Returns The minimum value for all accumulated measurement acquisitions for measurement <x>.

Examples `MEASUrement:MEAS1:RESUlts:ALLAcqs:MINimum?` might return `:MEASUREMENT:MEAS1:RESULTS:ALLACQS:MINIMUM 2.220` indicating the measurement minimum value is 2.220 V.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:PK2PK? (Query Only)

This query-only command returns the peak-to-peak value for all accumulated measurement acquisitions for measurement <x>.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:ALLAcqs:PK2PK?`

Returns The peak-to-peak value for all accumulated measurement acquisitions for measurement <x>.

Examples `MEASUrement:MEAS1:RESUlts:ALLAcqs:PK2PK?` might return `:MEASUREMENT:MEAS1:RESULTS:ALLACQS:PK2PK 200.0E-3` indicating the measurement peak-to-peak value is 200 mV.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:POPUlation? (Query Only)

This query-only command returns the population measurement value for measurement <x>.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:ALLAcqs:POPUlation?`

Returns The population measurement value for measurement <x>.

Examples `MEASUrement:MEAS1:RESUlts:ALLAcqs:POPUlation?` might return `:MEASUREMENT:MEAS1:RESULTS:ALLACQS:POPULATION 29825` indicating the measurement population is 29825 measurements.

MEASUrement:MEAS<x>:RESUlts:ALLAcqs:STDDev? (Query Only)

This query-only command returns the standard deviation for all accumulated measurement acquisitions for measurement <x>.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:ALLAcqs:STDDev?`

Returns The standard deviation for all accumulated measurement acquisitions for measurement <x>.

Examples `MEASUrement:MEAS1:RESUlts:ALLAcqs:STDDev?` might return `:MEASUREMENT:MEAS1:RESULTS:ALLACQS:STDDEV 23.5741246494459E-3` indicating the measurement standard deviation is 23.57 mV.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:MAXimum? (Query Only)

This query-only command returns the maximum value found for the specified measurement since the last statistical reset. The measurement number is specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:CURRentacq:MAXimum?`

Returns The maximum value found for the specified measurement since the last statistical reset.

Examples `MEASUrement:MEAS1:RESUlts:CURRentacq:MAXimum?` might return `:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MAXIMUM 2.280` indicating the current measurement maximum value is 2.28 V.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:MEAN? (Query Only)

This query-only command returns the mean value for the measurement for the current acquisition.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:CURRentacq:MEAN?`

Arguments MEAS<x> is the measurement number for which to return a value. This is the equivalent of the number shown in the measurement badge on the UI.

Returns The mean value accumulated for the specified measurement since the last statistical reset.

Examples `MEASUrement:MEAS3:RESUlts:CURRentacq:MEAN?` might return `:MEASUREMENT:MEAS3:RESULTS:CURRENTACQ:MEAN 2.30` indicating the mean of measurement 3 in the current acquisition is 2.3 V.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:MINimum? (Query Only)

This query-only command returns the minimum value found for the specified measurement since the last statistical reset. The measurement number is specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:CURRentacq:MINimum?`

Returns The minimum value found for the specified measurement since the last statistical reset.

Examples `MEASUrement:MEAS1:RESUlts:CURRentacq:MINimum?` might return `:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:MINIMUM 2.260` indicating the minimum value of the measurement in the current acquisition is 2.26 V.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:PK2PK? (Query Only)

This query-only command returns the peak-to-peak value for the specified measurement for the current acquisition. The measurement number is specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:CURRentacq:PK2PK?`

Returns The peak-to-peak value for the specified measurement.

Examples `MEASUrement:MEAS1:RESUlts:CURRentacq:PK2PK?` might return `:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:PK2PK 0.0E+0` indicating the peak-to-peak value for the specified measurement for the current acquisition is 0.0 V.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:POPUlation? (Query Only)

This query-only command returns the population for the specified measurement for the current acquisition. The measurement number is specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:CURRentacq:POPUlation?`

Returns The population measurement value for the specified measurement.

Examples `MEASUrement:MEAS<x>:RESUlts:CURRentacq:POPUlation?` might return `:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:POPULATION 1` indicating the population for the current measurement is 1 measurement.

MEASUrement:MEAS<x>:RESUlts:CURRentacq:STDDev? (Query Only)

This query-only command returns the standard deviation for the specified measurement for all acquisitions accumulated since statistics were last reset. The measurement number is specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:RESUlts:CURRentacq:STDDev?`

Returns The standard deviation of values accumulated for the specified measurement since the last statistical reset.

Examples `MEASUrement:MEAS<x>:RESUlts:CURRentacq:STDDev?` might return `:MEASUREMENT:MEAS1:RESULTS:CURRENTACQ:STDDEV 0.0E+0` indicating the standard deviation for the measurement is 0.0 V.

MEASUrement:MEAS<x>:RIPPLEFREQ<n>Val

Sets or queries the ripple frequency per specified rail for Ripple measurement.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax
```
MEASUrement:MEAS<x>:RIPPLEFREQ<n>Val <NR3>
MEASUrement:MEAS<x>:RIPPLEFREQ<n>Val?
```

Arguments MEAS<x> is the measurement number.

<n> is the rail number.

<NR3> is the frequency, in the range of 1 Hz to 4 GHz.

Examples MEASUrement:MEAS1:RIPPLEFREQ3Val 5.2e6 sets the ripple frequency for rail 3 of measurement 1 to 5.2 MHz.

MEASUrement:MEAS3:RIPPLEFREQ1val? might return 1.0000E+6, indicating that the ripple frequency of rail 1 on measurement 3 is set to 1 MHz.

MEASUrement:MEAS<x>:SIGNALFREQUEncy

This command sets or queries line frequency for Harmonics and Ripple line frequency. The measurement number is specified by x.

Group Measurement, IMDA

Syntax
```
MEASUrement:MEAS<x>:SIGNALFREQUEncy
{AUTO|FIFTYHZ|SIXTYHZ|FOURHUNDREDHZ|CUSTOM}
MEASUrement:MEAS<x>:SIGNALFREQUEncy?
```

Arguments MEAS<x> specifies the measurement number.

AUTO specifies the line frequency as auto. Only available for Harmonics line frequency.

FIFTYHZ specifies the line frequency as 50 Hz.

SIXTYHZ specifies the line frequency as 60 Hz.

FOURHUNDREDHZ specifies the line frequency as 400 Hz. .

CUSTOM specifies the line frequency as custom.

Examples MEASUrement:MEAS1:SIGNALFREQUEncy FIFTYHZ specifies the line frequency as 50 Hz for measurement 1.

MEASUrement:MEAS1:SIGNALFREQUEncy? might return :MEASUREMENT:MEAS1:SIGNALFREQUEncy FIFTYHZ, indicating the line frequency is 50 Hz.

MEASUrement:MEAS<x>:SIGNALType

This command sets or queries the signal type of source 1 for the measurement. The measurement number is specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:SIGNALType {CLOCK|DATA|AUTO}
MEASUrement:MEAS<x>:SIGNALType?
```

Arguments

MEAS<x> specifies the measurement number.

CLOCK specifies a clock signal type.

DATA specifies a data signal type.

AUTO automatically selects the signal type.

Examples

MEASUrement:MEAS1:SIGNALType CLOCK specifies a clock signal type.

MEASUrement:MEAS1:SIGNALType? might return :MEASUREMENT:MEAS1:SIGNALTYPE AUTO indicating the instrument automatically selects the signal type.

MEASUrement:MEAS<x>:SOUrce<x>

This command sets or queries the measurement source. The measurement number and source are specified by x.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:SOUrce<x>
MEASUrement:MEAS<x>:SOUrce<x>?
```

Arguments

MEAS<x> specifies the measurement number.

SOUrce<x> specifies the source number.

Examples

MEASUrement:MEAS1:SOUrce CH1 sets the source for measurement 1 to channel 1.

MEASUrement:MEAS1:SOUrce? might return :MEASUrement:MEAS1:SOUrce CH1 indicating the measurement source is channel 1.

MEASUrement:MEAS<x>:SSC:NOMinalfreq

This command sets or queries the user-defined frequency for the measurement when the measurement type is SSC. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:SSC:NOMinalfreq <NR3>
MEASUrement:MEAS<x>:SSC:NOMinalfreq?
```

Arguments

MEAS<x> specifies the measurement number.

<NR3> is the user-defined nominal frequency type for SSC configurations.

Examples

MEASUrement:MEAS1:SSC:NOMinalfreq 2.0E+9 sets the frequency to 2.0 GHz.

MEASUrement:MEAS1:SSC:NOMinalfreq? might return :MEASUREMENT:MEAS1:SSC:NOMINALFREQ 2.5000E+9 indicating the frequency is 2.5 GH.

MEASUrement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype

This command sets or queries the frequency detection type for the measurement when the measurement type is SSC. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype
{AUTO|MANual}
MEASUrement:MEAS<x>:SSC:NOMinalfreq:SELECTIONtype?
```

Arguments

MEAS<x> specifies the measurement number.

AUTO automatically sets the detection type.

MANual specifies the manual detection type.

Examples

MEASUrement:MEAS1:SSC:NOMinalfreq:SELECTIONtype AUTO specifies the auto-detection type.

MEASUrement:MEAS1:SSC:NOMinalfreq:SELECTIONtype? might return :MEASUREMENT:MEAS1:SSC:NOMINALFREQ:SELECTIONTYPE AUTO indicating the auto-detection type is selected.

MEASUrement:MEAS<x>:STATUS? (Query Only)

This command returns the pass fail status, if applicable, for the selected measurement. Measur are specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:STATUS?
```

Related Commands ACTONEVent:ENable

ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE

ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE

ACTONEVent:MEASUrement:ACTION:SRQ:STATE

ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE

Arguments PASS specifies that the user specified measurement limit has not been violated.

FAIL specifies that the user specified measurement limit has been violated.

Examples MEASUrement:MEAS1:STATUS? might return :MEASUrement:MEAS1:STATUS PASS indicating that the user specified measurement limit has not been violated.

MEASUrement:MEAS<x>:TCKAVG

This command sets or queries the average clock period value used in DDR measurements.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:TCKAVG <NR3>
MEASUrement:MEAS<x>:TCKAVG?
```

Related Commands MEASurement:MEAS<x>:TIMINGMode

Arguments MEAS<x> specifies the measurement number.

NR3 is a floating point number that represents the DDR average clock period in seconds.

Examples MEASurement:MEAS7:TCKAVG 2.5E-9 sets the average clock period value as 3.2 ns for measurement 7.

MEASurement:MEAS2:TCKAVG? might return MEASurement:MEAS2:TCKAVG 2.0E-9 indicating that average clock period is set to 2.0 ns for measurement 2.

MEASurement:MEAS<x>:TIMINGMode

This command sets or queries the Timing mode for the specified DDR measurement.

Group Measurement

Syntax
```
MEASurement:MEAS<x>:TIMINGMode
{EACHCLOCKCYCLE|TWOCLOCKCYCLES}
MEASurement:MEAS<x>:TIMINGMode?
```

Related Commands MEASUrement:MEAS<x>:TCKAVG

Arguments MEAS<x> specifies the measurement number.

EACHCLOCKCYCLE sets the DDR Timing Mode to use each clock cycle at a time.

TWOCLOCKCYCLES sets the DDR Timing Mode to use two cycles at a time.

Examples MEASurement:MEAS1:TIMINGMode EACHCLOCKCYCLE sets the Timing Mode to consider each clock cycle.

MEASurement:MEAS1:TIMINGMode? might return MEASurement:MEAS1:TIMINGMode EACHCLOCKCYCLE, indicating that Timing Mode is set to EACHCLOCKCYCLE.

MEASUrement:MEAS<x>:TOEdge

This command sets or queries the 'to edge' type for the measurement. The measurement number is specified by x.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:TOEdge
{SAMEas|OPPositeas|RISe|FALL|BOTH}
MEASUrement:MEAS<x>:TOEdge?
```

Arguments

MEAS<x> specifies the measurement number.

FALL specifies the falling edge of the waveform.

RISE specifies the rising edge of the waveform.

BOTH specifies both a rising and falling edge of the waveform.

SAMEas specifies that both edges of the waveform are the same.

OPPositeas specifies that the edges of the waveform are not the same.

Examples

MEASUrement:MEAS1:TOEdge FALL specifies the to edge is the falling edge of the waveform.

MEASUrement:MEAS1:TOEdge? might return :MEASUREMENT:MEAS1:TOEDGE SAMEAS indicating that both edges of the waveform are the same.

MEASUrement:MEAS<x>:TOEDGESEARCHDIRect

This command sets or queries the to edge search direction for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TOEDGESEARCHDIRect {FORWard|BACKWard}
MEASUrement:MEAS<x>:TOEDGESEARCHDIRect?
```

Arguments

MEAS<x> specifies the measurement number.

FORWard specifies a forward search to the edge.

BACKWard specifies a backward search to the edge.

Examples

MEASUrement:MEAS1:TOEDGESEARCHDIRect FORWARD specifies a forward search to the edge.

MEASUrement:MEAS1:TOEDGESEARCHDIRect? might return :MEASUREMENT:MEAS1:TOEDGESEARCHDIRECT FORWARD indicating the instrument will search in the forward direction for the to edge.

MEASUrement:MEAS<x>:TOSYmbol:LOGIC2SOUrce

This command sets or queries the DDR digital measurement logic 1 source To Symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:TOSYmbol:LOGIC2SOUrce {H|L|X}
MEASUrement:MEAS<x>:TOSYmbol:LOGIC2SOUrce?
```

Related Commands MEASUrement:ADDMEAS

Arguments `MEAS<x>` specifies the measurement number.

`H` sets the logic source 1 To Symbol to H (High).

`L` sets the logic source 1 To Symbol to L (Low).

`X` sets the logic source 1 To Symbol to X (Don't care).

Examples `MEASUrement:MEAS1:TOSYmbol:LOGIC2SOUrce H` sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) logic 1 source To Symbol to High.

`MEASUrement:MEAS1:TOSYmbol:LOGIC2SOUrce?` might return `MEASUrement:MEAS1::TOSYmbol:LOGIC2SOUrce H`, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) To Symbol is set to High.

MEASUrement:MEAS<x>:TOSYmbol:LOGIC3SOUrce

This command sets or queries the DDR digital measurement logic 2 source To Symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:TOSYmbol:LOGIC3SOUrce {H|L|X}
MEASUrement:MEAS<x>:TOSYmbol:LOGIC3SOUrce?
```

Related Commands MEASUrement:ADDMEAS

Arguments `MEAS<x>` specifies the measurement number.

`H` sets the logic source 2 To Symbol to H (High).

`L` sets the logic source 2 To Symbol to L (Low).

`X` sets the logic source 2 To Symbol to X (Don't care).

Examples `MEASUrement:MEAS1:TOSYmbol:LOGIC3SOUrce H` sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) logic 1 source To Symbol to High.

`MEASUrement:MEAS1:TOSYmbol:LOGIC3SOUrce?` might return `MEASUrement:MEAS1::TOSYmbol:LOGIC3SOUrce H`, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) To Symbol is set to High.

MEASUrement:MEAS<x>:TOSYmbol:LOGIC4SOUrce

This command sets or queries the DDR digital measurement logic 3 source To Symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Measurement

Syntax
```
MEASUrement:MEAS<x>:TOSYmbol:LOGIC4SOUrce {H|L|X}
MEASUrement:MEAS<x>:TOSYmbol:LOGIC4SOUrce?
```

Related Commands MEASUrement:ADDMEAS

Arguments

MEAS<x> specifies the measurement number.

H sets the logic source 3 To Symbol to H (High).

L sets the logic source 3 To Symbol to L (Low).

X sets the logic source 3 To Symbol to X (Don't care).

Examples

MEASUrement:MEAS1:TOSYmbol:LOGIC4SOUrce H sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) logic 1 source To Symbol to High.

MEASUrement:MEAS1:TOSYmbol:LOGIC4SOUrce? might return MEASUrement:MEAS1::TOSYmbol:LOGIC4SOUrce H, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) To Symbol is set to High.

MEASUrement:MEAS<x>:TOSYmbol:LOGIC5SOUrce

This command sets or queries the DDR digital measurement logic 4 source To Symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:TOSYmbol:LOGIC5SOUrce {H|L|X}
MEASUrement:MEAS<x>:TOSYmbol:LOGIC5SOUrce?
```

Related Commands

MEASUrement:ADDMEAS

Arguments

MEAS<x> specifies the measurement number.

H sets the logic source 4 To Symbol to H (High).

L sets the logic source 4 To Symbol to L (Low).

X sets the logic source 4 To Symbol to X (Don't care).

Examples

MEASUrement:MEAS1:TOSYmbol:LOGIC5SOUrce H sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) logic 1 source To Symbol to High.

MEASUrement:MEAS1:TOSYmbol:LOGIC5SOUrce? might return MEASUrement:MEAS1::TOSYmbol:LOGIC5SOUrce H, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) To Symbol is set to High.

MEASUrement:MEAS<x>:TOSYmbol:MEASUREAT

This command sets or queries the DDR digital measurement ToSymbol MeasureAT value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Measurement

Syntax

```
MEASUrement:MEAS<x>:TOSYmbol:MEASUREAT
{Start|Stop|ClockEdge}
MEASUrement:MEAS<x>:TOSYmbol:MEASUREAT?
```

Related Commands MEASUrement:ADDMEAS

Arguments MEAS<x> specifies the measurement number.

Start sets the ToSymbol MeasureAT to Start.

Stop sets the ToSymbol MeasureAT to Stop.

ClockEdge sets the sets the ToSymbol MeasureAT to ClockEdge.

Examples MEASUrement:MEAS1:ToSYmbol:MEASUREAT Start sets the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) ToSymbol MeasureAT value to Start.

MEASUrement:MEAS1:ToSYmbol:MEASUREAT? might return the MEASUrement:MEAS1:ToSYmbol:MEASUREAT Stop, indicating that the DDR digital measurement (DDRTRPMRS/DDRTRPACT/DDRTCKSRE/DDRTCKSRX) ToSymbol MeasureAT value is set to Stop.

MEASUrement:MEAS<x>:TRANSition

This command sets or queries the transition edges flag for the measurement. The measurement number is specified by x.

Group Measurement

Syntax `MEASUrement:MEAS<x>:TRANSition {<NR1>|OFF|ON}`

Arguments `<NR1>` = 1, the measurement is computed on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only. If it is set to 0, the measurement is computed on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges.

`OFF` computes the measurement on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges.

`ON` computes the measurement on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only.

Examples `MEASUrement:MEAS1:TRANSition 1` specifyingthe measurement is computed on rising (if measurement type is rise time) or falling edges (if measurement type is fall time) following a double transition only.

`MEASUrement:MEAS1:TRANSition?` might return `:MEASUREMENT:MEAS1:TRANSITION 0` indicating the measurement is computed on all rising (if measurement type is rise time) or falling (if measurement type is fall time) edges.

MEASUrement:MEAS<x>:TYPe

This command sets or queries the measurement type for the measurement specified by x.

Conditions Refer to the Arguments list for measurements are require the DJA option or are not available on 4 Series MSO instruments.

Group Measurement, IMDA

Syntax `MEASUrement:MEAS<x>:TYPe {ACCOMMONMODE |ACRMS |AMPlITUDE |AREA |BASE |BITAMPLITUDE |BITHIGH | BITLOW |BURSTWIDTH |COMMONMODE |DATARATE |DCD |DDJ |DDRAOS |DDRAOSPERTCK |DDRAOSPERUI |DDRAUS |DDRAUSPERTCK |DDRAUSPERUI |DDRHOLDDIFF`

```
|DDRSETUPDIFF |DDRTCHABS |DDRTCHAVERAGE |DDRTCKAVERAGE
|DDRTCLABS |DDRTCLAVERAGE |DDRTERRMN |DDRTERRN |DDRTJITCC
|DDRTJITDUTY |DDRTJITPER |DDRTPST |DDRTRPRE |DDRTWPRE
|DDRVIXAC |DDRTDQSCK |DELAY |DJ |DJDIRAC |DPMOVERSHOOT
|DPMUNDERSHOOT |DPMRIPPLE |DPMTURNOFFTIME |DPMTURNONTIME
|EYEHIGH |EYELOW |FALLSLEWRATE |FALLTIME |FREQUENCY
|F2 |F4 |F8 |HIGH |HEIGHT |HEIGHTBER |HIGHTIME |HOLD
|IMDAPOWERQUALITY |IMDAHARMONICS |IMDAINPUTVOLTAGE
|IMDAINPUTCURRENT |IMDAINPUTPOWER |IMDAPHASORDIAGRAM
|IMDAEFFICIENCY |IMDALINERIPPLE |IMDASWITCHRIPPLE
|JITTERSUMMARY |J2 |J9 |LOW |LOWTIME |MAXIMUM |MEAN |MINIMUM
|NDUtY |NPERIOD |NPJ |NOVERSHOOT |NWIDTH |PDUTTY |PERIOD
|PHASE |PHASENOISE |PJ |PK2Pk |POVERSHOOT |PWIDTH |QFACTOR
|RISESLEWRATE |RISETIME |RJ |RJDIRAC |RMS |SRJ |SSCFREQDEV
|SSCMODRATE |SETUP |SKEW |TIE |TIMEOUTSIDELEVEL |TJBER
|TNTRATIO |TOP |UNITINTERVAL |VDIFFXOVR |WIDTH |WIDTHBER}
MEASUrement:MEAS<x>:TYPe?
```

Arguments

ACCOMMONMODE AC Common Mode (Pk-Pk) is the peak-to-peak of the common mode voltage of two sources. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

ACRMS (AC RMS) is the true Root Mean Square of the data points, about the Mean. This measurement can be made across the entire record, or on each cycle in the record.

AMPLITUDE is the difference between the Top value and the Base value. This measurement can be made across the entire record, or on each cycle in the record.

Amplitude = High - Low

AREA is the area under the curve, calculated by integrating the data points. The area measured above ground is positive. The area measured below ground is negative. This measurement can be made across the entire record, or on each cycle in the record.

BASE is the most common data value below the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

BITAMPLITUDE (Bit Amplitude) is the difference between the amplitudes of the 1 bit and the 0 bit surrounding a transition. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each transition bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BITHIGH (Bit High) is the amplitude of a 1 bit. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire

record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BITLOW (Bit Low) is the amplitude of a 0 bit. The amplitude is measured over a user specified portion at the center of the recovered unit interval. This measurement is made on each high bit in the record (Mean) or across the entire record (Mode). This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

BURSTWIDTH (Burst Width) is the duration of a series of adjacent crossings of the Mid reference level (RM). Bursts are separated by a user-defined idle time (tI). This measurement is made on each burst in the record.

COMMONMODE (DC Common Mode) is the arithmetic mean of the common mode voltage of two sources. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DATARATE (Data Rate) is the reciprocal of Unit Interval. This measurement is made on each bit in the record.

DCD (duty cycle distortion) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the signal polarity. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DDJ (data dependent jitter) is the peak-to-peak amplitude of the component of the deterministic jitter correlated with the data pattern in the waveform. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DDRAOS (area above signal) is the total area of the signal above a specified reference level. This measurement is made across the entire record.

DDRAOSPERTCK (area over signal for tCK events) is the total area of the signal above a specified reference level calculated over consecutive tCK intervals. It is applicable to clock and address/command waveforms.

DDRAOSPERUI (area over signal for UI events) is the total area of the signal above a specified reference level calculated over consecutive unit intervals. It is applicable to data and data strobe waveforms.

DDRAUS (area under signal) is the total area of the signal below a specified reference level. This measurement is made across the entire record.

DDRAUSPERTCK (area under signal for tCK events) is the total area of the signal below a specified reference level calculated over consecutive tCK intervals. It is applicable to clock and address/command waveforms.

DDRAUSPERUI (area under signal for UI events) is the total area of the signal below a specified reference level calculated over consecutive unit intervals. It is applicable to data and data strobe waveforms.

DDRHOLDDIFF (hold difference) is the elapsed time between the specified edge of a single-ended clock waveform and the specified edge of a differential data waveform. The measurement uses the closest respective waveform edges that fall within the range limits.This measurement is made across the entire record.

DDRSETUPDIFF (setup difference) is the elapsed time between the specified edge of a single-ended clock waveform and when the specified edge of a differential data waveform crosses a specified level. The measurement uses the closest respective waveform edges that fall within the range limits. This measurement is made across the entire record.

DDRTCHABS (absolute high pulse width) is the absolute value of the high pulse width as measured from one rising edge to the next falling edge.

DDRTCHAVERAGE (average high pulse width) is the average value of the high pulse width as measured from one rising edge to the next falling edge, across 200 consecutive cycles.This measurement is made across the entire record.

DDRTCKAVERAGE (average clock period) is the average clock period calculated from rising edge to rising edge, across 200 consecutive cycles.This measurement is made across the entire record.

DDRTCLABS (absolute low pulse width) is the absolute value of the low pulse width as measured from a falling edge to the next rising edge. This measurement is made across the entire record.

DDRTCLAVERAGE (average low pulse width) is the average value of the low pulse width as measured from one falling edge to the next rising edge, across 200 consecutive cycles.

DDRTERRMN (cumulative error) is the cumulative error across multiple consecutive defined cycles from tCK(avg).

DDRTERRN (cumulative error) is the cumulative error across specified consecutive cycles from tCK(avg). In other words, this measures the time difference between the sum of the clock period from a 200 cycle window and n times tCK(avg).

DDRTJITCC (cycle to cycle jitter period) is the absolute difference in clock period between two consecutive clock cycles. This measurement is made across the entire record.

DDRTJITDUTY (half period jitter) is the largest elapsed time between tCH and tCH(avg), and tCL and tCL(avg), over 200 consecutive cycles.

DDRTJITPER (clock period jitter) is the largest deviation of any tCK signal from tCK(avg). This measurement is made across the entire record.

DDRTPST (read/write burst postamble) is the width of the Read or Write burst postamble, measured from the last falling edge of the mid reference level to the start of an undriven state. This measurement is made across the entire record.

DDRTRPRE (read burst preamble) is the width of the Read burst preamble, measured from exiting tristate levels to the first driving edge of the differential strobe. This measurement is made across the entire record.

DDRTWPRE is the width of the Write burst preamble, measured from exiting tristate levels to the first driving edge of the differential strobe. This measurement is made across the entire record.

DDRVIXAC is the differential input cross-point voltage measured from the true state transition (and it's compliment) to a specified reference level, measured on a single-ended signal.

DDRTDQSCK is the strobe output access time, measured between the rising edge of the clock and before or after the differential strobe Read preamble time. Signal edges are determined by the mid-ref threshold level settings.

DELay is the time between the specified Mid reference level (RM) crossing on one source to a specified Mid reference level (RM) crossing on a second source. This measurement is made on the first occurrence in the record.

DJ (deterministic jitter) is the peak-to-peak amplitude of all timing errors that exhibit deterministic behavior. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DJDIRAC (dual-dirac deterministic jitter) is deterministic jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

DPMOVERSHOOT is the difference between Maximum and Top, divided by the amplitude. This measurement can be made across the entire record or on each cycle in the record at the specified reference voltage.

DPMUNDERSHOOT is the difference between Minimum and Base, divided by the amplitude. This measurement can be made across the entire record or on each cycle in the record at the specified reference voltage.

DPMRIPPLE is the RMS and peak-to-peak values of the output signal on the DC Rail.

DPMTURNOFFTIME is the time delay between load current going low to other rail outputs going low during power down.

DPMTURNONTIME is the time delay between load current going high to other rail outputs going high during power on

EYEHIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

EYELOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

FALLSLEWRATE (Falling Slew Rate) is the rate of change in voltage as an edge transitions from the Top reference level (RT) to the Bottom reference level (RB). This measurement is made on each cycle in the record.

FALLTIME (Fall Time) is the time required for an edge to fall from the Top reference level (RT) to the Base reference level (RB). This measurement is made on each cycle in the record.

FREQuency is the reciprocal of Period. This measurement is made on each cycle in the record.

F2 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 2. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

F4 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 4. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

F8 is the peak-to-peak amplitude of the periodic jitter occurring at a rate of Fb (data rate) divided by 8. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HIGH (Eye High) is the amplitude of a high (1) bit measured at a user specified location within the recovered unit interval. This measurement is made on each high bit in the record.

HEIGHT (Eye Height) is the minimum vertical eye opening at the center of the recovered unit interval. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HEIGHTBER (Eye Height@BER) is the predicted vertical eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

HIGH

HIGHTIME (High Time) is the time the signal remains above the Top reference level (RT). This measurement is made on each cycle in the record.

HOLD (Hold Time) is the time between the specified Mid reference level crossing (RM) on the Clock source to the closest specified Mid reference level (RM)

crossing on the Data source. This measurement is made on each specified Clock edge in the record.

IMDAPOWERQUALITY (Power Quality) measures the Frequency and RMS values of the voltage and current, Crest Factors of the voltage and current, True Power (PTRUE), Reactive Power (PRE), Apparent Power (PAPP), Power Factor, and Phase Angle (θ) of the AC signal.

IMDAHARMONICS (Harmonics) plots the signal amplitude at the fundamental line frequency and its harmonics and measures the RMS amplitude and Total Harmonic Distortion of the signal.

IMDAINPUTVOLTAGE (Input Voltage) measures the RMS value of the Input Voltages. The number of voltages can vary with wiring configuration.

IMDAINPUTCURRENT (Input Current) measures the RMS value of the Input Currents. The number of currents can vary with wiring configuration.

IMDAINPUTPOWER (Input Power) measures the RMS value of the Input Power. The power waveforms are shown as derived MATH waveforms, which are respective pairs of Input Voltage * Current waveforms

IMDAPHASORDIAGRAM (Phasor Diagram) represents magnitudes and phase angles between voltage(s) and current(s) per winding pair.

IMDAEFFICIENCY (Efficiency) measures the ratio of sum of output power(s) to input power for respective input and output Voltage and Current pairs. Note: the current release cannot support 3V and 3I pairs, since this requires 12 channels. We restrict to 2V and 2I which needs 8 scope channels.

IMDALINERIPPLE (Line Ripple) provides the RMS and peak-to-peak measurements of the line frequency portion of the respective AC Voltage or Current signals.

IMDASWITCHINGRIPPLE (Switching Ripple) provides the RMS and peak-to-peak measurements of the respective Voltage or Current signals.

JITTERSUMMARY (Jitter Summary) is a group consisting of the following measurements: TIE, TJ@BER, Eye Width@BER, Eye Height@BER, RJ-δδ, DJ-δδ, PJ, DDJ, DcD, F/2, F/4, F/8. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

J2 is the total jitter at a bit error rate of 2.5e-3 (TJ@2.5e-3). This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

J9 is the total jitter at a bit error rate of 2.5e-10 (TJ@2.5e-10). This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

LOW (Eye Low) is the amplitude of a low (0) bit measured at a user specified location within the recovered unit interval. This measurement is made on each low bit in the record.

LOWTIME (Low Time) is the time the signal remains below the Base reference level (RB). This measurement is made on each cycle in the record.

MAXimum is the maximum data point. This measurement can be made across the entire record, or on each cycle in the record.

MEAN is the arithmetic mean of the data points. This measurement can be made across the entire record, or on each cycle in the record.

MINImum is the minimum data point. This measurement can be made across the entire record, or on each cycle in the record.

NDUty (Negative Duty Cycle) is the ratio of the Negative Pulse Width to the Period. This measurement is made on each cycle in the record.

Negative Duty Cycle = (Negative Width) / Period × 100%

NPERIOD (Duration N-Periods) is the time required to complete N cycles. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

NPJ (non-periodic jitter) is the portion of the BUJ (bounded uncorrelated jitter) that is random. BUJ excludes DDJ, DCD and RJ. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

NOVershoot (Negative Overshoot) is the difference between Minimum and Base, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

Negative Overshoot = (Base - Minimum) / Amplitude × 100%)

NWIdth (Negative Pulse Width) is the time the signal remains below the Mid reference level (RM). This measurement is made on each cycle in the record.

PDUTY (Positive Duty Cycle) is the ratio of the Positive Pulse Width to the Period. This measurement is made on each cycle in the record.

Positive Duty Cycle = (Positive Width)/Period × 100%

PERIOD is the time required to complete a cycle. A cycle is the time between two adjacent (same direction) crossings of the Mid reference level (RM). This measurement is made on each cycle in the record.

PHASE is the ratio of the Skew between two sources to the Period of the first source. This measurement is made on each cycle in the record.

PHASENOISE (Phase Noise) is the RMS magnitude of all integrated jitter falling within a user specified offset range of the fundamental clock frequency. This measurement is made across the entire record. This measurement is not available on a 4 Series MSO instrument.

PJ (periodic jitter) is the peak-to-peak amplitude of the uncorrelated sinusoidal components of the deterministic jitter. This measurement is made across the

entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

PK2Pk (Peak-to-peak) is the difference between Maximum and Minimum. This measurement can be made across the entire record, or on each cycle in the record.

POVERSHOOT (Positive Overshoot) is the difference between Maximum and Top, divided by the Amplitude. This measurement can be made across the entire record, or on each cycle in the record.

Positive Overshoot = (Maximum - Top) / Amplitude ×100%

PWIDTH (Positive Pulse Width) is the time the signal remains above the Mid reference level (RM). This measurement is made on each cycle in the record.

QFACTOR (Q-Factor) is the ratio of the vertical eye opening to RMS vertical noise measured at a user specified location within the recovered unit interval. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RISESLEWRATE (Rising Slew Rate) is the rate of change in voltage as an edge transitions from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RISETIME Rise Time is the time required for an edge to rise from the Base reference level (RB) to the Top reference level (RT). This measurement is made on each cycle in the record.

RJ (random jitter) is the RMS magnitude of all random timing errors following a Gaussian distribution. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RJDIRAC (dual-dirac random jitter) is random jitter based on a simplifying assumption that the histogram of all deterministic jitter can be modeled as a pair of equal-magnitude Dirac functions. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

RMS is the true Root Mean Square of the data points. This measurement can be made across the entire record, or on each cycle in the record.

SRJ (sub-rate jitter) is the composite jitter due to periodic components at 1/2, 1/4 and 1/8 of the data rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SSCFREQDEV (SSC Frequency Deviation) is the spread spectrum clock frequency deviation. This measurement enables a time trend plot of the spread spectrum clock modulation profile. This measurement is made on each cycle in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SSCMODRATE (SSC Modulation Rate) is the modulating frequency of a spread spectrum clock. This measurement is made on each cycle in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

SETUP (Setup Time) is the time between the specified Mid reference level (RM) crossing on the Data source to the closest specified Mid reference level (RM) crossing on the Clock source. This measurement is made on each specified Clock edge in the record.

SKEW Skew is the time between the specified Mid reference level (RM) crossing on one source to the following specified Mid reference level (RM) crossing on a second source. This measurement is made on each cycle in the record.

TIE (time interval error) is the difference, in time, between an edge in the source waveform and the corresponding edge in a recovered reference clock. This measurement is made on each edge in the waveform. This measurement is not available on a 4 Series MSO instrument.

TIMEOUTSIDELEVEL Time Outside Level is the time the signal remains above the Top reference level (RT) and/or below the Base reference level (RB). This measurement is made on each occurrence in the record.

TJBER (total jitter at a specified bit error rate) is the predicted peak-to-peak amplitude of jitter that will only be exceeded with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

TNTRATIO T/nT Ratio is the ratio of a non-transition bit voltage (2nd and subsequent bit voltage after a transition) to its nearest preceding transition bit voltage (1st bit voltage after the transition). Bit voltages are measured at the interpolated midpoint of the recovered unit interval. This measurement is made on each non-transition bit in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

TOP is the most common data value above the midpoint of the waveform. This measurement can be made across the entire record, or on each cycle in the record.

UNITINTERVAL (Unit Interval) is the time difference between two successive bits. This measurement is made on each bit in the record.

VDIFFXOVR (Differential Crossover) is the voltage level of a differential signal pair at the crossover points. This measurement is made at each crossover point in the record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

WIDTH (Eye Width) is the minimum horizontal eye opening at the user specified reference level. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

WIDTHBER (Eye Width@BER) is the predicted horizontal eye opening that will be violated with a probability equal to the bit error rate. This measurement is made across the entire record. This measurement requires the DJA option and is not available on a 4 Series MSO instrument.

Examples

MEASUREMENT:MEAS2:TYPE FREQUENCY defines measurement 2 as a measurement of the frequency of a waveform.

MEASUREMENT:MEAS1:TYPE? might return :MEASUREMENT:MEAS1:TYPE RMS, indicating that measurement 1 is defined to measure the RMS value of a waveform.

MEASUrement:MEAS<x>:WAITTime

Sets or queries the wait time.

Conditions

Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group

DPM

Syntax

```
MEASUrement:MEAS<x>:WAITTime <NR2>
MEASUrement:MEAS<x>:WAITTime?
```

Arguments

MEAS<x> is the measurement number.

<NR2> is the wait time, in the range of 1 s to 500 s.

Examples

MEASUrement:MEAS1:WAITTime 60 sets the input trigger level of measurement 1 to +60 volts.

MEASUrement:MEAS3:WAITTime? might return 2.500, indicating that the wait time of measurement 3 is 2.5 seconds.

MEASUrement:MEAS<x>:WINDOWLENgth

This command sets or queries the window length for the measurement. The measurement number is specified by x.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:WINDOWLENgth <NR3>
MEASUrement:MEAS<x>:WINDOWLENgth?
```

Arguments

<NR3> is the value for the window length.

Examples

MEASUrement:MEAS1:WINDOWLENgth 10 sets the window length to 10.

MEASUrement:MEAS1:WINDOWLENgth? might return :MEASUREMENT:MEAS1:WINDOWLENGTH 10.0000 indicating the window length is 10.

MEASUrement:MEAS<x>:XUNIT? (Query Only)

Returns the horizontal scale units of the specified measurement.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:XUNIT?
```

Arguments

MEAS<x> specifies the measurement number.

Examples

MEASUrement:MEAS2:XUNIT? might return "V", indicating that the horizontal unit for measurement 2 is seconds.

MEASUrement:MEAS<x>:YUNIT? (Query Only)

Returns the vertical scale units of the specified measurement.

Group

Measurement

Syntax

```
MEASUrement:MEAS<x>:YUNIT?
```

Arguments

MEAS<x> specifies the measurement number.

Examples

MEASUrement:MEAS12:YUNIT? might return “V”, indicating that the vertical unit for measurement 12 is volts.

MEASUrement:MINUI

This command sets or queries the minimum number of unit intervals required for BUJ analysis.

Group Measurement

Syntax

```
MEASUrement:MINUI <NR3>
MEASUrement:MINUI?
```

Arguments <NR3> is the minimum number of unit intervals required for BUJ analysis.

Examples MEASUREMENT:MINUI 1.0000E+6 sets the minimum number of unit intervals required for BUJ analysis to 1,000,000.

MEASUREMENT:MINUI? might return :MEASUREMENT:MINUI 1.0000E+6 indicating the minimum number of unit intervals required for BUJ analysis is 1,000,000.

MEASUrement:POPUlation:LIMIT:STATE

This command sets or queries the global population limit state for the measurement.

Group Measurement

Syntax

```
MEASUrement:POPUlation:LIMIT:STATE {OFF|ON|0|1 }
MEASUrement:POPUlation:LIMIT:STATE?
```

Arguments OFF turns off the population limit.

ON turns on the population limit.

0 turns off the population limit.

1 turns on the population limit.

Examples MEASUREMENT:POPULATION:LIMIT:STATE 1 turns on the population limit.

MEASUREMENT:POPULATION:LIMIT:STATE? might return :MEASUREMENT:POPULATION:LIMIT:STATE 0 indicating the population limit is off.

MEASUrement:POPUlation:LIMIT:VALue

This command sets or queries the global population limit value for the measurement.

Group Measurement

Syntax
```
MEASUrement:POPUlation:LIMIT:VALue <NR1>
MEASUrement:POPUlation:LIMIT:VALue?
```

Arguments <NR1> is the current limit value.

Examples MEASUREMENT:POPULATION:LIMIT:VALUE 2000 sets the population limit to 2000.

MEASUREMENT:POPULATION:LIMIT:VALUE? might return :MEASUREMENT:POPULATION:LIMIT:VALUE 1000 indicating the population limit value is 1000.

MEASUrement:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax
```
MEASUrement:REFLevels:ABSolute:FALLHigh <NR3>
MEASUrement:REFLevels:ABSolute:FALLHigh?
```

Arguments <NR3> is the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples MEASUrement:REFLevels:ABSolute:FALLHigh 1.5 sets the high reference level to 1.5 V.

MEASUrement:REFLevels:ABSolute:FALLHigh? might return :MEASUREMENT:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating the high reference level is 1.0 V.

MEASUrement:REFLevels:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax
```
MEASUrement:REFLevels:ABSolute:FALLLow <NR3>
MEASUrement:REFLevels:ABSolute:FALLLow?
```

Arguments <NR3> is the value used as the low reference level of the falling edge.

Examples `MEASUrement:REFLevels:ABSolute:FALLLow -1.5` sets the low reference level to -1.5 V.

`MEASUrement:REFLevels:ABSolute:FALLLow?` might return `:MEASUREMENT:REFLEVELS:ABSOLUTE:FALLLOW -1.0000` indicating the low reference level is -1.0 V.

MEASUrement:REFLevels:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax
```
MEASUrement:REFLevels:ABSolute:FALLMid <NR3>
MEASUrement:REFLevels:ABSolute:FALLMid?
```

Arguments <NR3> is the value used as the mid reference level of the falling edge.

Examples `MEASUrement:REFLevels:ABSolute:FALLMid 10.0E-3` sets the mid reference level to 10.0 mV.

`MEASUrement:REFLevels:ABSolute:FALLMid?` might return `:MEASUREMENT:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0` indicating the mid reference level is 0.0 V.

MEASUrement:REFLevels:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Group Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:HYSTeresis <NR3>
MEASUrement:REFLevels:ABSolute:HYSTeresis?
```

Arguments <NR3> is the value of the hysteresis of the reference level.

Examples `MEASUrement:REFLevels:ABSolute:HYSTeresis 20.0E-3` sets the hysteresis to 20.0 mV.

`MEASUrement:REFLevels:ABSolute:HYSTeresis?` might return `:MEASUREMENT:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3` indicating the hysteresis is set to 30.0 mV.

MEASUrement:REFLevels:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax

```
MEASUrement:REFLevels:ABSolute:RISEHigh <NR3>
MEASUrement:REFLevels:ABSolute:RISEHigh?
```

Arguments <NR3> is the value used as the high reference level of the rising edge.

Examples `MEASUrement:REFLevels:ABSolute:RISEHigh 1.5` sets the high reference to 1.5 V.

`MEASUrement:REFLevels:ABSolute:RISEHigh?` might return `:MEASUREMENT:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000` indicating the high reference is 1.0 V.

MEASUrement:REFLevels:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax
```
MEASUrement:REFLevels:ABSolute:RISELow <NR3>
MEASUrement:REFLevels:ABSolute:RISELow?
```

Arguments <NR3> is the value used as the the low reference level of the rising edge

Examples `MEASUrement:REFLevels:ABSolute:RISELow -1.5` sets the low reference level to -1.5 V.

`MEASUrement:REFLevels:ABSolute:RISELow?` might return `:MEASUREMENT:REFLEVELS:ABSOLUTE:RISELOW -1.0000` indicating the low reference level is -1.0 V.

MEASUrement:REFLevels:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax
```
MEASUrement:REFLevels:ABSolute:RISEMid <NR3>
MEASUrement:REFLevels:ABSolute:RISEMid?
```

Arguments <NR3> is the mid reference level of the rising edge.

Examples `MEASUrement:REFLevels:ABSolute:RISEMid 10.0E-3` sets the mid reference to 10.0 mV.

`MEASUrement:REFLevels:ABSolute:RISEMid?` might return `:MEASUREMENT:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0` indicating the mid reference is 0.0 V.

MEASUrement:REFLevels:ABSolute:TYPE

This command sets or queries the reference level type for the measurement.

Group Measurement

Syntax
```
MEASUrement:REFLevels:ABSolute:TYPE {SAME|UNIQue}
MEASUrement:REFLevels:ABSolute:TYPE?
```

Arguments `SAME` specifies that the absolute levels are set the same.

`UNIQue` specifies that the absolute levels can be set independently.

Examples `MEASUrement:REFLevels:ABSolute:TYPE UNIQUE` specifies that the absolute levels can be set independently.

`MEASUrement:REFLevels:ABSolute:TYPE?` might return `:MEASUREMENT:REFLEVELS:ABSOLUTE:TYPE SAME` indicating the absolute levels are set the same.

MEASUrement:REFLevels:BASETop

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.

Group Measurement

Syntax
```
MEASUrement:REFLevels:BASETop
{AUTO|MINMax|MEANhistogram|MODEhistogram|EYEhistogram}
MEASUrement:REFLevels:BASETop?
```

Arguments Arguments are the base top methods.

Examples `MEASUREMENT:REFLEVELS:BASETOP MINMAX` selects the MINMAX base top method.

`MEASUREMENT:REFLEVELS:BASETOP?` might return `:MEASUREMENT:REFLEVELS:BASETOP MINMAX` indicating the base top method is MINMAX.

MEASUrement:REFLevels:JITTERMODE

This command sets or queries how often reference levels are calculated on Jitter measurements. If the mode is set to Latch, ref levels are calculated only on the first acquisition after a statistics reset. If it is set to Continuous, reference levels are calculated on every acquisition.

Group Measurement

Syntax
```
MEASUrement:REFLevels:JITTERMODE {CONTinuous|LATCh}
MEASUrement:REFLevels:JITTERMODE?
```

Arguments `CONTinuous` specifies that reference levels are calculated on every acquisition.

`LATCh` specifies that reference levels are calculated only on the first acquisition after a statistics reset.

Examples `MEASUREMENT:REFLEVELS:JITTERMODE CONTINUOUS` specifies that reference levels are calculated on every acquisition.

`MEASUREMENT:REFLEVELS:JITTERMODE?` might return LATCH indicating that reference levels are calculated only on the first acquisition after a statistics reset.

MEASUrement:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement.

Group Measurement

Syntax
```
MEASUrement:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:REFLevels:METHod?
```

Arguments `PERCent` specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the `MEASUrement:REFLevels:REFLevel:PERCent` commands.

`ABSolute` specifies that the reference levels are set explicitly using the `MEASUrement:REFLevels:REFLevel:ABSolute` commands. This method is useful when precise values are required.

Examples `MEASUrement:REFLevels:METHod ABSOLUTE` specifies that the reference levels are set explicitly.

`MEASUrement:REFLevels:METHod?` might return `:MEASUREMENT:REFLEVELS:METHOD PERCENT` indicating the reference levels are calculated as a percent relative to HIGH and LOW.

MEASUrement:REFLevels:MODE

This command sets or queries how often reference levels are calculated.

Group Measurement

Syntax
```
MEASUrement:REFLevels:MODE {LATCh|CONTinuous}
MEASUrement:REFLevels:MODE?
```

Arguments `LATCh` calculates reference levels only on the first acquisition after a statistics reset.

`CONTinuous` calculates reference levels on every acquisition.

Examples `MEASUREMENT:REFLEVELS:MODE CONTINUOUS` calculates reference levels on every acquisition.

`MEASUREMENT:REFLEVELS:MODE?` might return `:MEASUREMENT:REFLEVELS:MODE LATCH` indicating reference levels are calculated only on the first acquisition after a statistics reset.

MEASUrement:REFLevels:PERCent:FALLHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REFLevels:PERCent:FALLHigh <NR3>
MEASUrement:REFLevels:PERCent:FALLHigh?
```

Arguments `<NR3>` is the percentage used to calculate the high reference level of the falling edge

Examples `MEASUrement:REFLevels:PERCent:FALLHigh 90` sets the high reference level to 90%.

`MEASUrement:REFLevels:PERCent:FALLHigh?` might return `:MEASUREMENT:REFLEVELS:PERCENT:FALLHIGH 80.0000` indicating the high reference level is 80%.

MEASUrement:REFLevels:PERCent:FALLLow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REFLevels:PERCent:FALLLow <NR3>
MEASUrement:REFLevels:PERCent:FALLLow?
```

Arguments <NR3> is the percentage used to calculate the mid reference level of the falling edge.

Examples `MEASUrement:REFLevels:PERCent:FALLLow 10` sets the low reference level to 10%.

`MEASUrement:REFLevels:PERCent:FALLLow?` might return `:MEASUREMENT:REFLEVELS:PERCENT:FALLLOW 20.0000` indicating the low reference level is 20%.

MEASUrement:REFLevels:PERCent:FALLMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REFLevels:PERCent:FALLMid <NR3>
MEASUrement:REFLevels:PERCent:FALLMid?
```

Arguments <NR3> is the percentage used to calculate the mid reference level of the falling edge

Examples `MEASUrement:REFLevels:PERCent:FALLMid 55` sets the mid reference level to 55%.

`MEASUrement:REFLevels:PERCent:FALLMid?` might return `:MEASUREMENT:REFLEVELS:PERCENT:FALLMID 50.0000` indicating the mid reference level is 50%.

MEASUrement:REFLevels:PERCent:HYSTeresis

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:REFLevels:PERCent:HYSTeresis?
```

Arguments <NR3> is the percentage used to calculate the hysteresis of the reference level.

Examples `MEASUrement:REFLevels:PERCent:HYSTeresis 3` sets the hysteresis to 3%.

`MEASUrement:REFLevels:PERCent:HYSTeresis?` might return `:MEASUREMENT:REFLEVELS:PERCENT:HYSTERESIS 5.0000` indicating the hysteresis is set to 5%.

MEASUrement:REFLevels:PERCent:RISEHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REFLevels:PERCent:RISEHigh <NR3>
MEASUrement:REFLevels:PERCent:RISEHigh?
```

Arguments <NR3> is the percentage used to calculate the high reference level of the rising edge.

Examples `MEASUrement:REFLevels:PERCent:RISEHigh 90` sets the high reference level to 90%.

`MEASUrement:REFLevels:PERCent:RISEHigh?` might return `:MEASUREMENT:REFLEVELS:PERCENT:RISEHIGH 80.0000` indicating the high ref level is 80%.

MEASUrement:REFLevels:PERCent:RISELow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REFLevels:PERCent:RISELow <NR3>
MEASUrement:REFLevels:PERCent:RISELow?
```

Arguments <NR3> is the percentage used to calculate the low reference level of the rising edge.

Examples `MEASUrement:REFLevels:PERCent:RISELow` 10 sets the low reference level to 10%.

`MEASUrement:REFLevels:PERCent:RISELow?` might return `:MEASUREMENT:REFLEVELS:PERCENT:RISELOW 20.0000` indicating the low reference level is 20%.

MEASUrement:REFLevels:PERCent:RISEMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REFLevels:PERCent:RISEMid <NR3>
MEASUrement:REFLevels:PERCent:RISEMid?
```

Arguments <NR3> is the percentage used to calculate the mid reference level of the rising edge.

Examples `MEASUrement:REFLevels:PERCent:RISEMid 55` sets the mid reference level to 55%.

`MEASUrement:REFLevels:PERCent:RISEMid?` might return `:MEASUREMENT:REFLEVELS:PERCENT:RISEMID 50.0000` indicating the mid reference level is 50%.

MEASUrement:REFLevels:PERCent:TYPE

This command sets or queries the reference level percent type for the measurement.

Group Measurement

Syntax

```
MEASUrement:REFLevels:PERCent:TYPE
{TENNinety|TWENtyeighty|CUSTom}
MEASUrement:REFLevels:PERCent:TYPE?
```

Arguments

`TENNinety` sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

`TWENtyeighty` sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

`CUSTom` allows setting other reference level percents.

Examples

`MEASUrement:REFLevels:PERCent:TYPE TENNINETY` sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

`MEASUrement:REFLevels:PERCent:TYPE?` might return `:MEASUREMENT:REFLEVELS:PERCENT:TYPE CUSTOM` indicating that custom reference levels can be set.

MEASUrement:REFLevels:TYPE

This command sets or queries the shared reference level method used for sources of measurement calculations.

Group Measurement

Syntax

```
MEASUrement:REFLevels:TYPE {GLOBal|PERSource}
MEASUrement:REFLevels:TYPE?
```

Arguments

`GLOBal` shares reference levels across measurements.

`PERSource` causes reference levels to be used on individual measurements.

Examples

`MEASUREMENT:REFLEVELS:TYPE PERSource` causes reference levels to be used on individual measurements.

MEASUREMENT:REFLEVELS:TYPE? might return :MEASUREMENT:REFLEVELS:TYPE GLOBAL indicating reference levels are shared across measurements.

MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:FALLHigh?
```

Arguments <NR3> is the value used as the high reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples MEASUrement:REF1:REFLevels:ABSolute:FALLHigh 1.5 sets the reference level to 1.5 V.

MEASUrement:REF1:REFLevels:ABSolute:FALLHigh? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLHIGH 1.0000 indicating the reference level is 1.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow

This command sets or queries the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:FALLLow?
```

Arguments <NR3> is the value used as the low reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples MEASUrement:REF1:REFLevels:ABSolute:FALLLow -1.5 sets the low reference level to -1.5 V.

MEASUrement:REF1:REFLevels:ABSolute:FALLLow? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLLOW -1.0000 indicating the low reference level is -1.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:FALLMid

This command sets or queries the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:ABSolute:FALLMid <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:FALLMid?
```

Arguments <NR3> is the value used as the mid reference level of the falling edge when the measurement's ref level method is set to absolute.

Examples MEASUrement:REF1:REFLevels:ABSolute:FALLMid 50.0E-3 sets the mid reference level to 50.0 mV.

MEASUrement:REF1:REFLevels:ABSolute:FALLMid? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:FALLMID 0.0E+0 indicating the reference level is 0.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis

This command sets or queries the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:HYSTeresis?
```

Arguments <NR3> is the value of the hysteresis of the reference level when the measurement's ref level method is set to absolute.

Examples MEASUrement:REF1:REFLevels:ABSolute:HYSTeresis 20.0E-3 sets the hysteresis to 20.0 mV.

MEASUrement:REF1:REFLevels:ABSolute:HYSTeresis? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:HYSTERESIS 30.0000E-3 indicating the hysteresis is 30.0 mV.

MEASUrement:REF<x>:REFLevels:ABSolute:RISEHigh

This command sets or queries the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:ABSolute:RISEHigh <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:RISEHigh?
```

Arguments <NR3> is the value used as the high reference level of the rising edge when the measurement's ref level method is set to absolute.

Examples MEASUrement:REF1:REFLevels:ABSolute:RISEHigh 1.5 sets the reference level to 1.5 V.

MEASUrement:REF1:REFLevels:ABSolute:RISEHigh? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEHIGH 1.0000 indicating the reference level is 1.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:RISELow

This command sets or queries the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:ABSolute:RISELow <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:RISELow?
```

Arguments <NR3> is the value used as the low reference level of the rising edge when the measurement's ref level method is set to absolute.

Examples MEASUrement:REF1:REFLevels:ABSolute:RISELow -1.5 sets the reference level to -1.5 V.

MEASUrement:REF1:REFLevels:ABSolute:RISELow? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISELOW -1.0000 indicating the reference level is -1.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:RISEMid

This command sets or queries the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:ABSolute:RISEMid <NR3>
MEASUrement:REF<x>:REFLevels:ABSolute:RISEMid?
```

Arguments <NR3> is the value used as the mid reference level of the rising edge when the measurement's ref level method is set to absolute.

Examples MEASUrement:REF1:REFLevels:ABSolute:RISEMid 10.0E-3 sets the reference level to 10.0 mV.

MEASUrement:REF1:REFLevels:ABSolute:RISEMid? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:RISEMID 0.0E+0 indicating the mid reference level is 0.0 V.

MEASUrement:REF<x>:REFLevels:ABSolute:TYPE

This command sets or queries the reference level type for the measurement.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:ABSolute:TYPE {SAME|UNIQue}
MEASUrement:REF<x>:REFLevels:ABSolute:TYPE?
```

Arguments SAME specifies that the absolute levels are set the same.

UNIQue specifies that the absolute levels can be set independently.

Examples MEASUrement:REF1:REFLevels:ABSolute:TYPE UNIQUE specifies that the absolute levels can be set independently.

MEASUrement:REF1:REFLevels:ABSolute:TYPE? might return :MEASUREMENT:REF1:REFLEVELS:ABSOLUTE:TYPE SAME indicating that the absolute levels are set the same.

MEASUrement:REF<x>:REFLevels:BASETop

This command sets or queries the method used to calculate the TOP and BASE, used to calculate reference levels for the measurement.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:BASETop
```

Arguments

AUTO automatically chooses a reference level method.

MINMax specifies that reference levels are relative to the measurement MIN and MAX.

MEANhistogram specifies that reference levels are relative to the histogram mean BASE and TOP.

MODEhistogram specifies that reference levels are relative to the histogram mode BASE and TOP.

EYEhistogram specifies that reverence levels are relative to the eye histogram BASE and TOP.

Examples

MEASUrement:REF1:REFLevels:BASETop MINMAX specifies that reference levels are relative to the measurement MIN and MAX.

MEASUrement:REF1:REFLevels:BASETop? might return :MEASUREMENT:REF1:REFLEVELS:BASETOP AUTO indicating the instrument automatically chooses a reference level method.

MEASUrement:REF<x>:REFLevels:METHod

This command sets or queries the method used to calculate reference levels for the measurement.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:METHod {PERCent|ABSolute}
MEASUrement:REF<x>:REFLevels:METHod?
```

Arguments

PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:REF<x>:REFLevel:PERCent commands.

ABSolute specifies that the reference levels are set explicitly using the MEASUrement:REF<x>:REFLevel:ABSolute commands. This method is useful when precise values are required.

Examples

MEASUrement:REF1:REFLevels:METHod ABSOLUTE specifies that the reference levels are set explicitly.

MEASUrement:REF1:REFLevels:METHod? might return :MEASUREMENT:REF1:REFLEVELS:METHOD PERCENT indicating the reference levels are calculated as a percent relative to HIGH and LOW.

MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the falling edge when the measurement's ref level method is set to percent.

Group

Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:FALLHigh?
```

Arguments

<NR3> is the percentage used to calculate the high reference level of the falling edge

Examples

MEASUrement:REF1:REFLevels:PERCent:FALLHigh 95 sets the reference level to 95%.

MEASUrement:REF1:REFLevels:PERCent:FALLHigh? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLHIGH 90.0000 indicating the reference level is 90%.

MEASUrement:REF<x>:REFLevels:PERCent:FALLLow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the falling edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:PERCent:FALLLow <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:FALLLow?
```

Arguments <NR3> is the percentage used to calculate the low reference level

Examples `MEASUrement:REF1:REFLevels:PERCent:FALLLow 5` sets the low reference level to 5%.

`MEASUrement:REF1:REFLevels:PERCent:FALLLow?` might return `:MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLLOW 10.0000` indicating the low reference level is 10%.

MEASUrement:REF<x>:REFLevels:PERCent:FALLMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the falling edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:PERCent:FALLMid <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:FALLMid?
```

Arguments <NR3> is the percentage used to calculate the mid reference level of the falling edge.

Examples `MEASUrement:REF1:REFLevels:PERCent:FALLMid 55` sets the reference level to 50%.

`MEASUrement:REF1:REFLevels:PERCent:FALLMid?` might return `:MEASUREMENT:REF1:REFLEVELS:PERCENT:FALLMID 50.0000` indicating the mid reference level is 50%.

MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis

This command sets or queries the percentage (where 100% is equal to MAX and 0% is equal to MIN) used to calculate the hysteresis of the reference level when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:HYSTeresis?
```

Arguments <NR3> is the percentage used to calculate the hysteresis of the reference level.

Examples `MEASUrement:REF1:REFLevels:PERCent:HYSTeresis 2` sets the hysteresis to 2%.

`MEASUrement:REF1:REFLevels:PERCent:HYSTeresis?` might return `:MEASUREMENT:REF1:REFLEVELS:PERCENT:HYSTERESIS 5.0000` indicating the hysteresis is 5%.

MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the high reference level of the rising edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:RISEHigh?
```

Arguments <NR3> is the percentage used to calculate the high reference level of the rising edge

Examples `MEASUrement:REF1:REFLevels:PERCent:RISEHigh 95` sets the reference level to 95%.

`MEASUrement:REF1:REFLevels:PERCent:RISEHigh?` might return `:MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEHIGH 90.0000` indicating the reference level is 90%.

MEASUrement:REF<x>:REFLevels:PERCent:RISELow

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the low reference level of the rising edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:PERCent:RISELow <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:RISELow?
```

Arguments <NR3> is the percentage used to calculate the low reference level of the rising edge.

Examples `MEASUrement:REF1:REFLevels:PERCent:RISELow 5` sets the low reference level to 5%.

`MEASUrement:REF1:REFLevels:PERCent:RISELow?` might return `:MEASUREMENT:REF1:REFLEVELS:PERCENT:RISELOW 10.0000` indicating the reference level is 10%.

MEASUrement:REF<x>:REFLevels:PERCent:RISEMid

This command sets or queries the percentage (where 100% is equal to TOP and 0% is equal to BASE) used to calculate the mid reference level of the rising edge when the measurement's ref level method is set to percent.

Group Measurement

Syntax
```
MEASUrement:REF<x>:REFLevels:PERCent:RISEMid <NR3>
MEASUrement:REF<x>:REFLevels:PERCent:RISEMid?
```

Arguments <NR3> is the percentage used to calculate the mid reference level of the rising edge.

Examples `MEASUrement:REF1:REFLevels:PERCent:RISEMid 55` sets the mid reference level to 55%.

`MEASUrement:REF1:REFLevels:PERCent:RISEMid?` might return `:MEASUREMENT:REF1:REFLEVELS:PERCENT:RISEMID 50.0000` indicating the mid reference level is 50%.

MEASUrement:REF<x>:REFLevels:PERCent:TYPE

This command sets or queries the reference level percent type for the measurement.

Group Measurement

Syntax

```
MEASUrement:REF<x>:REFLevels:PERCent:TYPE
{TENNinety|TWENtyeighty|CUSTom}
MEASUrement:REF<x>:REFLevels:PERCent:TYPE?
```

Arguments

TENNinety sets the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

TWENtyeighty sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

CUSTom allows setting other reference level percents.

Examples

MEASUrement:REF1:REFLevels:PERCent:TYPE TWENTYEIGHTY sets the values for Low, Mid and High Ref are set to 20%, 50% and 80% respectively.

MEASUrement:REF1:REFLevels:PERCent:TYPE? might return :MEASUREMENT:REF1:REFLEVELS:PERCENT:TYPE TENNINETY indicating the values for Low, Mid and High Ref to 10%, 50% and 90% respectively.

MEASUrement:STATIstics:CYCLEMode

This command turns on and off cycle to cycle measurement statistics tracking and affects computation and display of cycle-cycle statistics in the Measurement Result table. It affects measurement statistics after being enabled and after new data is acquired and measured.

Group

Measurement

Syntax

```
MEASUrement:STATIstics:CYCLEMode {OFF|ON|0|1}
MEASUrement:STATIstics:CYCLEMode?
```

Arguments

OFF turns off statistics for all measurements. This is the default value.

ON turns on statistics and displays all statistics for each measurement.

0 turns off statistics for all measurements.

1 turns on statistics and displays all statistics for each measurement.

Examples

MEASUREMENT:STATISTICS:CYCLEMODE OFF turns off statistics for all measurements.

MEASUREMENT:STATISTICS:CYCLEMODE? might return :MEASUREMENT:STATISTICS:CYCLEMODE 1 indicating that statistics are displayed for each measurement.

NEWpass (No Query Form)

This command (no query form) changes the password that enables access to password protected data. The PASSWord command must be successfully executed before using this command or an execution error will be generated.

Group Miscellaneous

Syntax `NEWpass <QString>`

Related Commands PASSWord

*PUD

Arguments `<QString>` is the new password, which can contain up to 10 characters.

Examples `NEWPASS"mypassword"` creates a new password (mypassword) for accessing your protected data.

*OPC

This command generates the operation complete message in the Standard Event Status Register (SESR) when all pending commands that generate an OPC message are complete. The *OPC? query places the ASCII character "1" into the output queue when all such OPC commands are complete. The *OPC? response is not available to read until all pending operations finish. For a complete discussion of the use of these registers and the output queue, see Registers and Queues.

The *OPC command allows you to synchronize the operation of the instrument with your application program. For more information, see Synchronization Methods.

Table 2-51: Commands that Generate an OPC Message

Command
ACQuire:STATE ON or ACQuire:STATE RUN or ACQuire:STATE 1 (when ACQuire:STOPAfter is set to SEQuence)
AUTOSet, CALibrate:INTERNal, CALibrate:INTERNal:STARt Refer to the Service Manual.
DIAg:STATE, SAVe:IMAGe, SAVe:WAVEform, CH<x>:PRObe:AUTOZero, CH<x>:PRObe:DEGAUSS
FACtory, RECAll:SETUp, *RST, TEKSecure

Group Status and Error

Syntax
```
*OPC
*OPC?
```

Related Commands BUSY?

*WAI

Examples *OPC generates the operation complete message in the SESR at the completion of all pending OPC operations.

*OPC? might return 1 to indicate that all pending OPC operations are finished.

*OPT? (Query Only)

This query-only command returns a comma separated list of installed options as an arbitrary ASCII string (no quotes) of the form:

<optionCode>:<optionDescription>,<optionCode>:<optionDescription>...

The last section of each entry (the text following the last hyphen) indicates the license type.

If no options are found, NONE is returned.

Group Status and Error

Syntax
```
*OPT?
```

Examples *OPT? response (with each option listed on a separate line for clarity):

5-BW-1000 – 1 GHz ,

5-DJA – Advanced Jitter and Eye Analysis– NODE.

PASSWord (No Query Form)

This command (no query form) enables the *PUD and NEWpass set commands. Sending PASSWord without any arguments disables these same commands. Once the password is successfully entered, the *PUD and NEWpass commands are enabled until the instrument is powered off, or until the FACtory command, the PASSWord command with no arguments, or the *RST command is issued.

To change the password, you must first enter the valid password with the PASSWord command and then change to your new password with the NEWpass command. Remember that the password is case sensitive.

Group Miscellaneous

Syntax `PASSWord <QString>`

Related Commands NEWpass

*PUD

Arguments `<QString>` is the password, which can contain up to 10 characters. The factory default password is "XYZZY" and is always valid.

Examples `PASSWORD "XYZZY"` enables the *PUD and NEWPass set commands.

`PASSWORD` disables the `*PUD` and `NEWPass` set commands. You can still use the query version of *PUD.

PAUSe (No Query Form)

This command causes the interface to pause the specified number of seconds before processing any other commands.

Group Miscellaneous

Syntax `PAUSe <NR3>`

Arguments `<NR3>` is the specified number of seconds the interface is to pause before processing any other commands. The pause time is specified as a floating point value in seconds and must be > 0.0 and ≥1800.0.

Examples `PAUSE 10.0e0;:ACQUIRE:NUMACQ` causes the interface to pause 10 seconds before returning the number of acquisitions.

PLOT:ADDNew (No Query Form)

This command adds the specified plot.

Group Plot

Syntax `PLOT:ADDNew <QString>`

Arguments `<QString>` is the specified plot. The argument is of the form "`PLOT<NR1>`", where `<NR1>` ≥ 1.

Examples `PLOT:ADDNEW "PLOT1"` adds PLOT1.

PLOT:DELete (No Query Form)

This command deletes the specified plot.

Group Plot

Syntax `PLOT:DELete <QString>`

Arguments `<QString>` is the specified plot. Argument is of the form "`PLOT<NR1>`, where `<NR1>` is ≥ 1).

Examples `PLOT:DELETE` "PLOT1" deletes PLOT1.

PLOT:LIST? (Query Only)

This command lists all currently defined plots.

Group Plot

Syntax `PLOT:LIST?`

Returns A list of all currently defined plots is returned.

Examples `PLOT:LIST?` might return `:PLOT:LIST PLOT1,PLOT3,PLOT4,PLOT5,PLOT6,PLOT7` listing all currently defined plots.

PLOT:PLOT<x>:BATHtub:BER

This command sets or queries the bathtub BER value.

Group Plot

Syntax
```
PLOT:PLOT<x>:BATHtub:BER <NR1>
PLOT:PLOT<x>:BATHtub:BER?
```

Arguments <NR1> is the bathtub BER value.

Examples PLOT:PLOT1:BATHtub:BER 16 sets the BER value to 16.

PLOT:PLOT1:BATHtub:BER? might return :PLOT:PLOT1:BATHTUB:BER 14 indicating the BER value is 14.

PLOT:PLOT<x>:BATHtub:XAXISUnits

This command sets or queries the X-Axis unit, either unit intervals or seconds.

Group Plot

Syntax
```
PLOT:PLOT<x>:BATHtub:XAXISUnits {UNITIntervals|SECOnds}
PLOT:PLOT<x>:BATHtub:XAXISUnits?
```

Arguments UNITIntervals specifies units as unit intervals.

SECOnds specifies units as seconds.

Examples PLOT:PLOT1:BATHtub:XAXISUnits SECONDS sets the units to seconds.

PLOT:PLOT1:BATHtub:XAXISUnits? might return SECONDS.

PLOT:PLOT<x>:BITType

This command sets or queries the bit type to display for the specified eye diagram plot.

Group Plot

Syntax

```
PLOT:PLOT<x>:BITType {ALLBits|TRANSition|NONTRANSition}
PLOT:PLOT<x>:BITType?
```

Arguments

PLOT<x> is the plot number.

ALLBits sets the eye diagram plot to show both transition and nontransition bits.

TRANSition sets the eye diagram plot to show only bits where a logic level transition occurs.

NONTRANSition sets the eye diagram plot to show only bits where no logic level transition occurs.

Examples

PLOT:PLOT1:BITType TRANSition sets the eye diagram in Plot 1 to only show transition bits.

PLOT:PLOT3:BITType? might return ALLBITS, indicating that the eye diagram in Plot 3 is set to show both transition and nontransition bits in the plot.

PLOT:PLOT<x>:EXPORTRaw? (Query Only)

This command returns a binary stream of double values containing the x,y and hits value. Use this command along with MEASUREMENT:ADDMEAS TIE, PLOT:PLOT1:TYPE EYEDIAGRAM, DISplay:SELect:VIEW plotview1 to export the eye diagram plot data.

Group

Plot

Syntax

```
PLOT:PLOT<x>:EXPORTRaw?
```

Related Commands

MEASUrement:ADDMEAS

DISplay:SELect:VIEW

PLOT:PLOT<x>:TYPe

Returns

A binary stream of double values containing the x,y and hits value. The data that is returned is of binary format.

Examples

PLOT:PLOT1:EXPORTRaw? might return :#73000000{øÔ$y¾„´Æ B @, indicating the x,y and hits value.

PLOT:PLOT<x>:EXTENDuis

This command sets or queries number of UIs surrounding the eye boundary UIs that are used for fast eye rendering.

Conditions Requires option DJA.

Group Plot

Syntax

```
PLOT:PLOT<x>:EXTENDuis <NR1>
PLOT:PLOT<x>:EXTENDuis?
```

Arguments

PLOT<x> is the plot number.

<NR1> is the number of surrounding UIs.

Examples

PLOT:PLOT2:EXTENDUIS 250 sets fast eye rendering to use 250 UIs in plot 2.

PLOT:PLOT2:EXTENDUIS? might return :PLOT:PLOT2:EXTENDUIS 250, indicating that 250 UIs are used for fast eye rendering in plot 2.

PLOT:PLOT<x>:EYERender

This command sets or queries the eye rendering method for the specified plot.

Conditions Requires option DJA.

Group Plot

Syntax

```
PLOT:PLOT<x>:EYERender {FAst|COMPlete}
PLOT:PLOT<x>:EYERender?
```

Arguments

PLOT<x> is the plot number.

FAst sets the eye rendering method to fast.

COMPlete sets the eye rendering method to complete.

Examples

PLOT:PLOT2:EYERENDER FAST enables fast eye rendering for plot 2.

PLOT:PLOT2:EYERENDER? might return :PLOT:PLOT2:EYERENDER FAST, indicating that fast eye rendering is enabled in plot 2.

PLOT:PLOT<x>:LABel:COLor

This command sets or queries the color of the specified trend label. This command/query only applies to Time Trend plots.

Group Plot

Syntax `PLOT:PLOT<x>:LABel:COLor <QString>`

Arguments `<QString>` is the label color. The default color is specified by a quoted empty string, and is the only available color.

Examples `PLOT:PLOT2:LABel:COLor " "` sets the plot color to the default color.

`PLOT:PLOT2:LABel:COLor?` might return `:PLOT:PLOT2:LABEL:COLOR ""` indicating the plot color is the default color.

PLOT:PLOT<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified trend label. This command/query only applies to Time Trend plots.

Group Plot

Syntax `PLOT:PLOT<x>:LABel:FONT:BOLD {<NR1>|OFF|ON}`

Arguments `<NR1>` = 0 disables bold font; any other value turns this feature on.

`OFF` disables bold font.

`ON` enables bold font.

Examples `PLOT:PLOT1:LABEL:FONT:BOLD 1` sets the label to a bold font.

`PLOT:PLOT1:LABEL:FONT:BOLD?` might return `:PLOT:PLOT1:LABEL:FONT:BOLD 1` indicating the label is bold.

PLOT:PLOT<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified trend label. This command/query only applies to Time Trend plots.

Group Plot

Syntax `PLOT:PLOT<x>:LABel:FONT:ITALic {<NR1>|OFF|ON}`

Arguments `<NR1>` = 0 disables italic font; any other value turns this feature on.

`OFF` disables italic font.

`ON` enables italic font.

Examples `PLOT:PLOT1:LABel:FONT:ITALic 1` sets the font to italic.

`PLOT:PLOT1:LABel:FONT:ITALic?` might return `:PLOT:PLOT1:LABEL:FONT:ITALIC 0` indicating the font in not italic.

PLOT:PLOT<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified trend label. This command/query only applies to Time Trend plots.

Group Plot

Syntax `PLOT:PLOT<x>:LABel:FONT:SIZE <NR1>`

Arguments `<NR1>` is the font size.

Examples `PLOT:PLOT1:LABel:FONT:SIZE 12` sets the font size to 12 points.

`PLOT:PLOT1:LABel:FONT:SIZE?` might return `:PLOT:PLOT1:LABEL:FONT:SIZE 72` indicating the font size is 72 points.

PLOT:PLOT<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified trend label, such as Arial or Times New Roman. This command/query only applies to Time Trend plots.

Group Plot

Syntax `PLOT:PLOT<x>:LABel:FONT:TYPE <QString>`

Arguments <QString> is the font type: Times New Roman, Arial, Frutiger LT Std 55 Roman, DejaVu Sans, DejaVu Sans Mono, Frutiger LT Std, Monospace, Sans Serif, Serif, Ubuntu, Ubuntu Condensed, and Ubuntu Mono.

Examples PLOT:PLOT1:LABel:FONT:TYPE Arial sets the font type to Arial.

PLOT:PLOT1:LABel:FONT:TYPE? might return :PLOT:PLOT1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman" indicating the font type is Frutiger LT Std 55 Roman.

PLOT:PLOT<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified trend label. This command/query only applies to Time Trend plots.

Group Plot

Syntax `PLOT:PLOT<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}`

Arguments <NR1> = 0 disables underline font; any other value turns this feature on.

OFF disables underline font.

ON enables underline font.

Examples `PLOT:PLOT1:LABel:FONT:UNDERline 1` set the font to underlined.

`PLOT:PLOT1:LABel:FONT:UNDERline?`
`:PLOT:PLOT1:LABEL:FONT:UNDERLINE 0` indicating the font is underlined.

PLOT:PLOT<x>:LABel:NAMe

This command sets or queries the specified trend's label. This command/query only applies to Time Trend plots.

Group Plot

Syntax `PLOT:PLOT<x>:LABel:NAMe <QString>`

Arguments <QString> is the label.

Examples `PLOT:PLOT1:LABel:NAMe "Time Trend number 1"` sets the plot name to Time Trend number 1.

`PLOT:PLOT1:LABel:NAMe?` might return `:PLOT:PLOT1:LABEL:NAME "This is a label test plot"` indicating the label name is This is a label test plot.

PLOT:PLOT<x>:LABel:XPOS

This command sets or queries the x-position of the specified trend label. This command/query only applies to Time Trend plots.

Group Plot

Syntax `PLOT:PLOT<x>:LABel:XPOS <NR3>`

Arguments `<NR3>` is the y-position, in pixels relative to the left edge of the display, of the label.

Examples `PLOT:PLOT1:LABel:XPOS 200` sets the X position to 200.

`PLOT:PLOT1:LABel:XPOS?` might return `:PLOT:PLOT1:LABEL:XPOS 45.0000` indicating the X position is 45.0 pixels to the right of the left edge of the display.

PLOT:PLOT<x>:LABel:YPOS

This command sets or queries the y-position of the specified trend label. This command/query only applies to Time Trend plots.

Group Plot

Syntax `PLOT:PLOT<x>:LABel:YPOS <NR3>`

Arguments `<NR3>` is the x-position, in pixels relative to the baseline of the waveform, of the label.

Examples `PLOT:PLOT1:LABel:YPOS 100` sets the Y position to 100.

`PLOT:PLOT1:LABel:YPOS?` might return `:PLOT:PLOT1:LABEL:YPOS 0.0E+0` indicating the Y position of the label is at the baseline of the waveform.

PLOT:PLOT<x>:MASK? (Query Only)

This command returns the name of the mask test associated with the specified eye diagram plot.

Group Plot

Syntax `PLOT:PLOT<x>:MASK?`

Arguments `PLOT<x>` is the plot number.

Returns `<String>` is a quoted string that contains the name of the eye diagram mask test associated with the specified plot. If a plot does not have an eye diagram mask test, the command returns an empty string "".

Examples `PLOT:PLOT1:MASK?` might return `"mask2"`, indicating that the eye diagram mask test mask2 is associated with plot 1.

PLOT:PLOT<x>:NUMBins

This command sets or queries the current histogram resolution.

Group Plot

Syntax `PLOT:PLOT<x>:NUMBins {TWENtyfive|FIFTY|HUNdred|TWOFifty|FIVEHundred|TWOThousand|MAXimum}`

Arguments Arguments are the number of bins.

Examples `PLOT:PLOT4:NUMBins TWENTYFIVE` sets the number of bins to 25.

`PLOT:PLOT4:NUMBins?` might return `:PLOT:PLOT4:NUMBINS TWOFIFTY` indicating the histogram resolution is 250 bins.

PLOT:PLOT<x>:RAILNUM

Sets the DPM histogram source.

Conditions Requires option 5-DPM (5 Series MSO instruments) or 6-DPM (6 Series MSO instrument)

Group DPM

Syntax
```
PLOT:PLOT<x>:RAILNUM RAIL[1–7]
PLOT:PLOT<x>:RAILNUM?
```

Arguments `PLOT<x>` is the plot number.

`Rail[1–7]` is the rail number.

Examples `PLOT:PLOT2:RAILNUM RAIL6` sets the DPM histogram source to Rail 6 of plot 2.

`PLOT:PLOT5:RAILNUM?` might return `RAIL1`, indicating that the DPM histogram source is Rail 1 of plot 5.

PLOT:PLOT<x>:SOUrce<x>

This command sets or queries the plot source.

Group Plot

Syntax
```
PLOT:PLOT<x>:SOUrce<x> MEAS<x>
PLOT:PLOT<x>:SOUrce<x>?
```

Arguments `MEAS<x>` is the specified measurement source for the specified plot.

Examples `PLOT:PLOT1:SOUrce1 MEAS2` sets source 1 of plot 1 to measurement 2.

`PLOT:PLOT1:SOUrce1?` might return `:PLOT:PLOT1:SOURCE1 MEAS1` indicating the specified source of the specified plot is measurement 1.

PLOT:PLOT<x>:SPECtrum:BASE

This command sets or queries the spectrum base. Undefined for non-spectrum plots.

Group Plot

Syntax

```
PLOT:PLOT<x>:SPECtrum:BASE <NR1>
PLOT:PLOT<x>:SPECtrum:BASE?
```

Arguments

<NR1> is the spectrum base.

Examples

PLOT:PLOT3:SPECtrum:BASE -10 sets the base to -10.

PLOT:PLOT3:SPECtrum:BASE? might return :PLOT:PLOT3:SPECTRUM:BASE -15 indicating the base is -15.

PLOT:PLOT<x>:SPECtrum:DYNRange

This command sets or queries the dynamic range value.

Group

Plot

Syntax

```
PLOT:PLOT<x>:SPECtrum:DYNRange <NR3>
PLOT:PLOT<x>:SPECtrum:DYNRange?
```

Arguments

<NR3> is the dynamic range value.

Examples

PLOT:PLOT3:SPECtrum:DYNRange 150 sets the dynamic range to 150 dB.

PLOT:PLOT3:SPECtrum:DYNRange? might return :PLOT:PLOT3:SPECTRUM:DYNRANGE 100 indicating the dynamic range is 100 dB.

PLOT:PLOT<x>:TYPe

This command sets or returns the current plot type of the specified plot.

Group

Plot

Syntax

```
PLOT:PLOT<x>:TYPe
{NONE|BATHTUB|EYEDIAGRAM|HARMONICS|HISTOGRAM|
INDUCTANCE|IVSINTEGRALV|MAGPROPERTY|PHASENOISE|PHASOR
|SOA|SPECTRUM|SSCPROFILE|SWL|TIEHISTOGRAM|TIETIMETREND|
TIESPECTRUM|TIMETREND|XY|XYZ}
```

Arguments

<x> is the plot number. This is the equivalent of the number shown on a plot heading in the UI.

NONE does not create a plot.

BATHTUB creates a bathtub plot.

EYEDIAGRAM creates an eye diagram. This plot type is not available on a 4 Series MSO instrument.

HARMONICS creates a harmonics bar graph

HISTOGRAM creates a histogram plot.

INDUCTANCE creates a inductance plot

IVSINTEGRALV creates a I vs. ∫V plot

PHASENOISE creates a phase noise plot.

PHASOR creates the Phasor Diagram.

MAGPROPERTY creates a BH curve

SOA creates an SOA plot

SPECTRUM creates a spectrum plot.

SSCPROFILE creates a SSC profile plot.

SWL creates a Switching Loss plot

TIEHISTOGRAM creates a TIE histogram plot.

TIESPECTRUM creates a TIE spectrum plot.

TIETIMETREND creates a TIE time trend plot.

TIMETREND creates a time trend plot.

XY creates a XY plot.

XYZ creates a XYZ plot.

Examples

PLOT:PLOT2:TYPe BATHTUB creates a bathtub plot.

PLOT:PLOT2:TYPe? might return :PLOT:PLOT2:TYPE EYEDIAGRAM indicating the plot is an eye diagram.

POWer:ADDNew (No Query Form)

This command adds the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:ADDNew “POWER<x>”`

Examples `POWER:ADDNEW “POWER1”` adds POWER1 measurement badge and selects Switching Loss measurement by default.

POWer:DELete (No Query Form)

This command deletes the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:DELete "POWER<x>"`

Examples `POWER:DELETE "POWER1"` deletes the POWER1 measurement badge.

POWer:POWer<x>:AUTOSet (No Query Form)

This command executes power autoset for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:AUTOSet EXECute`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

Examples `POWer:POWer1:AUTOSet EXECute` executes the power autoset for the power measurement Power 1.

POWer:POWer<x>:CLRESPONSE:AMP[1-10]Val

This command sets or queries the generator amplitude value of the specified configuration step for the Control Loop Response power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:AMP[1-10]Val <NR3>
POWer:POWer<x>:CLRESPONSE:AMP[1-10]Val?
```

Arguments `POWer<x>` sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`AMP[1–10]` sets the configuration step number, in the range of 1 to 10. Values outside this range will report an error.

`<NR3>` sets the generator amplitude for the specified configuration step, in the range of –100 V to 100 V.

Examples `POWer:POWer1:CLRESPONSE:AMP3Val 20` sets the generator output amplitude for configuration step 3 to 20 volts, for power measurement 1.

`POWer:POWer2:CLRESPONSE:AMP8Val?` might return `60`, indicating that the generator output amplitude setting of configuration step 8 is 60 volts, for power measurement 2.

POWer:POWer<x>:CLRESPONSE:AMPMode

This command sets or queries the amplitude mode for the Control Loop Response power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:AMPMode {CONSTant|PROFile}
POWer:POWer<x>:CLRESPONSE:AMPMode?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CONSTant` sets the amplitude mode to output a constant amplitude signal from the DUT stimulus generator for all frequency bands.

`PROFile` enables configuring the generator to set amplitude values for each frequency band.

Examples

`POWer:POWer1:CLRESPONSE:AMPMode CONSTant` sets the amplitude mode to Constant for power measurement 1.

`POWer:POWer2:CLRESPONSE:AMPMode?` might return `PROFile`, indicating that the amplitude mode power measurement 2 is set to Profile.

POWer:POWer<x>:CLRESPONSE:CONNECTSTATus? (Query Only)

Queries connection status to the external generator used with the specified Control Loop Response power measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:CONNECTSTATus?
```

Related Commands

POWer:POWer<x>:CLRESPONSE:GENIPADDress

POWer:POWer<x>:CLRESPONSE:GENerator

Arguments

`POWer<x>` is the number of the power measurement.

Returns

`SUCCess` indicates the instrument recognizes the connection to the external generator.

`FAILure` indicates the instrument could not make the connection to the external generator.

`UNSUPPorted` indicates that the connected generator is not recognized.

Examples `POWer:POWer<x>:CLRESPONSE:CONNECTSTATus?` might return `FAILURE`, indicating that the instrument could not connect to the external generator at the specified IP address.

POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude

This command sets or queries the constant amplitude voltage for the Control Loop Response power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude <NR3>
POWer:POWer<x>:CLRESPONSE:CONSTAMPlitude?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the constant amplitude voltage value for the measurement, in the range of –100 V to 100 V.

Examples `POWer:POWer3:CLRESPONSE:CONSTAMPlitude 120` sets the constant amplitude voltage for Power measurement 3 to be 120 volts.

`POWer:POWer5:CLRESPONSE:CONSTAMPlitude?` might return `-15`, indicating that the constant amplitude voltage for power measurement 5 is –15 volts.

POWer:POWer<x>:CLRESPONSE:FREQ[1-11]Val

This command sets or queries the generator frequency value of the specified configuration step for the Control Loop Response power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:FREQ[1-11]Val <NR3>
POWer:POWer<x>:CLRESPONSE:FREQ[1-11]Val?
```

Arguments

`POWer<x>` sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`FREQ[1–11]` sets the configuration step number, in the range of 1 to 11. Values outside this range will report an error.

`<NR3>` sets the frequency of the specified configuration step number, in the range of 10 Hz to 50 MHz.

Examples

`POWer:POWer5:CLRESPONSE:FREQ1Val 200` sets the generator frequency value for frequency band 1 to 200 Hz, for power measurement 5.

`POWer:POWer2:CLRESPONSE:FREQ3Val?` might return `2.000E+6`, indicating that the generator frequency output for frequency band 3 is 2.0 MHz, for power measurement 2.

POWer:POWer<x>:CLRESPONSE:GENerator

This command sets or queries the generator source used to send stimulus signals to the DUT, for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:GENerator {INTernal}
POWer:POWer<x>:CLRESPONSE:GENerator?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`INTernal` sets the generator to the instrument AFG. This is the only valid argument.

Examples

`POWer:POWer3:CLRESPONSE:GENerator INTernal` sets the generator to the instrument AFG for power measurement 3.

`POWer:POWer2:CLRESPONSE:GENerator?` might return `INTERNAL`, indicating that the generator source for power measurement 2 is the instrument AFG.

POWer:POWer<x>:CLRESPONSE:GENerator

Sets or queries the generator source for the specified Control Loop Response power measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:GENerator {INTernal|EXTernal}
POWer:POWer<x>:CLRESPONSE:GENerator?
```

Arguments `POWer<x>` is the number of the power measurement.`INTernal` sets the internal generator as the source for the Control Loop Response power measurement.

`EXTernal` sets the external generator as the source for the Control Loop Response power measurement.

Examples `POWer:POWer1:CLRESPONSE:GENerator Internal` sets the internal generator as the source for the Control Loop Response measurement.

`POWer:POWer1:CLRESPONSE:GENerator?` might return `EXTERNAL`, indicating that the Control Loop Response measurement is set to use an external generator.

POWer:POWer<x>:CLRESPONSE:GENIPADDress

Sets or queries the IP address of the external generator to be used with the specified Control Loop Response measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:GENIPADDress <NR2>
POWer:POWer<x>:CLRESPONSE:GENIPADDress?
```

Arguments `POWer<x>` is the number of the power measurement.`<NR2>` is the IP address of the generator.

Examples

`POWer:POWer4:CLRESPONSE:GENIPADDress 127.1.1.1` sets the IP address of the external generator to 127.1.1.1.

`POWer:POWer1:CLRESPONSE:GENIPADDress?` might return "", indicating that there is no address set for the external generator associated with Control Loop Response power measurement 4.

POWer:POWer<x>:CLRESPONSE:IMPEDance

This command sets or queries the vertical termination impedance for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:IMPEDance {FIFTy|HIGHZ}
POWer:POWer<x>:CLRESPONSE:IMPEDance?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`FIFTy` sets the impedance to be 50 Ω.

`HIGHZ` sets the impedance to be 1 MΩ.

Examples

`POWer:POWer3:CLRESPONSE:IMPEDance FIFTy` sets the vertical termination impedance for power measurement 3 to be 50 Ω.

`POWer:POWer2:CLRESPONSE:IMPEDance?` might return `HIGHZ`, indicating that the vertical termination impedance for power measurement 2 is 1 MΩ.

POWer:POWer<x>:CLRESPONSE:INPUTSOurce

This command sets or queries the input source for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:INPUTSOurce CH<x>
POWer:POWer<x>:CLRESPONSE:INPUTSOurce?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> sets the channel to use for the output signal source.

Examples

POWer:POWer2:CLRESPONSE:INPUTSOurce CH8 sets the input for power measurement 2 to be the channel 8 waveform.

POWer:POWer2:CLRESPONSE:INPUTSOurce? might return CH1, indicating that the input source for power measurement 1 is channel 1.

POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce

This command sets or queries the output source for the Control Loop Response power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce CH<x>
POWer:POWer<x>:CLRESPONSE:OUTPUTSOurce?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> sets the channel to use for the output signal source.

Examples

POWer:POWer1:CLRESPONSE:OUTPUTSOurce CH5 sets the output source for power measurement 1 to be the channel 5 waveform.

POWer:POWer3:CLRESPONSE:OUTPUTSOurce? might return CH6, indicating that the output source for power measurement 3 is channel 6.

POWer:POWer<x>:CLRESPONSE:PPD

This command sets or queries the points per decade (PPD) value for the Control Loop Response power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax
```
POWer:POWer<x>:CLRESPONSE:PPD <NR3>
POWer:POWer<x>:CLRESPONSE:PPD?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the PPD value for the measurement, in the range of 10 to 100 points.

Examples `POWer:POWer3:CLRESPONSE:PPD 30` sets the PPD for Power measurement 3 to be 30 points.

`POWer:POWer5:CLRESPONSE:PPD?` might return `20`, indicating that the PPD value for power measurement 5 is 20 points.

POWer:POWer<x>:CLRESPONSE:STARTFREQuency

This command sets or queries the start frequency value for the Control Loop Response power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax
```
POWer:POWer<x>:CLRESPONSE:STARTFREQuency <NR3>
POWer:POWer<x>:CLRESPONSE:STARTFREQuency?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the starting frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples `POWer:POWer8:CLRESPONSE:STARTFREQuency 12` sets the starting frequency for power measurement 8 to be 12 Hz.

`POWer:POWer3:CLRESPONSE:STARTFREQuency?` might return `2400`, indicating that the starting frequency for power measurement 3 is 2400 Hz.

POWer:POWer<x>:CLRESPONSE:STOPFREQuency

This command sets or queries the stop frequency value for the Control Loop Response power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:STOPFREQuency <NR3>
POWer:POWer<x>:CLRESPONSE:STOPFREQuency?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the stop frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples

`POWer:POWer3:CLRESPONSE:STOPFREQuency 120` sets the stop frequency for the control loop response power measurement 8 to be 120 Hz.

`POWer:POWer5:CLRESPONSE:STOPFREQuency?` might return `2000`, indicating that the stop frequency for the control loop response power measurement 5 is 2000 Hz.

POWer:POWer<x>:CLRESPONSE:TESTCONNection (No Query Form)

This command tests the connection to the external generator used with the specified Control Loop Response power measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group Power

Syntax

```
POWer:POWer<x>:CLRESPONSE:TESTCONNection EXECute
```

Related Commands POWer:POWer<x>:CLRESPONSE:CONNECTSTATus?

POWer:POWer<x>:CLRESPONSE:GENIPADDress

Arguments POWer<x> is the number of the PSRR power measurement.

EXECute runs the test connection function.

Examples POWer:POWer5:CLRESPONSE:TESTCONNection EXECute runs the test connection function for Control Loop Response power measurement 5.

POWer:POWer<x>:CYCLEAmp:INPUTSOurce

This command sets or queries the input source for cycle amplitude measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:CYCLEAmp:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEAmp:INPUTSOurce?
```

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples POWer:POWer1:CYCAmp:INPUTSOurce CH1 sets the input source for cycle amplitude measurement as CH1 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLEBase:INPUTSOurce

This command sets or queries the input source for cycle base measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:CYCLEBase:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEBase:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:CYCLEBase:INPUTSOurce CH2 sets the input source for cycle base measurement as CH2 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLEMAX:INPUTSOurce

This command sets or queries the input source for cycle maximum measurement in the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:CYCLEMAX:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEMAX:INPUTSOurce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:CYCMAX:INPUTSOurce CH2 sets the input source for cycle maximum measurement as CH2 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLEMin:INPUTSOurce

This command sets or queries the input source for cycle minimum measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:CYCLEMin:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEMin:INPUTSOurce?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:CYCMin:INPUTSOurce CH2` sets the input source for cycle minimum measurement as CH2 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLEPKPK:INPUTSOurce

This command sets or queries the input source for cycle peak-to-peak measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:CYCLEPKPK:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLEPKPK:INPUTSOurce?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples POWer:POWer1:CYCPKPK:INPUTSOurce REF1 sets the input source for cycle peak-to-peak measurement as REF1 for the power measurement badge Power 1.

POWer:POWer<x>:CYCLETop:INPUTSOurce

This command sets or queries the input source for cycle top measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:CYCLETop:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:CYCLETop:INPUTSOurce?
```

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples POWer:POWer1:CYCTop:INPUTSOurce REF3 sets the input source for cycle top measurement as REF3 for the power measurement badge Power 1.

POWer:POWer<x>:DIDT:INPUTSOurce

This command sets or queries the input source for di/dt measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:DIDT:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:DIDT:INPUTSOurce?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:DIDT:INPUTSOurce CH6` sets the input source for di/dt measurement as channel 2.

POWer:POWer<x>:DIDT:SOURCEEDGEType

This command sets or queries the edge type for di/dt measurement of the specified power measurement number. <x> specifies the number of the power measurement badge.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:DIDT:SOURCEEDGEType {RISE|FALL}
POWer:POWer<x>:DIDT:SOURCEEDGEType?
```

Examples

`POWer:POWer1:DIDT:SOURCEEDGEType RISE` sets the edge type as rise for the di/dt measurement.

POWer:POWer<x>:DVDT:INPUTSOurce

This command sets or queries the input source for dv/dt measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:DVDT:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:DVDT:INPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:DVDT:INPUTSOurce CH1` sets the input source for dv/dt measurement as channel 1.

POWer:POWer<x>:DVDT:SOURCEEDGEType

This command sets or queries the edge type for dv/dt measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:DVDT:SOURCEEDGEType {RISE|FALL}
POWer:POWer<x>:DVDT:SOURCEEDGEType?
```

Examples

`POWer:POWer1:DVDT:SOURCEEDGEType RISE` sets the edge type as rise for the dv/dt measurement.

POWer:POWer<x>:EFFICIENCY:INPUTType

This command sets or queries the input type (AC or DC) for power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:INPUTType {AC|DC}
POWer:POWer<x>:EFFICIENCY:INPUTType?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:OUTPUT1Type

Arguments

`Power<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

`AC` sets the input voltage type to AC.

`DC` sets the input voltage type to DC.

Examples

`POWer:POWer1:EFFICIENCY:INPUTType DC` sets the input voltage type to DC for power Efficiency measurement 1

`POWer:POWer2:EFFICIENCY:INPUTType?` might return `AC`, indicating that the input voltage type for power Efficiency measurement 2 is AC.

POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce

This command sets or queries the output 1 current source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce

POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce

Arguments

`Power<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:EFFICIENCY:IOUT1SOUrce CH4` sets the source for the output 1 current efficiency measurement of power measurement 1 to Channel 4.

`POWer:POWer1:EFFICIENCY:IOUT1SOUrce?` might return `CH6`, indicating that the source for the output 1 current efficiency measurement of power measurement 1 is Channel 6.

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce

This command sets or queries the output 2 current source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce

POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce

Arguments

`Power<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer3:EFFICIENCY:IOUT2SOUrce REF2` sets the source for the output 2 current efficiency measurement of power measurement 3 to Reference waveform 2.

`POWer:POWer1:EFFICIENCY:IOUT2SOUrce?` might return `MATH1`, indicating that the source for the output 2 current efficiency measurement of power measurement 1 is Math waveform 1.

POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce

This command sets or queries the output 3 current source for the power Efficiency measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce?
```

Related Commands POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce

Arguments `Power<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:EFFICIENCY:IOUT3SOUrce REF2` sets the source for the output 3 current efficiency measurement of power measurement 1 to Reference waveform 2.

`POWer:POWer4:EFFICIENCY:IOUT3SOUrce?` might return `CH8`, indicating that the source for the output 3 current efficiency measurement of power measurement 4 is Channel 8.

POWer:POWer<x>:EFFICIENCY:ISOUrce

This command sets or queries the current source for the power Efficiency measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:ISOUrce
POWer:POWer<x>:EFFICIENCY:ISOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:VSOUrce

Arguments

`Power<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:EFFICIENCY:ISOUrce CH2` sets the input current source for power Efficiency measurement 1 to Channel 2.

`POWer:POWer6:EFFICIENCY:ISOUrce?` might return `CH2`, indicating that the input current source for power Efficiency measurement 6 is Channel 2.

POWer:POWer<x>:EFFICIENCY:NUMOFOutputs

This command sets or queries the number of outputs for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:NUMOFOutputs {ONE|TWO|THREE}
POWer:POWer<x>:EFFICIENCY:NUMOFOutputs?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:IOUT1SOUrce

POWer:POWer<x>:EFFICIENCY:IOUT2SOUrce

POWer:POWer<x>:EFFICIENCY:IOUT3SOUrce

POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce

POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce

POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce

Arguments

`Power<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`ONE`, `TWO`, `THREE` sets the number of outputs to test in the power efficiency measurement.

Examples

`POWer:POWer4:EFFICIENCY:NUMOFOutputs TWO` sets the number of outputs to measure in power measurement 4 to two.

`POWer:POWer3:EFFICIENCY:NUMOFOutputs?` might return `ONE`, indicating that the number of outputs being measured in power measurement 3 is one.

POWer:POWer<x>:EFFICIENCY:OUTPUT1Type

This command sets or queries the Output1 type (AC or DC) for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:OUTPUT1Type {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUT1Type?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:INPUTType

Arguments

`Power<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

`AC` sets the output1 voltage type to AC.

`DC` sets the output1 voltage type to DC.

Examples

`POWer:POWer1:EFFICIENCY:OUTPUT1Type DC` sets the output type to DC for power Efficiency measurement 1.

`POWer:POWer3:EFFICIENCY:OUTPUT1Type?` might return `AC`, indicating that the power output type for power Efficiency measurement 3 is AC.

POWer:POWer<x>:EFFICIENCY:OUTPUT2Type

This command sets or queries the Output2 type (AC or DC) for the power Efficiency measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:EFFICIENCY:OUTPUT2Type {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUT2Type?
```

Related Commands POWer:POWer<x>:EFFICIENCY:INPUTType

Arguments `POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

`AC` sets the Output2 voltage type to AC.

`DC` sets the Output2 voltage type to DC.

Examples `POWer:POWer1:EFFICIENCY:OUTPUT2Type DC` sets the Output2 type to DC for power Efficiency measurement 1.

`POWer:POWer3:EFFICIENCY:OUTPUT2Type?` might return `AC`, indicating that the power Output2 type for power Efficiency measurement 3 is AC.

POWer:POWer<x>:EFFICIENCY:OUTPUT3Type

This command sets or queries the Output3 type (AC or DC) for the power Efficiency measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:EFFICIENCY:OUTPUT3Type {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUT3Type?
```

Related Commands POWer:POWer<x>:EFFICIENCY:INPUTType

Arguments `POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

`AC` sets the Output3 voltage type to AC.

DC sets the Output3 voltage type to DC.

Examples

POWer:POWer1:EFFICIENCY:OUTPUT3Type DC sets the Output3 type to DC for power Efficiency measurement 1.

POWer:POWer3:EFFICIENCY:OUTPUT3Type? might return AC, indicating that the power Output3 type for power Efficiency measurement 3 is AC.

POWer:POWer<x>:EFFICIENCY:OUTPUTType

This command sets or queries the output type (AC or DC) for power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:OUTPUTType {AC|DC}
POWer:POWer<x>:EFFICIENCY:OUTPUTType?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:INPUTType

Arguments

Power<x> is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge of the UI.

AC sets the output voltage type to AC.

DC sets the output voltage type to DC.

Examples

POWer:POWer1:EFFICIENCY:OUTPUTType DC sets the output type to DC for power Efficiency measurement 1

POWer:POWer3:EFFICIENCY:OUTPUTType? might return AC, indicating that the power output type for power Efficiency measurement 3 is AC.

POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce

This command sets or queries the output 1 voltage source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce?
```

Related Commands POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce

POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce

Arguments `POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer4:EFFICIENCY:VOUT1SOUrce CH4` sets the source for the output 1 voltage efficiency measurement of power measurement 4 to Channel 4.

`POWer:POWer1:EFFICIENCY:VOUT1SOUrce?` might return `CH3`, indicating that the source for the output 1 voltage efficiency measurement of power measurement 1 is Channel 3.

POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce

This command sets or queries the output 2 voltage source for the power Efficiency measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce?
```

Related Commands POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce

POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer4:EFFICIENCY:VOUT2SOUrce CH5` sets the source for the output 2 voltage efficiency measurement of power measurement 4 to Channel 5.

`POWer:POWer1:EFFICIENCY:VOUT2SOUrce?` might return `CH3`, indicating that the source for the output 2 voltage efficiency measurement of power measurement 1 is Channel 3.

POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce

This command sets or queries the output 3 voltage source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:EFFICIENCY:VOUT3SOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:VOUT1SOUrce

POWer:POWer<x>:EFFICIENCY:VOUT2SOUrce

Arguments

`POWer<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown on the power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer4:EFFICIENCY:VOUT3SOUrce REF7` sets the source for the output 3 voltage efficiency measurement of power measurement 4 to Reference waveform 7.

`POWer:POWer1:EFFICIENCY:VOUT3SOUrce?` might return `CH5`, indicating that the source for the output 3 voltage efficiency measurement of power measurement 1 is Channel 5.

POWer:POWer<x>:EFFICIENCY:VSOUrce

This command sets or queries the voltage source for the power Efficiency measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:EFFICIENCY:VSOUrce
POWer:POWer<x>:EFFICIENCY:VSOUrce?
```

Related Commands

POWer:POWer<x>:EFFICIENCY:ISOUrce

Arguments

`Power<x>` is the number of a power efficiency measurement. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer2:EFFICIENCY:VSOUrce CH4` sets the input voltage source for power Efficiency measurement 2 to Channel 4.

`POWer:POWer1:EFFICIENCY:VSOUrce?` might return `CH3`, indicating that the input voltage source for power Efficiency measurement 1 is Channel 3.

POWer:POWer<x>:FREQUENCY:EDGe

This command sets or queries the edge type for frequency measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:FREQUENCY:EDGe {RISE|FALL}
POWer:POWer<x>:FREQUENCY:EDGe?
```

Related Commands POWer:POWer<x>:FREQUENCY:INPUTSOurce

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

Examples `POWer:POWer1:FREQUENCY:EDGe FALL` sets the edge type as fall for the frequency measurement.

POWer:POWer<x>:FREQUENCY:INPUTSOurce

This command sets or queries the input source for frequency measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:FREQUENCY:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:FREQUENCY:INPUTSOurce?
```

Related Commands POWer:POWer<x>:FREQUENCY:EDGe

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:FREQUENCY:INPUTSOurce REF1` sets the input source for frequency measurement as REF1.

POWer:POWer<x>:GATing

This command sets or queries the gating type for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:GATing {NONE|CURSOR|SCREEN|LOGIC}
POWer:POWer<x>:GATing?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`NONE` makes measurement across the entire waveform record.

`CURSOR` makes measurements on that portion of the waveform between the cursors. Selecting Cursors opens cursors on the measurement source. Set the cursors so that the waveform area of interest is in between the cursors.

`SCREEN` takes measurements on that portion of the waveform shown in the display. When Zoom is on, the display is the zoom window.

`LOGIC` takes measurements only when the logical state of a specified waveform is true.

Examples `POWer:POWer1:GATing CURSOR` sets the gating type for the power measurement badge 1 as Cursor.

`POWer:POWer2:GATing?` might return SCREEN indicating the configured gating type for the power measurement badge 2.

POWer:POWer<x>:GATing:GLOBal

This command sets or queries the gating settings for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:GATing:GLOBal {ON|OFF|1|0}
POWer:POWer<x>:GATing:GLOBal?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`1` | `ON` selects the gating settings as Global.

`0` | `OFF` selects the gating settings as Local.

Examples `POWer:POWer1:GATing:GLOBal 1` selects the gating settings as Global for the power measurement badge 1.

POWer:POWer<x>:HARMONICS:CLASs

This command sets or queries the class type for the harmonics measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:HARMONICS:CLASs {CLASSA | CLASSB | CLASSC | CLASSD}
POWer:POWer<x>:HARMONICS:CLASs?
```

Related Commands POWer:POWer<x>:HARMONICS:STANDard

Examples `POWer:POWer1:HARMONICS:CLASs CLASSB` sets the class type for harmonics measurement as CLASSB.

POWer:POWer<x>:HARMONICS:CMEThod

This command sets or queries the fundamental current method for the harmonics measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:HARMONICS:CMEThod {RATed | MEASured}
POWer:POWer<x>:HARMONICS:CMEThod?
```

Related Commands POWer:POWer<x>:HARMONICS:STANDard

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`RATed`: select to use the standard input current values in the measurement.

`MEASured`: select to use the measured the input current values in the measurement.

Examples `POWer:POWer1:HARMONICS:CMEThod RATed` sets the current method for harmonics measurement as rated.

POWer:POWer<x>:HARMONICS:FUNDCURRent

This command sets or queries the fundamental current value for the harmonics measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:HARMONICS:FUNDCURRent <NR1>
POWer:POWer<x>:HARMONICS:FUNDCURRent?
```

Related Commands POWer:POWer<x>:HARMONICS:CLASs

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 0 to 16.

Examples `POWer:POWer<x>:HARMONICS:FUNDCURRent 1.5` sets the fundamental current for the harmonics measurement as 1.5.

POWer:POWer<x>:HARMONICS:HORDer

This command sets or queries the order value for the harmonics measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:HARMONICS:HORDer <NR1>
POWer:POWer<x>:HARMONICS:HORDer?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 40 to 100.

Examples `POWer:POWer1:HARMONICS:HORDer 70` sets the order value for the harmonics measurement as 70.

POWer:POWer<x>:HARMONICS:HSOURce

This command sets or queries the source type for the harmonics measurement of the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:HARMONICS:HSOURce {CURRent | VOLTage}
POWer:POWer<x>:HARMONICS:HSOURce?
```

Related Commands POWer:POWer<x>:HARMONICS:STANDard

Examples `POWer:POWer1:HARMONICS:HSOURce CURRent` sets the source as current for the harmonics measurement.

POWer:POWer<x>:HARMONICS:IPOWer

This command sets or queries the input power value for the harmonics measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:HARMONICS:IPOWer <NR1>
POWer:POWer<x>:HARMONICS:IPOWer?
```

Related Commands POWer:POWer<x>:HARMONICS:CLASs

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 0 to 600.

Examples `POWer:POWer<x>:HARMONICS:IPOWer 150` sets the input power for the harmonics measurement as 150.

POWer:POWer<x>:HARMONICS:ISOURce

This command sets or queries the current source for SOA measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:HARMONICS:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:HARMONICS:ISOURce?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples POWer:POWer1:HARMONICS:ISOURce CH2 sets the current source for harmonics measurement as channel 2.

POWer:POWer<x>:HARMONICS:LINEFREQUEncy

This command sets or queries the value for the line frequency for the Harmonics measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:HARMONICS:LINEFREQUEncy {Auto|FIFTyhz|SIXTyhz
|THREESIXTyhz|FOURHUNDREdhz|SIXFIFTyhz|EIGHTHUNDREdhz|CUSTom
POWer:POWer<x>:HARMONICS:LINEFREQUEncy?
```

Related Commands POWer:POWer<x>:HARMONICS:LINEFREQUEncy

Arguments

POWer<x> is the Power measurement identifier number.

Auto automatically detects and sets the line frequency value.

FIFTyhz sets the line frequency value to 50 Hz.

SIXTyhz sets the line frequency value to 60 Hz.

THREESIXTyhz sets the line frequency value to 360 Hz.

FOURHUNDREdhz sets the line frequency value to 400 Hz.

SIXFIFTyhz sets the line frequency value to 650 Hz.

EIGHTHUNDREdhz sets the line frequency value to 800 Hz.

CUSTom sets the line frequency value to Custom. The default value for custom is 100 Hz. Use the POWer:POWer<x>:HARMONICS:LINEFREQUEncy command to set a custom line frequency value.

Examples POWer:POWer3:HARMONICS:LINEFREQUEncy FIFTyhz sets the line frequency value for the Harmonics power measurement 3 to 50 Hz.

POWer:POWer1:HARMONICS:LINEFREQUEncy? might return POWer:POWer1:HARMONICS:LINEFREQUEncy CUSTom, indicating that the line frequency setting for Harmonics power measurement 1 is a custom value.

POWer:POWer<x>:HARMONICS:ODDEVen

This command sets or queries the harmonics value analysis format of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:HARMONICS:ODDEVen {ALL | ODD | EVEN}
POWer:POWer<x>:HARMONICS:ODDEVen?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

ALL to display all harmonics values

ODD to display only the odd values of harmonics

EVEN to display only the even values of harmonics

Examples POWer:POWer3:HARMONICS:ODDEVen ALL sets the harmonics value analysis format to display all result values for harmonics measurement 3.

POWer:POWer<x>:HARMONICS:PFACtor

This command sets or queries the value of power factor for the harmonics measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:HARMONICS:PFACtor <NR1>
POWer:POWer<x>:HARMONICS:PFACtor?
```

Related Commands POWer:POWer<x>:HARMONICS:CLASs

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 0 to 1.

Examples `POWer:POWer2:HARMONICS:PFACtor 0.7` sets the power factor as 0.7 for harmonics measurement 2.

POWer:POWer<x>:HARMONICS:POWERRating

This command sets or queries the power level for the harmonics measurement of the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:HARMONICS:POWERRating {HIGH | LOW}
POWer:POWer<x>:HARMONICS:POWERRating?
```

Related Commands POWer:POWer<x>:HARMONICS:STANDard

Examples `POWer:POWer1:HARMONICS:POWERRating HIGH` sets the power level as high for harmonics measurement 1.

POWer:POWer<x>:HARMONICS:RCURRent

This command sets or queries the rated current for the harmonics measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:HARMONICS:RCURRent <NR1>
POWer:POWer<x>:HARMONICS:RCURRent?
```

Related Commands

POWer:POWer<x>:HARMONICS:CMEThod

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR1>` ranges from 0 to 100

Examples

`POWer:POWer2:HARMONICS:RCURRent 1.5` sets the rated current as 1.5 for harmonics measurement 2.

POWer:POWer<x>:HARMONICS:STANDard

This command sets or queries the test mode for harmonics measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:HARMONICS:STANDard {NONe | IEC | MIL | AM14}
POWer:POWer<x>:HARMONICS:STANDard?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`NONe` = No standard

`IEC` = IEC 61000-3-2 standard

`MIL` = MIL-STD-1399 standard

`AM14` = AM14 standard

Examples

`POWer:POWer3:HARMONICS:STANDard AM14` sets the test mode for harmonics measurement as AM14 for harmonics measurement 3.

POWer:POWer<x>:HARMONICS:STARTFREQUEncy

This command sets or queries the value for the start frequency for the Harmonics measurement. in the range of 1 Hz to 1 GHz.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:HARMONICS:STARTFREQUEncy <NR2>
POWer:POWer<x>:HARMONICS:STARTFREQUEncy?
```

Arguments `POWer<x>` is the Power measurement.

`<NR2>` sets the starting frequency, in hertz.

Examples `POWer:POWer1:HARMONICS:STARTFREQUEncy 1000` sets the Harmonics measurement start frequency to 1 kHz for power measurement 1.

`POWer:POWer4:HARMONICS:STARTFREQUEncy?` might return `POWer:POWer4:HARMONICS:STARTFREQUEncy 60`. indicating that the start frequency for Harmonics power measurement 4 is 60 Hz.

POWer:POWer<x>:HARMONICS:UNITs

This command sets or queries the harmonics results units of the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:HARMONICS:UNITs {LOG | LINear}
POWer:POWer<x>:HARMONICS:UNITs?
```

Examples `POWer:POWer1:HARMONICS:UNITs LOG` sets the harmonics results units as logarithmic for harmonics measurement 1.

POWer:POWer<x>:HARMONICS:VSOURce

This command sets or queries the voltage source for SOA measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:HARMONICS:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:HARMONICS:VSOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer4:HARMONICS:VSOURce CH1` sets the voltage source for harmonics measurement 4 as channel 1.

POWer:POWer<x>:IMPEDANCE:AMP[1-10]Val

Sets or queries the signal generator amplitude setting of the specified profile step, for the specified Impedance measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:AMP[1-10]Val <NR3>
POWer:POWer<x>:IMPEDANCE:AMP[1-10]Val?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`AMP[1–10]` specifies the number of the profile step.

<NR3> sets the generator output amplitude for the specified profile step, in the range –100 V to 100 V.

Examples

POWer:Power2:IMPEDANCE:AMP1Val 35 sets the generator output amplitude of profile step 1 to 35 volts, for impedance measurement 2.

POWer:Power3:IMPEDANCE:AMP2Val? might return 200.0000E-3, indicating that the signal generator output amplitude is set to 200 mV for profile step 2, for impedance measurement 3.

POWer:POWer<x>:IMPEDANCE:AMPMode

Sets or queries the power amplitude mode for the Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:AMPMode {CONSTant|PROFile}
POWer:POWer<x>:IMPEDANCE:AMPMode?
```

Related Commands

POWer:POWer<x>:IMPEDANCE:AMP[1-10]Val

POWer:POWer<x>:IMPEDANCE:FREQ[1-10]Val

Arguments

POWer<x> is the number of the Impedance power measurement.

CONSTant sets the generator to output a constant level signal.

PROFile uses related commands to set the generator output signal profile (Start frequency, Stop frequency, and Amplitude for each profile step).

Examples

POWer:POWer3:Impedance:AMPMode PROFile sets the Impedance measurement 3 amplitude mode to Profile.

POWer:POWer1:Impedance:AMPMode? might return CONSTANT, indicating that the Impedance measurement 1 amplitude mode is set to Constant.

POWer:POWer<x>:IMPEDANCE:CONNECTSTATus? (Query Only)

Queries the instrument's connection status to the external generator, for the specified Impedance measurement. The Impedance measurement generator IP

address (for external generators) is set with POWer:POWer<x>:IMPEDANCE:GENIPADDress.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group Power

Syntax `POWer:POWer<x>:IMPEDANCE:CONNECTSTATus?`

Arguments `POWer<x>` is the number of the Impedance power measurement.

Returns `SUCCess` indicates the instrument recognizes the connection to the external generator.

`FAILure` indicates the instrument could not make the connection to the external generator.

`UNSUPPorted` indicates that the connected generator is not recognized.

Examples `POWer:POWer1:IMPEDANCE:CONNECTSTATus?` might return `FAILURE`, indicating that the instrument could not connect to the external generator at the specified IP address.

POWer:POWer<x>:IMPEDANCE:CONSTAMPlitude

Sets or queries the constant amplitude value for the specified Impedance measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group Power

Syntax
```
POWer:POWer<x>:IMPEDANCE:CONSTAMPlitude <NR3>
POWer:POWer<x>:IMPEDANCE:CONSTAMPlitude?
```

Arguments `POWer<x>` is the number of the Impedance power measurement.

`<NR3>` specifies the constant amplitude value, in the range of –100 V to 100 V.

Examples

`POWer:POWer3:IMPEDANCE:CONSTAMPlitude 20` sets the constant amplitude value for Impedance power measurement 3 to 20 volts.

`POWer:POWer2:IMPEDANCE:CONSTAMPlitude?` might return `100.0000`, indicating the constant amplitude setting for Impedance power measurement 2 is 100`Hz.

POWer:POWer<x>:IMPEDANCE:FREQ[1-10]Val

Sets or queries the signal generator start frequency of the specified profile step, for the specified Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:FREQ[1-10]Val <NR3>
POWer:POWer<x>:IMPEDANCE:FREQ[1-10]Val?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`FREQ[1–10]` specifies the number of the profile step.

`<NR3>` sets the start frequency, in the range of 10 Hz to 50 MHz, for the specified profile step. You can only set the starting frequency for each profile step; the stop frequency is automatically set to same value as the start frequency of the next profile step. For example, if Step one is set to 1`MHz, and Step two is set to 2`MHz, then the Step one stop frequency is 2`MHz.

Examples

`POWer:POWer5:IMPEDANCE:FREQ3Val 1e3` sets the start frequency of profile step 3 to 1 kHz, for impedance measurement 5.

`POWer:POWer8:IMPEDANCE:FREQ2Val?` might return `2.0000E+6`, indicating that the start frequency of profile step 2 is 2`MHz, for impedance measurement 8.

POWer:POWer<x>:IMPEDANCE:GENerator

Sets or queries the generator source for the Impedance power measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group Power

Syntax
```
POWer:POWer<x>:IMPEDANCE:GENerator {INTernal|EXTernal}
POWer:POWer<x>:IMPEDANCE:GENerator?
```

Arguments `POWer<x>` is the number of the Impedance power measurement.

`INTernal` sets the internal generator as the source for the Impedance power measurement.

`EXTernal` sets the external generator as the source for the Impedance power measurement. Supported external generators are the Tektronix AFG31000 and AFG3000 series.

Examples `POWer:POWer1:Impedance:GENerator Internal` sets the internal generator as the source for the Impedance measurement.

`POWer:POWer7:Impedance:GENerator?` might return `EXTERNAL`, indicating that the Impedance measurement 7 is set to use an external generator.

POWer:POWer<x>:IMPEDANCE:GENIPADDress

Sets or queries the external generator IP Address associated with the specified Impedance measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group Power

Syntax
```
POWer:POWer<x>:IMPEDANCE:GENIPADDress <String>
POWer:POWer<x>:IMPEDANCE:GENIPADDress?
```

Arguments `POWer<x>` is the number of the Impedance power measurement.`<string>` is the IP address of the generator.

Examples `POWer:POWer4:IMPEDANCE:GENIPADDress 164.32.1.5` sets the IP address of the external generator to 164.32.1.5 for the Impedance power measurement 4.

`POWer:POWer2:IMPEDANCE:GENIPADDress?` might return "", indicating that there is no address set for the external generator associated with Impedance power measurement 2.

POWer:POWer<x>:IMPEDANCE:IMPEDANCE

Sets or queries the output impedance of the generator for the specified Impedance power measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:IMPEDANCE {FIFTy|HIGHZ}
POWer:POWer<x>:IMPEDANCE:IMPEDANCE?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`FIFTy` sets the measurement impedance to 50`Ω.

`HIGHZ` sets the measurement impedance to 1`MΩ.

Examples

`POWer:POWer1:IMPEDANCE:IMPEDANCE FIFTY` sets the impedance value for Impedance power measurement 1 to 50 Ω.

`POWer:POWer2:IMPEDANCE:IMPEDANCE?` might return `HIGHZ`, indicating that the impedance value for Impedance power measurement 2 is 1`MΩ

POWer:POWer<x>:IMPEDANCE:INPUTSOurce

Sets or queries the source for the Impedance input measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:IMPEDANCE:INPUTSOurce?
```

Arguments

`POWer<x>` is the number of the Impedance power measurement.

`CH<x>` specifies the instrument input channel number.

`MATH<x>` specifies the instrument math waveform number.

REF<x> specifies the instrument reference waveform number.

Examples

POWer:POWer1:IMPEDANCE:INPUTSOurce MATH5 sets the signal source for the input Impedance power measurement 2 to Math waveform 5.

POWer:POWer2:IMPEDANCE:INPUTSOurce? might return CH3, indicating that the signal source for the input Impedance power measurement 2 is channel 3.

POWer:POWer<x>:IMPEDANCE:OUTPUTSOurce

Sets or queries the source for the Impedance output measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group

Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:OUTPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:IMPEDANCE:OUTPUTSOurce?
```

Arguments

POWer<x> is the number of the Impedance power measurement.

CH<x> specifies the instrument input channel number.

MATH<x> specifies the instrument math waveform number.

REF<x> specifies the instrument reference waveform number.

Examples

POWer:POWer1:IMPEDANCE:OUTPUTSOurce REF3 sets the signal source for the output Impedance power measurement 1 to Reference waveform 3.

POWer:POWer2:IMPEDANCE:OUTPUTSOurce? might return CH3, indicating that the signal source for the output Impedance power measurement 2 is channel 3.

POWer:POWer<x>:IMPEDANCE:PPD

Sets or queries the value for points per decade for the specified Impedance measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group Power

Syntax
```
POWer:POWer<x>:IMPEDANCE:PPD <NR1>
POWer:POWer<x>:IMPEDANCE:PPD?
```

Arguments POWer<x> is the number of the Impedance power measurement.

<NR1> specifies the number of frequency points between the start and stop frequency in terms of log scale, in the range of 10 to 100 points.

Examples POWer:POWer1:IMPEDANCE:PPD 20 sets the constant amplitude value for Impedance power measurement 1 to 20 points per decade.

POWer:POWer2:IMPEDANCE:PPD? might return 30, indicating that the points per decade setting for Impedance power measurement 2 is 30 points per decade.

POWer:POWer<x>:IMPEDANCE:STARTFREQuency

Sets or queries the value for the start frequency of the specified Impedance measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group Power

Syntax
```
POWer:POWer<x>:IMPEDANCE:STARTFREQuency <NR3>
POWer:POWer<x>:IMPEDANCE:STARTFREQuency?
```

Related Commands POWer:POWer<x>:IMPEDANCE:STOPFREQuency

Arguments POWer<x> is the number of the Impedance power measurement.

<NR3> is a floating point number representing the start frequency, in the range of 10 Hz to 50 MHz.

Examples POWer:POWer1:IMPEDANCE:STARTFREQuency 1.500e3 sets the start frequency for Impedance power measurement 1 to 1.5 kHz.

POWer:POWer2:IMPEDANCE:STARTFREQuency? might return 100.0000, indicating the start frequency for Impedance power measurement 2 is 100 Hz.

POWer:POWer<x>:IMPEDANCE:STOPFREQuency

Sets or queries the value of the specified Impedance measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:STOPFREQuency <NR3>
POWer:POWer<x>:IMPEDANCE:STOPFREQuency?
```

Related Commands POWer:POWer<x>:IMPEDANCE:STARTFREQuency

Arguments `POWer<x>` is the number of the Impedance power measurement.

`<NR3>` specifies the start frequency, in the range of 10 Hz to 50 MHz.

Examples `POWer:POWer1:IMPEDANCE:STOPFREQuency 180` sets the stop frequency for Impedance power measurement 1 to 180 Hz.

`POWer:POWer2:IMPEDANCE:STARTFREQuency?` might return `100.0000`, indicating the stop frequency for Impedance power measurement 2 is 100`Hz.

POWer:POWer<x>:IMPEDANCE:TESTCONNection (No Query Form)

This command tests the connection with the external instrument for the specified Impedance measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument).

Group Power

Syntax

```
POWer:POWer<x>:IMPEDANCE:TESTCONNection {EXECute}
```

Related Commands POWer:POWer<x>:IMPEDANCE:CONNECTSTATus?

POWer:POWer<x>:IMPEDANCE:GENIPADDress

Arguments

POWer<x> is the number of the Impedance power measurement.

EXECute runs the test connection function.

Examples

POWer:POWer3:PSRR:TESTCONNection EXECute runs the test connection function for the Impedance power measurement 3.

POWer:POWer<x>:INDUCTANCE:EDGESource

This command sets or queries the edge source for the power inductance measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INDUCTANCE:EDGESource {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:INDUCTANCE:EDGESource?
```

Related Commands

POWer:POWer<x>:INDUCTANCE:ISOUrce

POWer:POWer<x>:INDUCTANCE:VSOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:HARMONICS:INDUCTANCE:EDGESource CH2 sets the signal edge source for inductance measurement 1 as Channel 2.

POWer:POWer3:HARMONICS:INDUCTANCE:EDGESource? might return CH4, indicating that the signal edge source for inductance measurement 3 is Channel 4.

POWer:POWer<x>:INDUCTANCE:ISOUrce

This command sets or queries the current signal source for the inductance measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:INDUCTANCE:ISOUrce
POWer:POWer<x>:INDUCTANCE:ISOUrce?
```

Related Commands POWer:POWer<x>:INDUCTANCE:EDGESource

POWer:POWer<x>:INDUCTANCE:VSOURce

Arguments `POwer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer7:HARMONICS:INDUCTANCE:ISource CH3` sets the current signal source for inductance measurement 7 as Channel 3.

`POWer:POWer3:HARMONICS:INDUCTANCE:ISource?` might return `CH4`, indicating that the current signal source for inductance measurement 3 is Channel 4.

POWer:POWer<x>:INDUCTANCE:VSOURce

This command sets or queries the voltage source for inductance measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:INDUCTANCE:VSOURce
POWer:POWer<x>:INDUCTANCE:VSOURce?
```

Related Commands POWer:POWer<x>:INDUCTANCE:EDGESource

POWer:POWer<x>:INDUCTANCE:ISOUrce

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer7:HARMONICS:INDUCTANCE:VSource CH3 sets the voltage signal source for inductance measurement 7 as Channel 3.

POWer:POWer1:HARMONICS:INDUCTANCE:VSource? might return MATH2, indicating that the voltage signal source for inductance measurement 1 is Math waveform 2.

POWer:POWer<x>:INPUTCAP:ISOURce

This command sets or queries the inrush current input source of the specified Input Capacitance measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INPUTCAP:ISOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:INPUTCAP:ISOURce?
```

Arguments

POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer1:INRUSHcurrent:ISOURce REF3 sets the current source of Input Capacitance 1 to Reference waveform 3.

POWer:POWer2:INRUSHcurrent:ISOURce? might return POWer:POWer2:INRUSHcurrent:ISOURce CH6, indicating that the current source of the Input Capacitance measurement 2 is Channel 6.

POWer:POWer<x>:INPUTCAP:PEAKCURRent

This command sets or queries the peak current value of the specified Input Capacitance measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:INPUTCAP:PEAKCURRent <NR3>
POWer:POWer<x>:INPUTCAP:PEAKCURRent?
```

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the peak current value, in amps, in the range -100 A to 100 A.

Examples POWer:POWer4:INPUTCAP:PEAKCURRent 12.5 sets the peak current value of Input Capacitance measurement 4 to 12.5 amps.

POWer:POWer2:INPUTCAP:PEAKCURRent? might return POWer:POWer2:INPUTCAP:PEAKCURRent –85, indicating that the peak current setting of Input Capacitance measurement 2 is –85 amps.

POWer:POWer<x>:INPUTCAP:PEAKVOLTage

This command sets or queries the peak voltage value of the specified Input Capacitance measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:INPUTCAP:PEAKVOLTage <NR3>
POWer:POWer<x>:INPUTCAP:PEAKVOLTage?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the peak voltage value in the range -100 V to 100 V.

Examples

`POWer:POWer4:INPUTCAP:PEAKVOLTage 122.5` sets the peak voltage value of Input Capacitance measurement 4 to 122.5 volts.

`POWer:POWer2:INPUTCAP:PEAKVOLTage?` might return `POWer:POWer2:INPUTCAP:PEAKVOLTage 955`, indicating that the peak voltage setting of Input Capacitance measurement 2 to 955 volts.

POWer:POWer<x>:INPUTCAP:VSOURce

This command sets or queries the input voltage source of the specified Input Capacitance measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:INPUTCAP:VSOURce {CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:INPUTCAP:VSOURce?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer1:INPUTCAP:VSOURce CH2` sets the voltage source of the input capacitance measurement 1 to Reference Channel 2.

`POWer:POWer1:INPUTCAP:VSOURce?` might return `POWer:POWer2:INPUTCAP:VSOURce MATH6`, indicating that the voltage source of Input Capacitance measurement 2 is Math waveform 6.

POWer:POWer<x>:INRUSHcurrent:INPUTSOurce

This command sets or returns the input source of the specified Inrush Current measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:INRUSHcurrent:INPUTSOurce
POWer:POWer<x>:INRUSHcurrent:INPUTSOurce?
```

Arguments

`POwer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer3:INRUSHcurrent:INPUTSOurce MATH1` sets the input source of Inrush Current measurement 3 to Math waveform 1.

`POWer:POWer2:INRUSHcurrent:INPUTSOurce?` might return `POWer:POWer2:INRUSHcurrent:INPUTSOurce CH7`, indicating that the input source of Inrush Current measurement 2 is Channel 7.

POWer:POWer<x>:INRUSHcurrent:PEAKCURRent

This command sets or returns the peak current value of the specified Inrush Current measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:INRUSHcurrent:PEAKCURRent <NR3>
POWer:POWer<x>:INRUSHcurrent:PEAKCURRent?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the peak current value, in amps, in the range -100 A to 100 A.

Examples

`POWer:POWer4:INRUSHcurrent:PEAKCURRent 12.5` sets the peak current value of Inrush Current measurement 4 to 12.5 amps.

`POWer:POWer2:INRUSHcurrent:PEAKCURRent?` might return `POWer:POWer2:INRUSHcurrent:PEAKCURRent 75`, indicating that the peak current setting of Inrush Current measurement 2 is 75 amps.

POWer:POWer<x>:IVSINTEGRALV:ISOURce

This command sets or queries the current source for I vs Integral V measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:IVSINTEGRALV:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:IVSINTEGRALV:ISOURce?
```

Related Commands

POWer:POWer<x>:IVSINTEGRALV:VSOURce

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:IVSINTEGRALV:ISOURce CH7` sets power measurement 1 to use Channel 7 as the current signal source for the measurement.

`POWer:POWer2:IVSINTEGRALV:ISOURce?` might return `CH1`, indicating that channel 1 is the current signal source for power measurement number 2.

POWer:POWer<x>:IVSINTEGRALV:VSOURce

This command sets or queries the voltage source for I vs Integral V measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:IVSINTEGRALV:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:IVSINTEGRALV:VSOURce?
```

Related Commands POWer:POWer<x>:IVSINTEGRALV:ISOURce

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer2:IVSINTEGRALV:VSOURce CH1` sets I vs Integral V power measurement 2 to use Channel 1 as the voltage signal source for the measurement.

`POWer:POWer4:IVSINTEGRALV:VSOURce?` might return `CH1`, indicating that channel 1 is the voltage source for I vs Integral V power measurement number 4.

POWer:POWer<x>:LABel

This command sets or queries the label for the specified power measurement. As the label can contain non 7-bit ASCII text, it is stored in Percent Encoding format. The power measurement badge is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:LABel <QString>
POWer:POWer<x>:LABel?
```

Examples `POWer:POWer1:LABel "NewMeasurement"` sets the custom measurement name for the measurement in Power 1 badge as New Measurement.

POWer:POWer<x>:LINERIPPLE:INPUTSOurce

This command sets or queries the input source for line ripple measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:LINERIPPLE:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:LINERIPPLE:INPUTSOurce?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:LINERIPPLE:INPUTSOurce CH1` sets channel 1 as the input source for line ripple measurement of power measurement badge Power 1.

POWer:POWer<x>:LINERIPPLE:LFREQuency

This command sets or queries the frequency present for line ripple measurement of the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:LINERIPPLE:LFREQuency {FIFty | SIXty |
FOURHundred}
POWer:POWer<x>:LINERIPPLE:LFREQuency?
```

Examples `POWer:POWer2:LINERIPPLE:LFREQuency FIFTY` sets the frequency present for line ripple measurement as 50 Hz for power measurement 2.

POWer:POWer<x>:MAGNETICLOSS:ISOURce

This command sets or queries the current source for the magnetic loss measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:MAGNETICLOSS:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGNETICLOSS:ISOURce?
```

Related Commands POWer:POWer<x>:MAGNETICLOSS:VSOURce

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:MAGNETICLOSS:ISOURce CH7` sets power measurement 1 to use Channel 7 as the current signal source for the magnetic loss measurement.

`POWer:POWer2:MAGNETICLOSS:ISOURce?` might return `CH1`, indicating that channel 1 is the current signal source for the magnetic loss measurement number 2.

POWer:POWer<x>:MAGNETICLOSS:VSOURce

This command sets or queries the voltage source for magnetic measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:MAGNETICLOSS:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGNETICLOSS:VSOURce?
```

Related Commands

POWer:POWer<x>:MAGNETICLOSS:ISOURce

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer2:MAGNETICLOSS:VSOURce CH4` sets power measurement 2 to use Channel 4 as the voltage signal source for the magnetic loss measurement.

`POWer:POWer1:MAGNETICLOSS:VSOURce?` might return `CH1`, indicating that channel 1 is the voltage source for the magnetic loss measurement number 1.

POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection

This command sets or queries the coil cross section area for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection <NR2>
POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:UNITs

Arguments

`Power<x>` is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`<NR2>` is the cross section area in square meters, in the range of 1 nanometer^2 to 1 M^2.

Examples

`POWer:POWer2:MAGPROPERTY:AREAofcrosssection .005` sets the primary winding cross section area to 5 for power measurement 2.

`POWer:POWer4:MAGPROPERTY:AREAofcrosssection?` might return `10.54E-3`, indicating that the coil cross section area for power measurement 4 is 0.01054.

NOTE. *The coil Cross Section Area unit depends on the setting of the Units value (see POWer:POWer<x>:MAGPROPERTY:UNITs). The command returns the same numeric value regardless of the units setting.*

POWer:POWer<x>:MAGPROPERTY:EDGESOURce

This command sets or queries the edge source type for the magnetic property measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:EDGESOURce {Current | VOLTAGE}
POWer:POWer<x>:MAGPROPERTY:EDGESOURce?
```

Arguments

`Power<x>` is the magnetic property power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`Current` sets the measurement to use the primary voltage source as the signal edge for the magnetic property measurement.

`VOLTAGE` sets the measurement to use the primary current source as the signal edge for the magnetic property measurement.

Examples

`POWer:POWer1:MAGPROPERTY:EDGESource VOLTAGE` sets the signal edge source for power measurement 1 to use the primary voltage source.

`POWer:POWer3:MAGPROPERTY:EDGESource?` might return `CURRENT`, indicating that the signal edge source for power measurement 3 is the primary current source.

POWer:POWer<x>:MAGPROPERTY:ISOURce

This command sets or queries the current source for the magnetic property measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:MAGPROPERTY:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:ISOURce?
```

Related Commands POWer:POWer<x>:MAGPROPERTY:VSOURce

Arguments `Power<x>` is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:MAGPROPERTY:ISOURce CH4` sets power measurement 1 to use channel 4 as the primary winding current source for the magnetic power measurement.

`POWer:POWer2:MAGPROPERTY:ISOURce?` might return `CH1`, indicating that channel 1 is the primary winding current source for the magnetic power measurement number 2.

POWer:POWer<x>:MAGPROPERTY:LENgth

This command sets or queries the conductor length of the primary winding for magnetic measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:MAGPROPERTY:LENgth <NR2>
POWer:POWer<x>:MAGPROPERTY:LENgth?
```

Related Commands POWer:POWer<x>:MAGPROPERTY:UNITs

Arguments

`Power<x>` is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`<NR2>` is the magnetic length, in the range of 1.00E-09 through 1,000,000.

Examples

`POWer:POWer2:MAGPROPERTY:LENgth 2.5` sets the primary winding length to 2.5 for power measurement 2.

`POWer:POWer5:MAGPROPERTY:LENgth?` might return `10.54E-3`, indicating that the primary winding length for power measurement 5 is 0.01054.

NOTE. *The Magnetic Length unit depends on the setting of the Units value (see POWer:POWer<x>:MAGPROPERTY:UNITs). The command returns the same numeric value regardless of the units setting.*

POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs

This command sets or queries the number of primary turns for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:PRIMARYTURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:LENgth

POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection

Arguments

`Power<x>` is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`<NR1>` is the integer number of turns in the primary winding, in the range of 1 to 1 M.

Examples

`POWer:POWer2:MAGPROPERTY:PRIMARYTURNs 150` sets the primary winding turn count to 150 for power measurement 2.

`POWer:POWer5:MAGPROPERTY:PRIMARYTURNs?` might return `50`, indicating that the primary winding has 50 turns for power measurement 5.

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce

This command sets or queries the current source channel for secondary winding 1 for magnetic measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC1SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC1SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer3:MAGPROPERTY:SEC1SOURce CH7` sets power measurement number 3 to use channel 7 as the current source for measuring secondary winding 1.

`POWer:POWer1:MAGPROPERTY:SEC1SOURce?` might return `CH3`, indicating that channel 3 is the source for measuring secondary winding 1 of magnetic power measurement number 1.

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs

This command sets or queries the number of turns of secondary winding 1 for magnetic measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC1TURNs?
```

Related Commands POWer:POWer<x>:MAGPROPERTY:SEC2TURNs

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`<NR1>` is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples `POWer:POWer3:MAGPROPERTY:SEC1TURNs 500` sets the number of turns on secondary winding 1 to 500 for magnetic power measurement number 3.

`POWer:POWer3:MAGPROPERTY:SEC1TURNs?` might return `2500`, indicating that their are 2,500 turns on secondary winding 4 for magnetic power measurement number 3.

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce

This command sets or queries the current source for secondary winding2 for magnetic measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:MAGPROPERTY:SEC2SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC2SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:MAGPROPERTY:SEC2SOURce CH2 sets power measurement number 1 to use channel 2 as the current source for measuring secondary winding 2.

POWer:POWer3:MAGPROPERTY:SEC2SOURce? might return MATH2, indicating that math waveform 2 is the source for measuring secondary winding 2 of magnetic power measurement number 3.

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs

This command sets or queries the number of turns of secondary winding 2 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC2TURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

POWer:POWer1:MAGPROPERTY:SEC2TURNs 50 sets the number of turns on secondary winding 2 to 50 for magnetic power measurement number 3.

POWer:POWer7:MAGPROPERTY:SEC2TURNs? might return 250, indicating that their are 250 turns on secondary winding 2for magnetic power measurement number 7.

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce

This command sets or queries the current source channel for secondary winding 3 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC3SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer1:MAGPROPERTY:SEC3SOURce CH1 sets power measurement number 1 to use channel 1 as the current source for measuring secondary winding 3.

POWer:POWer1:MAGPROPERTY:SEC3SOURce? might return CH1, indicating that channel 1 is the source for measuring secondary winding 3 of magnetic power measurement number 1.

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs

This command sets or queries the number of turns of secondary winding 3 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC3TURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

POWer:POWer1:MAGPROPERTY:SEC3TURNs 15 sets the number of turns on secondary winding 3 to 15 for magnetic power measurement number 1.

POWer:POWer3:MAGPROPERTY:SEC3TURNs? might return 2500, indicating that their are 2,500 turns on secondary winding 3 for magnetic power measurement number 3.

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce

This command sets or queries the current source for secondary winding 4 for magnetic measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC4SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC4SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer3:MAGPROPERTY:SEC4SOURce CH7` sets power measurement number 3 to use channel 7 as the current source for measuring secondary winding 4.

`POWer:POWer12:MAGPROPERTY:SEC4SOURce?` might return `CH1`, indicating that Channel 1 is the source for measuring secondary winding 4 of magnetic power measurement number 12.

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs

This command sets or queries the number of turns of secondary winding 4 for magnetic measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC4TURNs?
```

Related Commands POWer:POWer<x>:MAGPROPERTY:SEC1TURNs

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`<NR1>` is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples `POWer:POWer4:MAGPROPERTY:SEC4TURNs 500` sets the number of turns on secondary winding 4 to 500 for magnetic power measurement number 4.

`POWer:POWer1:MAGPROPERTY:SEC4TURNs?` might return 90, indicating that their are 90 turns on secondary winding 4 for magnetic power measurement number 3.

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce

This command sets or queries the current source for secondary winding 5 for magnetic measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC5SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC5SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer3:MAGPROPERTY:SEC5SOURce MATH1 sets power measurement number 3 to use math waveform 1 as the current source for measuring secondary winding 5.

POWer:POWer4:MAGPROPERTY:SEC5SOURce? might return CH2, indicating that channel 2 is the source for measuring secondary winding 5 of magnetic power measurement number 4.

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs

This command sets or queries the number of turns of secondary winding 5 for magnetic measurement of the specified power measurement badge.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC5TURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`<NR1>` is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

`POWer:POWer3:MAGPROPERTY:SEC5TURNs 250` sets the number of turns on secondary winding 5 to 250 for magnetic power measurement number 3.

`POWer:POWer1:MAGPROPERTY:SEC5TURNs?` might return `150`, indicating that their are 150 turns on secondary winding 5for magnetic power measurement number 1.

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

This command sets or queries the current source for secondary winding 6 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:SEC6SOURce?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples

POWer:POWer3:MAGPROPERTY:SEC6SOURce CH4 sets power measurement number 3 to use channel 4 as the current source for measuring secondary winding 6.

POWer:POWer2:MAGPROPERTY:SEC6SOURce? might return CH4, indicating that channel 4 is the source for measuring secondary winding 3 of magnetic power measurement number 2.

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

This command sets or queries the number of turns of secondary winding 6 for magnetic measurement of the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs <NR1>
POWer:POWer<x>:MAGPROPERTY:SEC6TURNs?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

<NR1> is the number of turns on the secondary winding, and ranges from 0 to 1,000,000.

Examples

POWer:POWer8:MAGPROPERTY:SEC6TURNs 22 sets the number of turns on secondary winding 6 to 22 for magnetic power measurement number 8.

POWer:POWer9:MAGPROPERTY:SEC6TURNs? might return 7000, indicating that their are 7,000 turns on secondary winding 6 for magnetic power measurement number 9.

POWer:POWer<x>:MAGPROPERTY:SECWINDings

This command sets or queries the number of secondary windings for the magnetic property measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:SECWINDings
{None|ONE|TWO|THREE|FOUR|FIVE|SIX}
POWer:POWer<x>:MAGPROPERTY:SECWINDings?
```

Related Commands

POWer:POWer<x>:MAGPROPERTY:SEC1SOURce

POWer:POWer<x>:MAGPROPERTY:SEC1TURNs

POWer:POWer<x>:MAGPROPERTY:SEC2SOURce

POWer:POWer<x>:MAGPROPERTY:SEC2TURNs

POWer:POWer<x>:MAGPROPERTY:SEC3SOURce

POWer:POWer<x>:MAGPROPERTY:SEC3TURNs

POWer:POWer<x>:MAGPROPERTY:SEC4SOURce

POWer:POWer<x>:MAGPROPERTY:SEC4TURNs

POWer:POWer<x>:MAGPROPERTY:SEC5SOURce

POWer:POWer<x>:MAGPROPERTY:SEC5TURNs

POWer:POWer<x>:MAGPROPERTY:SEC6SOURce

POWer:POWer<x>:MAGPROPERTY:SEC6TURNs

Arguments

`Power<x>` is the magnetic property power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`None`, `ONE`, `TWO`, `THREE`, `FOUR`, `FIVE`, `SIX` sets the number of secondary windings to the specified value.

Examples

`POWer:POWer2:MAGPROPERTY:SECWINDings 4` sets power measurement 2 to use 4 secondary transformer windings.

`POWer:POWer11:MAGPROPERTY:SECWINDings?` might return `FOUR`, indicating that there are four secondary transformer windings for power measurement 11.

POWer:POWer<x>:MAGPROPERTY:UNITs

This command sets or queries the units for magnetic measurements of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:UNITs {SI|CGS}
POWer:POWer<x>:MAGPROPERTY:UNITs?
```

Related Commands POWer:POWer<x>:MAGPROPERTY:AREAofcrosssection

POWer:POWer<x>:MAGPROPERTY:LENgth

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`SI` sets the measurement to International System of Units.

`CGS` sets the measurement to Gaussian units.

Examples `POWer:POWer5:MAGPROPERTY:UNITs SI` sets magnetic property measurement 5 to use SI units.

`POWer:POWer1:MAGPROPERTY:UNITs?` might return `CGS`, indicating that the measurement unit for power measurement 1 is set to CGS.

POWer:POWer<x>:MAGPROPERTY:VSOURce

This command sets or queries the primary winding voltage source for the magnetic measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:MAGPROPERTY:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:MAGPROPERTY:VSOURce?
```

Related Commands POWer:POWer<x>:MAGPROPERTY:ISOURce

Arguments POWer<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples POWer:POWer1:MAGPROPERTY:VSOURce CH4 sets power measurement 1 to use channel 4 as the primary winding voltage source for the magnetic power measurement.

POWer:POWer2:MAGPROPERTY:VSOURce? might return CH1, indicating that channel 1 is the primary winding voltage source for the magnetic power measurement number 2.

POWer:POWer<x>:NDUTYCYCLE:EDGEType

This command sets or queries the clock edge type for negative duty cycle measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:NDUTYCYCLE:EDGEType {RISE | FALL | BOTH}
POWer:POWer<x>:NDUTYCYCLE:EDGEType?
```

Examples POWer:POWer1:NDUTYCYCLE:EDGEType RISE sets the clock edge type as rise for the negative duty cycle measurement.

POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce

This command sets or queries the input source for negative duty cycle measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:NDUTYCYCLE:INPUTSOurce?
```

Arguments POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:NDUTYCYCLE:INPUTSOurce CH3` sets the input source for negative duty cycle measurement as CH3 for the power measurement badge Power 1.

POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce

This command sets or queries the input source for negative pulse width measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce
{CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:NPULSEWIDTH:INPUTSOurce?
```

Arguments POWer<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:NPULSEWIDTH:INPUTSOurce CH1` sets the input source for negative pulse width measurement as channel 1.

POWer:POWer<x>:PDUTYCYCLE:EDGEType

This command sets or queries the clock edge type for positive duty cycle measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:PDUTYCYCLE:EDGEType {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:PDUTYCYCLE:EDGEType?
```

Examples `POWer:POWer1:PDUTYCYCLE:EDGEType BOTH` sets the clock edge type as both (rise and fall) for the positive duty cycle measurement.

POWer:POWer<x>:PDUTYCYCLE:INPUTSOurce

This command sets or queries the input source for positive duty cycle measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:PDUTYCYCLE:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:PDUTYCYCLE:INPUTSOurce?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:PDUTYCYCLE:INPUTSOurce CH4` sets the input source for positive duty cycle measurement as channel 4.

POWer:POWer<x>:PERIOD:EDGe

This command sets or queries the edge type for period measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:PERIOD:EDGe {RISE | FALL}
POWer:POWer<x>:PERIOD:EDGe?
```

Examples `POWer:POWer1:PERIOD:EDGe RISE` sets the edge type as rise for the period measurement.

POWer:POWer<x>:PERIOD:INPUTSOurce

This command sets or queries the input source for period measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:PERIOD:INPUTSOurce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:PERIOD:INPUTSOurce?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:PERIOD:INPUTSOurce CH6` sets the input source for period measurement as channel 6.

POWer:POWer<x>:POWERQUALITY:CCYCles

This command sets or queries the calculate cycles over full cycles settings for the specified power quality measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:POWERQUALITY:CCYCles {ON | OFF 1 | 0}
POWer:POWer<x>:POWERQUALITY:CCYCles?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`1 | ON` selects the calculate cycles over full cycles.

`0 | OFF` unselects the calculate cycles over full cycles.

Examples `POWer:POWer1:POWERQUALITY:CCYCles 1` selects the calculate cycles over full cycles for the power measurement badge 1.

POWer:POWer<x>:POWERQUALITY:FREFerence

This command sets or queries the frequency reference type for power quality measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:POWERQUALITY:FREFerence {VOLTage | CURRent}
POWer:POWer<x>:POWERQUALITY:FREFerence?
```

Examples `POWer:POWer1:POWERQUALITY:FREFerence CURRent` sets the frequency reference type for power quality measurement as current.

POWer:POWer<x>:POWERQUALITY:ISOURce

This command sets or queries the current source for power quality measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:POWERQUALITY:ISOURce {CH<x> | MATH<x> |
REF<x>}
POWer:POWer<x>:POWERQUALITY:ISOURce?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:POWERQUALITY:VSOURce CH2` sets the current source for power quality measurement as channel 2.

POWer:POWer<x>:POWERQUALITY:VSOURce

This command sets or queries the voltage source for power quality measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:POWERQUALITY:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:POWERQUALITY:VSOURce?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples POWer:POWer1:POWERQUALITY:VSOURce CH1 sets the voltage source for power quality measurement as channel 1.

POWer:POWer<x>:PPULSEWIDTH:INPUTSOurce

This command sets or queries the input source for positive pulse width measurement in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:PPULSEWIDTH:INPUTSOurce
{CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:PPULSEWIDTH:INPUTSOurce?
```

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples POWer:POWer1:PPULSEWIDTH:INPUTSOurce CH5 sets channel 5 as the input source for the positive pulse width measurement 1.

POWer:POWer<x>:PRESET (No Query Form)

This command runs a power preset action for the specified power measurement number.

Conditions Requires option 5-PWR, 6–PWR, or PS2.

Group Power

Syntax `POWer:POWer<x>:PRESET {EXECute}`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`EXECute` runs the power preset action.

Examples `POWer:POWer5:PRESET Execute` runs the power preset action for power measurement 5.

POWer:POWer<x>:PSRR:AMP[1-10]Val

This command sets or queries the generator amplitude value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax
```
POWer:POWer<x>:PSRR:AMP[1-10]Val <NR3>
POWer:POWer<x>:PSRR:AMP[1-10]Val?
```

Arguments `Power<x>` sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`AMP[1–10]` sets the configuration step number, in the range of 1 to 10. Values outside this range will report an error.

`<NR3>` sets the generator amplitude for the specified configuration step, in the range of –100 V to 100 V.

Examples `POWer:POWer1:PSRR:AMP3Val 20` sets the generator output amplitude for configuration step 3 to 20 volts, for power measurement 1.

`POWer:POWer2:PSRR:AMP8Val?` might return `60`, indicating that the generator output amplitude setting of configuration step 8 is 60 volts, for power measurement 2.

POWer:POWer<x>:PSRR:AMPMode

This command sets or queries the amplitude mode for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax

```
POWer:POWer<x>:PSRR:AMPMode {CONSTant|PROFile}
POWer:POWer<x>:PSRR:AMPMode?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CONSTant` sets the amplitude mode to output a constant amplitude signal from the DUT stimulus generator for all frequency bands.

`PROFile` enables configuring the generator to set amplitude values for each frequency band.

Examples

`POWer:POWer1:PSRR:AMPMode CONSTant` sets the amplitude mode to Constant for power measurement 1.

`POWer:POWer2:PSRR:AMPMode?` might return `PROFile`, indicating that the amplitude mode power measurement 2 is set to Profile.

POWer:POWer<x>:PSRR:CONNECTSTATus? (Query Only)

Queries the external instrument's connection status for the specified Power Supply Rejection Ratio (PSRR) measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group Power

Syntax

```
POWer:POWer<x>:PSRR:CONNECTSTATus?
```

Arguments

`POWer<x>` is the number of the PSRR power measurement.

Returns `SUCCess` indicates the instrument recognizes the connection to the external generator.

`FAILure` indicates the instrument could not make the connection to the external generator.

`UNSUPPorted` indicates that the connected generator is not recognized.

Examples `POWer:POWer<x>:PSRR:CONNECTSTATus?` might return `FAILURE`, indicating that the instrument could not connect to the external generator at the specified IP address.

POWer:POWer<x>:PSRR:CONSTAMPlitude

This command sets or queries the constant amplitude voltage for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax
```
POWer:POWer<x>:PSRR:CONSTAMPlitude <NR3>
POWer:POWer<x>:PSRR:CONSTAMPlitude?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the constant amplitude voltage value for the measurement, in the range of –100 V to 100 V.

Examples `POWer:POWer3:PSRR:CONSTAMPlitude 120` sets the constant amplitude voltage for Power measurement 3 to be 120 volts.

`POWer:POWer5:PSRR:CONSTAMPlitude?` might return `-15`, indicating that the constant amplitude voltage for power measurement 5 is –15 volts.

POWer:POWer<x>:PSRR:FREQ[1-11]Val

This command sets or queries the generator frequency value of the specified configuration step for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax

```
POWer:POWer<x>:PSRR:FREQ[1-11]Val <NR3>
POWer:POWer<x>:PSRR:FREQ[1-11]Val?
```

Arguments

`POWer<x>` sets the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`FREQ[1-11]` sets the configuration step number, in the range of 1 to 11. Values outside this range will report an error.

`<NR3>` sets the frequency of the specified configuration step number, in the range of 10 Hz to 50 MHz.

Examples

`POWer:POWer5:PSRR:FREQ1Val 200` sets the generator frequency value for frequency band 1 to 200 Hz, for power measurement 5.

`POWer:POWer2:PSRR:FREQ3Val?` might return `2.000E+6`, indicating that the generator frequency output for frequency band 3 is 2.0 MHz, for power measurement 2.

POWer:POWer<x>:PSRR:GENerator

This command sets or queries the generator source used to send stimulus signals to the DUT, for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax

```
POWer:POWer<x>:PSRR:GENerator {INTernal}
POWer:POWer<x>:PSRR:GENerator?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`INTernal` sets the generator to the instrument AFG signal source. This is the only valid argument.

Examples

`POWer:POWer3:PSRR:GENerator INTernal` sets the generator to the instrument AFG for power measurement 3.

`POWer:POWer2:PSRR:GENerator?` might return `INTERNAL`, indicating that the generator source for power measurement 2 is the instrument AFG.

POWer:POWer<x>:PSRR:GENerator

Sets or queries the generator source for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:GENerator {INTernal|EXTernal}
POWer:POWer<x>:PSRR:GENerator?
```

Arguments

`POWer<x>` is the number of the PSRR power measurement.

`INTernal` sets the internal generator as the source for the Power Supply Rejection Ratio (PSRR) power measurement.

`EXTernal` sets the external generator as the source for the Power Supply Rejection Ratio (PSRR) power measurement.

Examples

`POWer:POWer1:PSRR:GENerator Internal` sets the internal generator as the source for the Power Supply Rejection Ratio (PSRR) measurement.

`POWer:POWer7:PSRR:GENerator?` might return `EXTERNAL`, indicating that the Power Supply Rejection Ratio (PSRR) measurement 7 is set to use an external generator.

POWer:POWer<x>:PSRR:GENIPADDress

Sets or queries the instrument's IP Address associated with the specified Power Supply Rejection Ratio (PSRR) measurement.

Conditions

Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group Power

Syntax
```
POWer:POWer<x>:PSRR:GENIPADDress <Qstring>
POWer:POWer<x>:PSRR:GENIPADDress?
```

Arguments `POWer<x>` is the number of the PSRR power measurement.`<NR2>` is the IP address of the generator.

Examples `POWer:POWer4:PSRR:GENIPADDress 127.1.1.1` sets the IP address of the external generator to 127.1.1.1 for Power Supply Rejection Ratio (PSRR) power measurement 4.

`POWer:POWer2:PSRR:GENIPADDress?` might return "", indicating that there is no address set for the external generator associated with Power Supply Rejection Ratio (PSRR) power measurement 2.

POWer:POWer<x>:PSRR:IMPEDance

This command sets or queries the vertical termination impedance for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax
```
POWer:POWer<x>:PSRR:IMPEDance {FIFTy|HIGHZ}
POWer:POWer<x>:PSRR:IMPEDance?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`FIFTy` sets the impedance to be 50 Ω.

`HIGHZ` sets the impedance to be 1 MΩ.

Examples `POWer:POWer3:PSRR:IMPEDance FIFTy` sets the vertical termination impedance for power measurement 3 to be 50 Ω.

`POWer:POWer2:PSRR:IMPEDance?` might return `HIGHZ`, indicating that the vertical termination impedance for power measurement 2 is 1 MΩ.

POWer:POWer<x>:PSRR:INPUTSOurce

This command sets or queries the input source for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax

```
POWer:POWer<x>:PSRR:INPUTSOurce CH<x>
POWer:POWer<x>:PSRR:INPUTSOurce?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` sets the channel to use for the input source.

Examples

`POWer:POWer2:PSRR:INPUTSOurce CH3` sets the input for power measurement 2 to be the channel 3 waveform.

`POWer:POWer2:PSRR:INPUTSOurce?` might return `CH1`, indicating that the input source for power measurement 1 is channel 1.

POWer:POWer<x>:PSRR:OUTPUTSOurce

This command sets or queries the output source for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax

```
POWer:POWer<x>:PSRR:OUTPUTSOurce CH<x>
POWer:POWer<x>:PSRR:OUTPUTSOurce?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` sets the channel to use for the output source.

Examples

`POWer:POWer1:PSRR:OUTPUTSOurce CH3` sets the output source for power measurement 1 to be the channel 3 waveform.

`POWer:POWer3:PSRR:OUTPUTSOurce?` might return `CH6`, indicating that the output source for power measurement 3 is channel 6.

POWer:POWer<x>:PSRR:PPD

This command sets or queries the points per decade (PPD) value for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:PPD <NR3>
POWer:POWer<x>:PSRR:PPD?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the PPD value for the measurement, in the range of 10 to 100 points.

Examples

`POWer:POWer3:PSRR:PPD 30` sets the PPD for Power measurement 3 to be 30 points.

`POWer:POWer5:PSRR:PPD?` might return `20`, indicating that the PPD value for power measurement 5 is 20 points.

POWer:POWer<x>:PSRR:STARTFREQuency

This command sets or queries the start frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions

Requires option 5-PWR or 6-PWR

Group

Power

Syntax

```
POWer:POWer<x>:PSRR:STARTFREQuency <NR3>
POWer:POWer<x>:PSRR:STARTFREQuency?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the starting frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples `POWer:POWer8:PSRR:STARTFREQuency 12` sets the starting frequency for power measurement 8 to be 12 Hz.

`POWer:POWer3:PSRR:STARTFREQuency?` might return `2400`, indicating that the starting frequency for power measurement 3 is 2400 Hz.

POWer:POWer<x>:PSRR:STOPFREQuency

This command sets or queries the stop frequency value for the Power Supply Rejection Ratio (PSRR) power measurement.

Conditions Requires option 5-PWR or 6-PWR

Group Power

Syntax
```
POWer:POWer<x>:PSRR:STOPFREQuency <NR3>
POWer:POWer<x>:PSRR:STOPFREQuency?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR3>` is the stop frequency for the measurement, in the range of 10 Hz to 50 MHz.

Examples `POWer:POWer3:PSRR:STOPFREQuency 120` sets the stop frequency for Power measurement 8 to be 120 Hz.

`POWer:POWer5:PSRR:STOPFREQuency?` might return `2000`, indicating that the stop frequency for power measurement 5 is 2000 Hz.

POWer:POWer<x>:PSRR:TESTCONNection (No Query Form)

This command tests the connection with the external instrument for the specified Power Supply Rejection Ratio (PSRR) measurement.

Conditions Requires option 5-PWR (5 Series MSO instruments) or 6-PWR (6 Series MSO instrument)

Group Power

Syntax `POWer:POWer<x>:PSRR:TESTCONNection {EXECute}`

Related Commands POWer:POWer<x>:PSRR:CONNECTSTATus?

POWer:POWer<x>:PSRR:GENIPADDress

Arguments `POWer<x>` is the number of the PSRR power measurement.

`EXECute` runs the test connection function.

Examples `POWer:POWer3:PSRR:TESTCONNection EXECute` runs the test connection function for the Power Supply Rejection Ratio (PSRR) power measurement 3.

POWer:POWer<x>:RDSON:DEVICEType

This command sets or queries the device type for the power drain source on resistance measurement for RDSon measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:RDSON:DEVICEType {SWITCHING | PNJUNCTION}
POWer:POWer<x>:RDSON:DEVICEType?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`SWITCHING` sets the Device Type to a switching device (v/i).

`PNJUNCTION` sets the Device Type to a PN Junction device (dv/di).

Examples `POWer:POWer3:RDSON:DEVICEType PNJUNCTION` sets the RDSon power measurement number 3 to measure a PN junction device.

`POWer:POWer1:RDSON:DEVICEType?` might return `SWITCHING`, indicating that RDSon power measurement number 1 is set to measure a switching device.

POWer:POWer<x>:RDSON:ISOURce

This command sets or queries the current source for RDSon measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RDSON:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:RDSON:ISOURce?
```

Related Commands POWer:POWer<x>:RDSON:VSOURce

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

`CH<x>` sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples

`POWer:POWer1:RDSON:ISOURce REF2` sets RDSon power measurement number 1 to use Reference waveform 2 as the current signal source.

`POWer:POWer1:RDSON:ISOURce?` might return `CH1`, indicating that channel 1 is the current signal source for the RDSon power measurement number 1.

POWer:POWer<x>:RDSON:VSOURce

This command sets or queries the voltage source for RDSon measurement of the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RDSON:VSOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:RDSON:VSOURce?
```

Related Commands POWer:POWer<x>:RDSON:ISOURce

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown in the power measurement badge on the UI.

CH<x> sets the channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples POWer:POWer2:RDSON:VSOURce CH7 sets RDSon power measurement number 2 to use channel 7 as the voltage source.

POWer:POWer1:RDSON:VSOURce? might return CH7, indicating that channel 1 is the voltage signal source for RDSon power measurement number 1.

POWer:POWer<x>:REFLevels:ABSolute:FALLHigh

This command sets or queries the falling edge for high reference level in absolute units for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:FALLHigh <NR1>
POWer:POWer<x>:REFLevels:ABSolute:FALLHigh?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:ABSolute:TYPE

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from –40000 to 40000

Examples POWer:POWer1:REFLevels:ABSolute:FALLHigh 1 sets the high reference level for falling edge as 1 volt.

POWer:POWer<x>:REFLevels:ABSolute:FALLLow

This command sets or queries the falling edge for low reference level in absolute units for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:FALLLow <NR1>
POWer:POWer<x>:REFLevels:ABSolute:FALLLow?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:ABSolute:TYPE

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from –40000 to 40000

Examples `POWer:POWer1:REFLevels:ABSolute:FALLLow –1` sets the low reference level for falling edge as –1 volt.

POWer:POWer<x>:REFLevels:ABSolute:FALLMid

This command sets or queries the falling edge for mid reference level in absolute units for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:FALLMid <NR1>
POWer:POWer<x>:REFLevels:ABSolute:FALLMid?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:ABSolute:TYPE

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from -40000 to 40000

Examples `POWer:POWer1:REFLevels:ABSolute:FALLMid 0` sets the mid reference level for falling edge as 0 volt.

POWer:POWer<x>:REFLevels:ABSolute:HYSTeresis

This command sets or queries the absolute hysteresis value for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:REFLevels:ABSolute:HYSTeresis <NR1>
POWer:POWer<x>:REFLevels:ABSolute:HYSTeresis?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 0.0000005 to 10

Examples `POWer:POWer1:REFLevels:ABSolute:HYSTeresis 10` sets the reference level hysteresis value in absolute to 10.

POWer:POWer<x>:REFLevels:ABSolute:RISEHigh

This command sets or queries the rising edge for high reference level in absolute units for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:RISEHigh <NR1>
POWer:POWer<x>:REFLevels:ABSolute:RISEHigh?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:ABSolute:TYPE

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from –40000 to 40000

Examples

`POWer:POWer1:REFLevels:ABSolute:RISEHigh 1` sets the high reference level for rising edge as 1 volt.

POWer:POWer<x>:REFLevels:ABSolute:RISELow

This command sets or queries the rising edge for low reference level in absolute units for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:RISELow <NR1>
POWer:POWer<x>:REFLevels:ABSolute:RISELow?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:ABSolute:TYPE

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from –40000 to 40000.

Examples

`POWer:POWer1:REFLevels:ABSolute:RISELow -1` sets the low reference level for rising edge as –1 volt.

POWer:POWer<x>:REFLevels:ABSolute:RISEMid

This command sets or queries the rising edge for mid reference level in absolute units for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:RISEMid <NR1>
POWer:POWer<x>:REFLevels:ABSolute:RISEMid?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:ABSolute:TYPE

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from –40000 to 40000.

Examples `POWer:POWer1:REFLevels:ABSolute:RISEMid 0` sets the mid reference level for rising edge as 0 volt.

POWer:POWer<x>:REFLevels:ABSolute:TYPE

This command sets or queries the type of measurement levels when reference level is set to absolute for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:ABSolute:TYPE {SAME | UNIQue}
POWer:POWer<x>:REFLevels:ABSolute:TYPE?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`SAME`: select when the rising edge and falling edge reference levels are same.

`UNIQue`: select when the rising edge and falling edge reference levels are different.

Examples

`POWer:POWer1:REFLevels:ABSolute:TYPE UNIQue` sets the type of measurement levels as unique for the specified power measurement badge.

POWer:POWer<x>:REFLevels:BASETop

This command sets or queries the reference level base top method for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:BASETop {AUTO | MINMax |
MEANhistogram | MODEhistogram | EYEhistogram}
POWer:POWer<x>:REFLevels:BASETop?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod

Examples

`POWer:POWer1:REFLevels:BASETop AUTO` sets the reference level base top method as auto for the power measurement badge 1.

POWer:POWer<x>:REFLevels:METHod

This command sets or queries the method to configure reference level values for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:METHod {PERCent | ABSolute}
POWer:POWer<x>:REFLevels:METHod?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`PERCent` sets the power measurement to use absolute values to configure reference level values.

`ABSolute` sets the power measurement to use percentage to configure reference level values.

Examples

`POWer:POWer3:REFLevels:METHod PERCent` sets power measurement 3 to use percentage to configure reference level values.

POWer:POWer<x>:REFLevels:PERCent:FALLHigh

This command sets or queries the falling edge for high reference level in percentage for the specified power measurement number.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:FALLHigh <NR1>
POWer:POWer<x>:REFLevels:PERCent:FALLHigh?
```

Related Commands

POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:PERCent:TYPE

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 1 to 99.

Examples

`POWer:POWer1:REFLevels:PERCent:FALLHigh 1` sets the high reference level for falling edge as 1 percentage.

POWer:POWer<x>:REFLevels:PERCent:FALLLow

This command sets or queries the falling edge for low reference level in percentage for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:FALLLow <NR1>
POWer:POWer<x>:REFLevels:PERCent:FALLLow?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:PERCent:TYPE

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 1 to 99.

Examples `POWer:POWer1:REFLevels:PERCent:FALLLow 0` sets the low reference level for falling edge as 0 percentage.

POWer:POWer<x>:REFLevels:PERCent:FALLMid

This command sets or queries the falling edge for mid reference level in percentage for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:FALLMid <NR1>
POWer:POWer<x>:REFLevels:PERCent:FALLMid?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:PERCent:TYPE

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 1 to 99.

Examples `POWer:POWer1:REFLevels:PERCent:FALLMid 50` sets the mid reference level for falling edge as 50 percentage.

POWer:POWer<x>:REFLevels:PERCent:HYSTeresis

This command sets or queries the hysteresis in percentage for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:HYSTeresis <NR1>
POWer:POWer<x>:REFLevels:PERCent:HYSTeresis?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 1 to 99.

Examples `POWer:POWer1:REFLevels:PERCent:HYSTeresis 25` sets the hysteresis as 25 percentage.

POWer:POWer<x>:REFLevels:PERCent:RISEHigh

This command sets or queries the rising edge for high reference level in percentage for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:RISEHigh <NR1>
POWer:POWer<x>:REFLevels:PERCent:RISEHigh?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:PERCent:TYPE

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from 1 to 99.

Examples POWer:POWer1:REFLevels:PERCent:RISEHigh 85 sets the high reference level for rising edge as 85 percentage.

POWer:POWer<x>:REFLevels:PERCent:RISELow

This command sets or queries the rising edge for low reference level in percentage for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:REFLevels:PERCent:RISELow <NR1>
POWer:POWer<x>:REFLevels:PERCent:RISELow?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:PERCent:TYPE

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from 1 to 99.

Examples POWer:POWer1:REFLevels:PERCent:RISELow 10 sets the low reference level for rising edge as 10 percentage.

POWer:POWer<x>:REFLevels:PERCent:RISEMid

This command sets or queries the rising edge for mid reference level in percentage for the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:REFLevels:PERCent:RISEMid <NR1>
POWer:POWer<x>:REFLevels:PERCent:RISEMid?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:PERCent:TYPE

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 1 to 99.

Examples `POWer:POWer1:REFLevels:PERCent:RISEMid 55` sets the mid reference level for rising edge as 55 percentage.

POWer:POWer<x>:REFLevels:PERCent:TYPE

This command sets or queries the reference levels for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:REFLevels:PERCent:TYPE {TENNinety |
TWENtyeighty | CUSTom}
POWer:POWer<x>:REFLevels:PERCent:TYPE?
```

Related Commands POWer:POWer<x>:REFLevels:METHod

POWer:POWer<x>:REFLevels:PERCent:FALLHigh

POWer:POWer<x>:REFLevels:PERCent:FALLLow

POWer:POWer<x>:REFLevels:PERCent:FALLMid

POWer:POWer<x>:REFLevels:PERCent:RISEHigh

POWer:POWer<x>:REFLevels:PERCent:RISELow

POWer:POWer<x>:REFLevels:PERCent:RISEMid

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

TENNinety to set the low reference levels as 10% and high reference levels as 90%

TWENtyeighty to set the low reference levels as 20% and high reference levels as 80%

CUSTom to set the custom low, high, and mid reference levels for rising and falling edges

Examples

POWer:POWer1:REFLevels:PERCent:TYPE TENNinety sets the low reference levels as 10% and high reference levels as 90% for the power measurement badge 1.

POWer:POWer<x>:RESUlts:ALLAcqs:MAXimum? (Query Only)

This command queries the maximum value of all acquisitions for the measurement parameter in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:
MAXimum?  {InputPwr|Output1Pwr|
Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|
TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|
Br|Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|
RDS|TRUEPWR|APPPWR|REPWR|PWRFACTOR|PHASE|
PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|
TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|
TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|
LRIPRMS|LRIPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|
FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|
PKPK|HIGH|LOW|MAX|MIN|  INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For the Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

`POWer:POWer4:RESUlts:ALLAcqs:MAXimum? "PKPK"` might return `28.56`, indicating the maximum value of the Peak-to-Peak measurement for power measurement 4.

POWer:POWer<x>:RESUlts:ALLAcqs:MEAN? (Query Only)

This command queries the mean value of all acquisitions for the measurement parameter in the specified power measurement number <x>.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:MEAN?
{InputPwr|Output1Pwr|Output2Pwr|
Output3Pwr|Efficiency1|Efficiency2|
Efficiency3|TotalEfficiency|
INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|
IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|
APPPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|
ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|
TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|
```

TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|
LRIPRMS|LRIPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|
FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|
PKPK|HIGH|LOW|MAX|MIN| INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT is the parameter for the dI by dt measurement.

"SOAHITSCNT is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples `POWer:POWer2:RESUlts:ALLAcqs:MEAN? "AMPL"` might return `57.45`, indicating the mean value of the Amplitude measurement for all acquisitions of power measurement 2.

POWer:POWer<x>:RESUlts:ALLAcqs:MINimum? (Query Only)

This command queries the minimum value of all acquisitions for the measurement parameter of the specified power measurement <x>.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:
MINimum?  {InputPwr|Output1Pwr|
Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|
TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|
Bpeak|Br|Hc|Hmax|IRipple|DeltaB|DeltaH|
Permeability|RDS|TRUEPWR|APPPWR|REPWR|
PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|
IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|
TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|
DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|
SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|
NDUTY|PPULSE|NPULSE|AMPL|PKPK|
HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT is the parameter for the dI by dt measurement.

"SOAHITSCNT is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples `POWer:POWer5:RESUlts:ALLAcqs:MINimum? "FREQ"` might return `5.1307829019093E6`, indicating the minimum frequency measurement for all acquisitions of power measurement 5.

POWer:POWer<x>:RESUlts:ALLAcqs:PK2PK? (Query Only)

This command queries the peak-to-peak value of all acquisitions for the measurement parameter in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:PK2PK? {InputPwr|Output1Pwr|
Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|
TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|
Br|Hc|Hmax|IRipple|DeltaB|DeltaH|
Permeability|RDS|TRUEPWR|APPPWR|REPWR|
PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|
IRMS|VRMS|TONENRG|TONLOSS|TOFFENRG|
TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|
DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|
SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|
NDUTY|PPULSE|NPULSE|AMPL|PKPK|
HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

`POWer:POWer1:RESUlts:ALLAcqs:PK2PK? "TONLoss"` might return `9.91`, indicating the peak-to-peak value of T-On Energy for all acquisitions.

POWer:POWer<x>:RESUlts:ALLAcqs:POPUlation? (Query Only)

This command queries the population (number of complete cycles) of all acquisitions for the measurement parameter in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:
POPUlation?  {InputPwr|Output1Pwr|
Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|
TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|
Br|Hc|Hmax|IRipple|DeltaB|DeltaH|
Permeability|RDS|TRUEPWR|APPPWR|REPWR|PWRFACTOR|
PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|
TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|
TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|
LRIPRMS|LRIPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|
FREQ|PDUTY|NDUTY|PPULSE|NPULSE|
AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples `POWer:POWer1:RESUlts:ALLAcqs:POPUlation? "CondEnrg"` might return `4.91E-3`, indicating the population (number of complete cycles) of conduction energy for all acquisitions.

POWer:POWer<x>:RESUlts:ALLAcqs:STDDev? (Query Only)

This command queries the standard deviation value of all acquisitions for the measurement parameter in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:ALLAcqs:
STDDev? {InputPwr|Output1Pwr|
Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|
TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|
Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|
TRUEPWR|APPPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|
ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|TONLOSS|
TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|
TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|
LRIPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|
NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN|
INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples `POWer:POWer1:RESUlts:ALLAcqs:STDDev? "TONLoss"` might return `4.25`, indicating the standard deviation value of T-On Energy loss for all acquisitions.

POWer:POWer<x>:RESUlts:CURRentacq:F1MAG? (Query Only)

This command queries the first harmonics magnitude value for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:F1MAG? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer1:RESUlts:CURRentacq:F1MAG? "harmonics"` might return 1.4151834770090, indicating the value of the first harmonics magnitude for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:F3MAG? (Query Only)

This command queries the third harmonics magnitude value for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:F3MAG? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer1:RESUlts:CURRentacq:f3MAG? "harmonics"` might return 234.0187140104806E-6, indicating the value of third harmonics magnitude for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:FREQUENCY? (Query Only)

This command queries the fundamental frequency for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:FREQUENCY? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer1:RESUlts:CURRentacq:FREQUENCY? "harmonics"` might return 100.0067656931537E+3, indicating the fundamental frequency for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:IRMS? (Query Only)

This command queries the RMS current value for the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:IRMS? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer1:RESUlts:CURRentacq:IRMS? "harmonics"` might return 1.4149980733491, indicating the RMS current value for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:MAXimum? (Query Only)

This command queries the maximum value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:MAXimum?  {InputPwr|
Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|
Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|
Bpeak|Br|Hc|Hmax|IRipple|DeltaB|DeltaH|
Permeability|RDS|TRUEPWR|APPPWR|REPWR|PWRFACTOR|
PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|
TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|
TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|
LRIPRMS|LRIPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|
FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|
HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples `POWer:POWer1:RESUlts:CURRentacq:MAXimum? "TONEnrg"` might return `32.8`, indicating the maximum value of Ton Energy for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:MEAN? (Query Only)

This command queries the mean value of the current acquisition for the measurement parameter of the specified power measurement <x>.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:MEAN? {InputPwr|
Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|
Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|
Br|Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|
APPPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|
VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|
TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|
SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|
AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<QString> = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples `POWer:POWer2:RESUlts:CURRentacq:MEAN? "TruePwr"` might return `42.6097255943E-2`, indicating the mean value of true power for the current acquisition of power measurement 2.

POWer:POWer<x>:RESUlts:CURRentacq:MINimum? (Query Only)

This command queries the minimum value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:MINimum? {InputPwr|
Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|
Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|
Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|
APPPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|
VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|
TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|
SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|
PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "`SWRIPPKPK`" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples `POWer:POWer1:RESUlts:CURRentacq:MINimum? "TruePwr"` might return `4.17829019093E-9`, indicating the minimum value of true power for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:PK2PK? (Query Only)

This command queries the peak-to-peak value of the current acquisition for the measurement parameter in the specified power measurement number.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:PK2PK? {InputPwr|
Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|
Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|
Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|APPPWR|
REPWR|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|
TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|
TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|SWRIPRMS|
SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|
HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "`LRIPPKPK`" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

`POWer:POWer1:RESUlts:CURRentacq:PK2PK? "TONLoss"` might return `9.91`, indicating the peak-to-peak value of Ton Energy for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:POHCL? (Query Only)

This command queries the limit of partial odd harmonic current for the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

`POWer:POWer<x>:RESUlts:CURRentacq:POHCL? "harmonics"`

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples

`POWer:POWer1:RESUlts:CURRentacq:POHCL? "harmonics"` might return `251.3529788962128E-3`, indicating the limit of partial odd harmonic current for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:POHCM? (Query Only)

This command queries the measured value of partial odd harmonic current for the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:POHCM? "harmonics"`

Examples `POWer:POWer1:RESUlts:CURRentacq:POHCM? "harmonics"` might return 515.422617782020E-6, indicating the measured value of partial odd harmonic current for the power measurement badge 1.

POWer:POWer<x>:RESUlts:CURRentacq:POHCS? (Query Only)

This command queries the status of partial odd harmonic current for the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:POHCS? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer1:RESUlts:CURRentacq:POHCS? "harmonics"` might return `Pass`, indicating the status of partial odd harmonic current for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:POPUlation? (Query Only)

This command queries the population (number of complete cycles) of the current acquisition for the measurement parameter in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:POPUlation?  {InputPwr|
Output1Pwr|Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|
Efficiency3|TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|
Hc|Hmax|IRipple|DeltaB|DeltaH|Permeability|RDS|TRUEPWR|
APPPWR|REPWR|PWRFACTOR|PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|
VRMS|TONENRG|TONLOSS|TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|
TTLLOSS|TTLENRG|DVBYDT|DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|
SWRIPRMS|SWRIPPKPK|PRIOD|FREQ|PDUTY|NDUTY|PPULSE|NPULSE|
AMPL|PKPK|HIGH|LOW|MAX|MIN|INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples

`POWer:POWer1:RESUlts:CURRentacq:POPUlation? "CondEnrg"` might return `9.24E+4`, indicating the population (number of complete cycles) of conduction energy for the current acquisition.

POWer:POWer<x>:RESUlts:CURRentacq:RMS? (Query Only)

This command queries the RMS value of the source selected for the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:RMS? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer1:RESUlts:CURRentacq:RMS? "harmonics"` might return 1.4143420437461, indicating the RMS value of the source selected for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:STATUS? (Query Only)

This command queries the status of the measurement for the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:STATUS? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer1:RESUlts:CURRentacq:STATUS? "harmonics"` might return `Pass`, indicating that the measurement has passed according to the standard for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:STDDev? (Query Only)

This command queries the standard deviation value of the current acquisition for the measurement parameter in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:RESUlts:CURRentacq:STDDev?
{InputPwr|Output1Pwr|
Output2Pwr|Output3Pwr|Efficiency1|Efficiency2|Efficiency3|
TotalEfficiency|INDUCT|IVSINTV|MAGLOSS|Bpeak|Br|Hc|Hmax|IRipple|
DeltaB|DeltaH|Permeability|RDS|TRUEPWR|APPPWR|REPWR|PWRFACTOR|
PHASE|PWRFREQ|ICFACTOR|VCFACTOR|IRMS|VRMS|TONENRG|TONLOSS|
TOFFENRG|TOFFLOSS|CONDENRG|CONDLOSS|TTLLOSS|TTLENRG|DVBYDT|
DIBYDT|SOAHITSCNT|LRIPRMS|LRIPPKPK|SWRIPRMS|SWRIPPKPK|PRIOD|
FREQ|PDUTY|NDUTY|PPULSE|NPULSE|AMPL|PKPK|HIGH|LOW|MAX|MIN
|INRUSH|CAPACITANCE|OUTPUT1|
OUTPUT2|OUTPUT3|OUTPUT4|OUTPUT5|OUTPUT6|OUTPUT7|
GAINCROSSOVERFREQ|PHASECROSSOVERFREQ|GM|PM|
MAXPSRR|MAXPSRRFREQ|MINPSRR|MINPSRRFREQ}
```

NOTE. *Above entries are <QSTring> entries, and must be entered in enclosing quotes.*

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = the measurement result that you want to return from the specified power measurement number. Available results depend on the power measurement being taken in the specified measurement number. The valid <Qstring> arguments are:

For Efficiency measurement, the parameters are "InputPwr", "Output1Pwr", "Output2Pwr", "Output3Pwr", "Efficiency1", "Efficiency2", "Efficiency3", "TotalEfficiency".

For Inductance measurement, the parameter is "INDUCT".

For IVSIntegralV measurement, the parameter is "IVSINTV".

For Magnetic Loss measurement, the parameter is "MAGLOSS".

For Magnetic Property measurement, the parameters are "Bpeak", "Br", "Hc", "Hmax", "IRipple", "DeltaB", "DeltaH", "Permeability".

For RDSon measurement, the parameter is "RDS".

For Power Quality measurement, the parameters are "TRUEPWR", "APPPWR", "REPWR", "PWRFACTOR", "PHASE", "PWRFREQ", "ICFACTOR", "VCFACTOR", "IRMS", "VRMS".

For Switching Loss measurement, the parameters are "TONENRG", "TONLOSS", "TOFFENRG", "TOFFLOSS", "CONDENRG", "CONDLOSS", "TTLLOSS", "TTLENRG".

"DVBYDT" is the parameter for the dV by dt measurement.

"DIBYDT" is the parameter for the dI by dt measurement.

"SOAHITSCNT" is the parameter for the SOA measurement.

"LRIPRMS" and "LRIPPKPK" are the parameters for the Line Ripple measurement.

"SWRIPRMS" and "SWRIPPKPK" are the parameters for the Switching Ripple measurement.

"PRIOD" is the parameter for the Cycle Period measurement.

"FREQ" is the parameter for the Cycle Frequency measurement.

"PDUTY" is the parameter for the Positive Duty Cycle measurement.

"NDUTY" is the parameter for the Negative Duty Cycle measurement.

"PPULSE" is the parameter for the Positive Pulse Width measurement.

"NPULSE" is the parameter for the Negative Pulse Width measurement.

"AMPL" is the parameter for the Cycle Amplitude measurement.

"PKPK" is the parameter for the Cycle Peak–Peak measurement.

"HIGH" is the parameter for the Cycle Top measurement.

"LOW" is the parameter for the Cycle Base measurement.

"Max" is the parameter for the Cycle Max measurement.

"MIN" is the parameter for the Cycle Min measurement.

"INRUSH" is the parameter for the Inrush Current measurement.

"CAPACITANCE" is the parameter for the Input Capacitance measurement.

"OUTPUT1" - "OUTPUT7" are the parameters for the Turn On Time and Turn Off Time measurements.

"GAINCROSSOVERFREQ", "PHASECROSSOVERFREQ", "GM", "PM" are the parameters for the Control Loop Response measurement.

"MAXPSRR", "MAXPSRRFREQ", "MINPSRR", "MINPSRRFREQ" are the parameters for the PSRR measurement

Examples `POWer:POWer2:RESUlts:CURRentacq:STDDev? "TONLoss"` might return `1.31E+2`, indicating the standard deviation value of Ton Energy for current acquisition of power measurement 2.

POWer:POWer<x>:RESUlts:CURRentacq:THDF? (Query Only)

This command queries the total harmonic distortion (fundamental) value for the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:THDF? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer1:RESUlts:CURRentacq:THDF? "harmonics"` might return `96.9846996670887E-3`, indicating the value of total harmonic distortion (fundamental) for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:THDR? (Query Only)

This command queries the total harmonic distortion (RMS) value for the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:THDR? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer1:RESUlts:CURRentacq:THDR? "harmonics"` might return 107.1384597967292E-3, indicating the value of total harmonic distortion (fundamental) for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:TRPWR? (Query Only)

This command queries the true power value for the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:TRPWR? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer1:RESUlts:CURRentacq:TRPWR? "harmonics"` might return `2.0002612633993`, indicating the true power value for power measurement 1.

POWer:POWer<x>:RESUlts:CURRentacq:VRMS? (Query Only)

This command queries the RMS voltage value for the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax `POWer:POWer<x>:RESUlts:CURRentacq:VRMS? "harmonics"`

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

Examples `POWer:POWer3:RESUlts:CURRentacq:VRMS? "harmonics"` might return `1.4117680233354`, indicating the RMS voltage value for the power measurement 3.

POWer:POWer<x>:SEQSETup (No Query Form)

This command sets up the instrument's horizontal, vertical, and trigger parameters to optimize for taking the specified power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SEQSETup Execute
```

Arguments

`POWer<x>` is the Power measurement identifier number. The number must be for a power measurement that requires a single sequence acquisition.

`RUN` sets the measurement to run an acquisition and acquire data for the specified single sequence power measurement.

Examples `POWer:POWer3:SEQuence Execute` runs the power measurement 3 sequence setup.

POWer:POWer<x>:SEQuence

This command sets or queries the run state of a single sequence power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SEQuence {RUN|RERUN}
POWer:POWer<x>:SEQuence?
```

Arguments

`POWer<x>` is the Power measurement identifier number. The number must be for a power measurement that requires a single sequence acquisition.

`RUN` sets the measurement to run an acquisition and acquire data for the specified single sequence power measurement.

`RERUN` sets the measurement to rerun an acquisition and acquire data for the specified single sequence power measurement.

Examples `POWer:POWer3:SEQuence RUN` sets power measurement 3 to run the measurement.

`POWer:POWer1:SEQuence?` might return `POWer:POWer1:SEQuence RERUN`, indicating that the current state of the power measurement is rerun.

POWer:POWer<x>:SOA:ISOURce

This command sets or queries the current source for SOA measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SOA:ISOURce {CH<x>|MATH<x>|REF<x>}
POWer:POWer<x>:SOA:ISOURce?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels onyour instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:SOA:ISOURce CH2` sets the current source for the SOA measurement 1 as channel 2.

POWer:POWer<x>:SOA:POINT

This command sets or queries the X or Y coordinate value for an SOA mask of a specified power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SOA:POINT <QString> <NR1>
POWer:POWer<x>:SOA:POINT <QString>?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<QString>` = 1X | 2X | 3X | 4X | 5X | 6X | 7X | 8X | 9X | 10X | 11X | 12X | 13X | 14X | 15X | 16X | 17X | 18X | 19X | 20X | 21X | 22X | 23X | 24X | 25X | 26X | 27X | 28X | 29X | 30X | 31X | 32X | 1Y | 2Y | 3Y | 4Y | 5Y | 6Y | 7Y | 8Y | 9Y | 10Y | 11Y | 12Y | 13Y | 14Y | 15Y | 16Y | 17Y | 18Y | 19Y | 20Y | 21Y | 22Y | 23Y | 24Y | 25Y | 26Y | 27Y | 28Y | 29Y | 30Y | 31Y | 32Y

`<NR2>` sets the specified SOA mask X or Y point value, as a floating number, in the range from –5000 to 5000.

Examples

`POWer:POWer1:SOA:POINT6Y 0.125` sets the #6Y coordinate value of the SOA mask of power measurement 1 to 0.125.

`POWer:POWer3:SOA:POINT12X?` might return `POWer:POWer3:SOA:POINT12X 1.435`, indicating that the #12X coordinate value of the SOA mask of power measurement 3 is 1.435.

POWer:POWer<x>:SOA:RECAllmask

This command recalls or queries the recall mask file name in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SOA:RECAllmask
POWer:POWer<x>:SOA:RECAllmask?
```

Examples

`POWer:POWer1:SOA:RECAllmask?` might return `Tek000.msk`, indicating the file name of the mask that will be recalled.

POWer:POWer<x>:SOA:RECAllmask:FILEName

This command sets or queries the file name for saving SOA mask file name in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SOA:RECAllmask:FILEName
POWer:POWer<x>:SOA:RECAllmask:FILEName?
```

Examples
```
POWer:POWer1:SOA:RECAllmask:FILEName
```

POWer:POWer<x>:SOA:SAVemask

This command saves the mask file as per the name configured and at the configured path or queries the mask file name, path, and file type for the SOA measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SOA:SAVemask
POWer:POWer<x>:SOA:SAVemask?
```

Related Commands POWer:POWer<x>:SOA:SAVemask:FILEName

Examples `POWer:POWer1:SOA:SAVemask` saves the mask file of SOA measurement as the configured file name at the configured path.

POWer:POWer<x>:SOA:SAVemask:AUTOINCrement

This command sets or queries the state of auto-increment for saved SOA mask file names in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SOA:SAVemask:AUTOINCrement
POWer:POWer<x>:SOA:SAVemask:AUTOINCrement?
```

POWer:POWer<x>:SOA:SAVemask:FILEName

This command sets or queries the mask file name for SOA measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SOA:SAVemask:FILEName
POWer:POWer<x>:SOA:SAVemask:FILEName?
```

Examples `POWer:POWer1:SOA:SAVemask:FILEName "Tek001.msk"` sets the mask file name for SOA measurement as Tek001.msk.

POWer:POWer<x>:SOA:SAVemask:FOLDer

This command sets or queries the mask file folder path for SOA measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SOA:SAVemask:FOLDer
POWer:POWer<x>:SOA:SAVemask:FOLDer?
```

Examples `POWer:POWer1:SOA:SAVemask:FOLDer "C:"` sets the mask file folder path for SOA measurement as C drive.

POWer:POWer<x>:SOA:VSOURce

This command sets or queries the voltage source for SOA measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SOA:VSOURce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:SOA:VSOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:SOA:VSOURce CH1` sets the voltage source for SOA measurement as channel 1.

POWer:POWer<x>:SWITCHINGLOSS:DEVICEType

This command sets or queries the conduction calculation method for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:DEVICEType {MOSFET | BJT}
POWer:POWer<x>:SWITCHINGLOSS:DEVICEType?
```

Examples `POWer:POWer1:SWITCHINGLOSS:DEVICEType MOSFET` sets the conduction calculation method as mosfet for switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:GATESOurce

This command sets or queries the gate voltage (V_g) for the switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:GATESOurce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:SWITCHINGLOSS:GATESOurce?
```

Arguments `POwer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:SWITCHINGLOSS:GATESOurce MATH1` sets the gate voltage (Vg) source for switching loss measurement of power measurement 1 as MATH1.

`POWer:POWer2:SWITCHINGLOSS:VSOURce?` might return MATH3 indicating the gate voltage (Vg) for switching loss measurement of power measurement 2.

POWer:POWer<x>:SWITCHINGLOSS:ILEVELAbs

This command sets or queries the current level (Ton-Start & Stop) in absolute units for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SWITCHINGLOSS:ILEVELAbs <NR1>
POWer:POWer<x>:SWITCHINGLOSS:ILEVELAbs?
```

Related Commands POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from –100 to 100

Examples `POWer:POWer1:SWITCHINGLOSS:ILEVELAbs 1.2` sets the current level (Ton-Start & Stop) value as 1.2 for the switching loss measurement in the power measurement badge 1.

POWer:POWer<x>:SWITCHINGLOSS:ILEVELPct

This command sets or queries the current level (Ton-Start & Stop) in percentage for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SWITCHINGLOSS:ILEVELPct <NR1>
POWer:POWer<x>:SWITCHINGLOSS:ILEVELPct?
```

Related Commands POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 0.0001 to 90

Examples `POWer:POWer1:SWITCHINGLOSS:ILEVELPct 6` sets the current level (Ton-Start & Stop) value as 6 percentage for the switching loss measurement in the power measurement badge 1.

POWer:POWer<x>:SWITCHINGLOSS:ISOURce

This command sets or queries the current source for the switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:ISOURce {CH<x> | MATH<x> |
REF<x>}
POWer:POWer<x>:SWITCHINGLOSS:ISOURce?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`CH<x>` = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

`MATH<x>` = A math waveform specifier; <x> is ≥1.

`REF<x>` = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:SWITCHINGLOSS:ISOURce CH2` sets the current source for the switching loss measurement in the specified power measurement badge as CH2.

`POWer:POWer2:SWITCHINGLOSS:ISOURce?` might return CH1 indicating the current source for switching loss measurement of Power 2 power measurement badge.

POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts

This command sets or queries the level units for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts {PERCent | ABSolute}
POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`PERCent` to set the High, Mid, and Low reference levels in percentage.

`ABSolute` to set the High, Mid, and Low reference levels to specific signal levels.

Examples

`POWer:POWer1:SWITCHINGLOSS:LEVELUNIts ABSolute` sets the level units as Absolute for switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:RDSOn

This command sets or queries the RDS(on) value for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:RDSOn <NR1>
POWer:POWer<x>:SWITCHINGLOSS:RDSOn?
```

Related Commands

POWer:POWer<x>:SWITCHINGLOSS:DEVICEType

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 0 to 100

Examples

`POWer:POWer1:SWITCHINGLOSS:RDSOn 2` sets the switching loss RDSOn value as 2 for switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType

This command sets or queries the configuration type for the switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType {SMPS | PFC | FLYBACK}
POWer:POWer<x>:SWITCHINGLOSS:SWLCONFIGType?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`SMPS`: Select this option in case of signals without noise and ringing. The V_g source is not required. Select V_g souce (Source 3), in case of noisy signal.

`PFC`: Select this option when input DUT signals are from Power Factor Correction Circuit. For this case, V_g source is mandatory.

`FLYBACK`: Select this option when input signals are ringing. This option does not require a V_g source.

Examples `POWer:POWer1:SWITCHINGLOSS:SWLCONFIGType PFC` sets the configuration type as PFC for the switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:VCESat

This command sets or queries the value for the VCE(sat) value for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SWITCHINGLOSS:VCESat <NR1>
POWer:POWer<x>:SWITCHINGLOSS:VCESat?
```

Related Commands POWer:POWer<x>:SWITCHINGLOSS:DEVICEType

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from 0.001 to 100

Examples POWer:POWer1:SWITCHINGLOSS:VCESat 6 sets VCE(sat) value as 6 for switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:VGLevel

This command sets or queries the gate voltage value (V_g Level Ton-Start) for the switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SWITCHINGLOSS:VGLevel <NR1>
POWer:POWer<x>:SWITCHINGLOSS:VGLevel?
```

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from –100 to 100

Examples POWer:POWer1:SWITCHINGLOSS:VGLevel 1.2 sets the gate voltage value (V_g Level Ton-Start) for the switching loss measurement of the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs

This command sets or queries the voltage level (Ton-Start & Stop) in absolute units for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs <NR1>
POWer:POWer<x>:SWITCHINGLOSS:VLEVELAbs?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from –100 to 100

Examples `POWer:POWer1:SWITCHINGLOSS:VLEVELAbs 2` sets the voltage level (Ton-Start & Stop) value as 2 for the switching loss measurement in the power measurement badge 1.

POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct

This command sets or queries the voltage level (Ton-Start & Stop) in percentage for switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct <NR1>
POWer:POWer<x>:SWITCHINGLOSS:VLEVELPct?
```

Related Commands POWer:POWer<x>:SWITCHINGLOSS:LEVELUNIts

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<NR1>` ranges from 0.0001 to 90

Examples `POWer:POWer1:SWITCHINGLOSS:VLEVELPct 7` sets the voltage level (Ton-Start & Stop) value as 7 percentage for the switching loss measurement in the power measurement badge 1.

POWer:POWer<x>:SWITCHINGLOSS:VSOURce

This command sets or queries the voltage source for the switching loss measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SWITCHINGLOSS:VSOURce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:SWITCHINGLOSS:VSOURce?
```

Arguments POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples `POWer:POWer1:SWITCHINGLOSS:VSOURce REF1` sets the voltage source for the switching loss measurement in the specified power measurement badge as REF1.

`POWer:POWer2:SWITCHINGLOSS:VSOURce?` might return REF3 indicating the voltage source for switching loss measurement of Power 2 power measurement badge.

POWer:POWer<x>:SWITCHINGRIPPLE:INPUTSOurce

This command sets or queries the input source for switching ripple measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SWITCHINGRIPPLE:INPUTSOurce {CH<x> | MATH<x> | REF<x>}
POWer:POWer<x>:SWITCHINGRIPPLE:INPUTSOurce?
```

Arguments POWer<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

CH<x> = A channel specifier; <x> is 1 through 8 and is limited by the number of FlexChannels in your instrument.

MATH<x> = A math waveform specifier; <x> is ≥1.

REF<x> = A reference waveform specifier; <x> is ≥1.

Examples POWer:POWer1:SWITCHINGRIPPLE:INPUTSOurce CH5 sets the input source for switching ripple measurement as CH5 for the power measurement badge Power 1.

POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency

This command sets or queries the switching frequency for switching ripple measurement in the specified power measurement number. The power measurement number is specified by x.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency <NR1>
POWer:POWer<x>:SWITCHINGRIPPLE:LFREQuency?
```

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

<NR1> ranges from 50 to 1000000

Examples POWer:POWer1:SWITCHINGRIPPLE:SFREQuency 60000 sets the frequency present for line ripple measurement as 60000 Hz.

POWer:POWer<x>:TURNOFFtime:FREQuency

This command sets or queries the input frequency used by the AC or DC converter of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:TURNOFFtime:FREQuency <NR3>
POWer:POWer<x>:TURNOFFtime:FREQuency?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the frequency, in Hertz, from 1 Hz to 500 Hz.

Examples `POWer:POWer1:TURNOFFtime:FREQuency 50` sets the frequency value of Turn Off Time measurement 1 to 50 Hz.

`POWer:POWer4:TURNOFFtime:FREQuency?` might return `POWer:POWer4:TURNOFFtime:FREQuency 350`, indicating that the frequency value of Turn Off Time measurement 4 is set to 350 Hz.

POWer:POWer<x>:TURNOFFtime:INPUTLEVel

This command sets or queries the input voltage level of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:TURNOFFtime:INPUTLEVel <NR3>
POWer:POWer<x>:TURNOFFtime:INPUTLEVel?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the voltage level, in volts, from –500 V to 500 V.

Examples `POWer:POWer1:TURNOFFtime:INPUTLEVel -350` sets the input voltage level of Turn Off Time measurement 1 to –350 V.

`POWer:POWer4:TURNOFFtime:INPUTLEVel?` might return `POWer:POWer4:TURNOFFtime:INPUTLEVel 200`, indicating that the input voltage level value of Turn Off Time measurement 4 is set to 200 V.

POWer:POWer<x>:TURNOFFtime:INPUTSOurce

This command sets or queries the input source of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:INPUTSOurce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:INPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer1:TURNOFFtime:INPUTSOurce CH4` sets the input source of Turn Off Time measurement 1 to Channel 4.

`POWer:POWer6:TURNOFFtime:INPUTSOurce?` might return `POWer:POWer6:TURNOFFtime:INPUTSOurce MATH2`, indicating that the input source of Turn Off Time measurement 6 is set to Math waveform 2.

POWer:POWer<x>:TURNOFFtime:MAXTIMe

This command sets or queries the maximum turn off time of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:MAXTIMe <NR3>
POWer:POWer<x>:TURNOFFtime:MAXTIMe?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the maximum time value, in seconds, in the range 1 second to 500 seconds.

Examples

POWer:POWer1:TURNOFFtime:MAXTIMe 70 sets the maximum time value of Turn Off Time measurement 1 to 70 seconds.

POWer:POWer5:TURNOFFtime:MAXTIMe? might return POWer:POWer5:TURNOFFtime:MAXTIMe 120, indicating that the maximum time value of Turn Off Time measurement 5 is 120 seconds.

POWer:POWer<x>:TURNOFFtime:MAXVoltage

This command sets or queries the maximum voltage of the specified Turn OffTime measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:MAXVoltage <NR3>
POWer:POWer<x>:TURNOFFtime:MAXVoltage?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the maximum voltage in the range 1 V to 500 V.

Examples

POWer:POWer1:TURNOFFtime:MAXVoltage 50 sets the maximum voltage value of Turn Off Time measurement 1 to 50 volts.

POWer:POWer4:TURNOFFtime:MAXVoltage? might return POWer:POWer4:TURNOFFtime:MAXVoltage 12, indicating that the maximum voltage value of Turn Off Time measurement 4 is 12 volts.

POWer:POWer<x>:TURNOFFtime:NUMOUTputs

This command sets or queries the number of outputs of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:NUMOUTputs
{ONE|TWO|THREE|FOUR|FIVE| SIX|SEVEN}
POWer:POWer<x>:TURNOFFtime:NUMOUTputs?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce

POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage

POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce

POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage

POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce

POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage

POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce

POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage

POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce

POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage

POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce

POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage

POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce

POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage

Arguments

`POwer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`ONE` through `SEVEN` sets the number of outputs for the Turn Off Time measurement.

Examples

`POWer:POWer1:TURNOFFtime:NUMOUTputs TWO` sets the number of outputs of Turn Off Time measurement 1 to two.

`POWer:POWer3:TURNOFFtime:NUMOUTputs?` might return `POWer:POWer3:TURNOFFtime:NUMOUTputs FOUR`, indicating that the number of outputs of Turn Off Time measurement 3 is set to four inputs.

POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce

This command sets or queries the output 1 source of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer2:TURNOFFtime:OUTPUT1SOURce CH2` sets the output 1 source of Turn Off Time measurement 2 to Channel 2.

`POWer:POWer1:TURNOFFtime:OUTPUT1SOURce?` might return `POWer:POWer1:TURNOFFtime:OUTPUT1SOURce CH4`, indicating that the output 1 source of Turn Off Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage

This command sets or queries the output 1 voltage level of the of the specified Turn Off Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT1VOLTage?
```

Related Commands POWer:POWer<x>:TURNOFFtime:NUMOUTputs

POWer:POWer<x>:TURNOFFtime:OUTPUT1SOURce

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples POWer:POWer7:TURNOFFtime:OUTPUT1VOLTage -2550 sets the output 1 voltage value of Turn Off Time power measurement 7 to –2,550 volts.

POWer:POWer3:TURNOFFtime:OUTPUT1VOLTage? might return POWer:POWer3:TURNOFFtime:OUTPUT1VOLTage 1000, indicating that the output 1 voltage value of Turn Off Time power measurement 3 is set to 1,000 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce

This command sets or queries the output 2 source of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce?
```

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples POWer:POWer3:TURNOFFtime:OUTPUT2SOURce CH2 sets the output 2 source of Turn Off Time measurement 3 to Channel 2.

POWer:POWer5:TURNOFFtime:OUTPUT2SOURce? might return POWer:POWer5:TURNOFFtime:OUTPUT2SOURce CH8, indicating that the output 2 source of Turn Off Time measurement 5 is Channel 8.

POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage

This command sets or queries the output 2 voltage level of the specified Turn Off Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT2VOLTage?
```

Related Commands POWer:POWer<x>:TURNOFFtime:NUMOUTputs

POWer:POWer<x>:TURNOFFtime:OUTPUT2SOURce

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples POWer:POWer3:TURNOFFtime:OUTPUT2VOLTage -550 sets the output 2 voltage value of Turn Off Time power measurement 3 to –550 volts.

POWer:POWer1:TURNOFFtime:OUTPUT2VOLTage? might return POWer:POWer1:TURNOFFtime:OUTPUT2VOLTage 100, indicating that the output 2 voltage value of Turn Off Time power measurement 1 is set to 100 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce

This command sets or queries the output 3 source of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNOFFtime:OUTPUT3SOURce CH8 sets the output 3 source of Turn Off Time measurement 3 to Channel 8.

POWer:POWer1:TURNOFFtime:OUTPUT3SOURce? might return POWer:POWer1:TURNOFFtime:OUTPUT3SOURce CH4, indicating that the output 3 source of Turn Off Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage

This command sets or queries the output 3 voltage level of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT3VOLTage?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs

POWer:POWer<x>:TURNOFFtime:OUTPUT3SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer4:TURNOFFtime:OUTPUT3VOLTage -50` sets the output 3 voltage value of Turn Off Time power measurement 4 to –50 volts.

`POWer:POWer8:TURNOFFtime:OUTPUT3VOLTage?` might return `POWer:POWer8:TURNOFFtime:OUTPUT3VOLTage -200`, indicating that the output 3 voltage value of Turn Off Time power measurement 8 is set to –200 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce

This command sets or queries the output 4 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer3:TURNOFFtime:OUTPUT4SOURce CH2` sets the output 4 source of Turn Off Time measurement 3 is Channel 2.

`POWer:POWer10:TURNOFFtime:OUTPUT4SOURce?` might return `POWer:POWer10:TURNOFFtime:OUTPUT4SOURce REF4`, indicating that the output 4 source of Turn Off Time measurement 10 is Reference waveform 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage

This command sets or queries the output 4 voltage level of the specified Turn Off Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT4VOLTage?
```

Related Commands POWer:POWer<x>:TURNOFFtime:NUMOUTputs

POWer:POWer<x>:TURNOFFtime:OUTPUT4SOURce

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples `POWer:POWer8:TURNOFFtime:OUTPUT4VOLTage 50` sets the output 4 voltage value of Turn Off Time power measurement 8 to 50 volts.

`POWer:POWer4:TURNOFFtime:OUTPUT4VOLTage?` might return `POWer:POWer4:TURNOFFtime:OUTPUT4VOLTage 1000`, indicating that the output 4 voltage value of Turn Off Time power measurement 4 is set to 1,000 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce

This command sets or queries the output 5 source of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNOFFtime:OUTPUT5SOURce CH2 sets the output 5 source of Turn Off Time measurement 3 to Channel 2.

POWer:POWer11:TURNOFFtime:OUTPUT5SOURce? might return POWer:POWer11:TURNOFFtime:OUTPUT5SOURce CH4, indicating that the output 5 source of Turn Off Time measurement 11 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage

This command sets or queries the output 5 voltage level of the specified Turn Off Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT5VOLTage?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs

POWer:POWer<x>:TURNOFFtime:OUTPUT5SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer2:TURNOFFtime:OUTPUT5VOLTage -255 sets the output 5 voltage value of Turn Off Time power measurement 2 to –250 volts.

POWer:POWer3:TURNOFFtime:OUTPUT5VOLTage? might return POWer:POWer3:TURNOFFtime:OUTPUT5VOLTage 100, indicating that the output 5 voltage value of Turn Off Time power measurement 3 is set to 100 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce

This command sets or queries the output 6 source of the specified Turn Off Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer3:TURNOFFtime:OUTPUT6SOURce CH2` sets the output 6 source of Turn Off Time measurement 3 to Channel 2.

`POWer:POWer1:TURNOFFtime:OUTPUT6SOURce?` might return `POWer:POWer1:TURNOFFtime:OUTPUT6SOURce CH4`, indicating that the output 6 source of Turn Off Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage

This command sets or queries the output 6 voltage level of the specified Turn Off Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT6VOLTage?
```

Related Commands

POWer:POWer<x>:TURNOFFtime:NUMOUTputs

POWer:POWer<x>:TURNOFFtime:OUTPUT6SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer5:TURNOFFtime:OUTPUT6VOLTage 1550 sets the output 6 voltage value of Turn Off Time power measurement 5 to 1,550 volts.

POWer:POWer2:TURNOFFtime:OUTPUT6VOLTage? might return POWer:POWer2:TURNOFFtime:OUTPUT6VOLTage –100, indicating that the output 6 voltage value of Turn Off Time power measurement 2 is set to –100 volts.

POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce

This command sets or queries the output 7 source of the specified Turn Off Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNOFFtime:OUTPUT7SOURce CH2 sets the output 7 source of Turn Off Time measurement 3 is Channel 2.

POWer:POWer1:TURNOFFtime:OUTPUT7SOURce? might return POWer:POWer1:TURNOFFtime:OUTPUT7SOURce CH4, indicating that the output 7 source of Turn Off Time measurement 1 is channel 4.

POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage

This command sets or queries the output 7 voltage level of the specified Turn Off Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage <NR2>
POWer:POWer<x>:TURNOFFtime:OUTPUT7VOLTage?
```

Related Commands POWer:POWer<x>:TURNOFFtime:NUMOUTputs

POWer:POWer<x>:TURNOFFtime:OUTPUT7SOURce

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples POWer:POWer2:TURNOFFtime:OUTPUT7VOLTage -2100 sets the output 7 voltage value of Turn Off Time power measurement 2 to –2,100 volts.

POWer:POWer3:TURNOFFtime:OUTPUT7VOLTage? might return POWer:POWer3:TURNOFFtime:OUTPUT7VOLTage 450, indicating that the output 7 voltage value of Turn Off Time power measurement 3 is set to 450 volts.

POWer:POWer<x>:TURNOFFtime:TYPE

This command sets or queries the type of AC/DC converter used in the specified Turn Off Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNOFFtime:TYPE {DCDC|ACDC}
POWer:POWer<x>:TURNOFFtime:TYPE?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

DCDC sets the measurement to use a DC to DC converter.

ACDC sets the measurement to use an AC to DC converter.

Examples

POWer:POWer1:TURNOFFtime:TYPE ACDC sets Turn Off time measurement 1 to use an AC to DC converter.

POWer:POWer5:TURNOFFtime:TYPE? might return POWer:POWer5:TURNOFFtime:TYPE DCDC, indicating that the converter type for Turn Off Time measurement 5 is set to use a DC to DC converter.

POWer:POWer<x>:TURNONtime:FREQuency

This command sets or queries the input frequency used by the AC or DC converter of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:FREQuency <NR3>
POWer:POWer<x>:TURNONtime:FREQuency?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the frequency, in Hertz, from 1 Hz to 500 Hz.

Examples

POWer:POWer1:TURNONtime:FREQuency 50 sets the frequency value of the AC/DC converter in Turn On Time measurement 1 to 50 Hz.

POWer:POWer4:TURNONtime:FREQuency? might return POWer:POWer4:TURNONtime:FREQuency 200, indicating that the frequency value of the AC/DC converter in Turn On Time measurement 4 is set to 200 Hz.

POWer:POWer<x>:TURNONtime:INPUTLEVel

This command sets or returns the input voltage level of the specified Turn On Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNONtime:INPUTLEVel <NR3>
POWer:POWer<x>:TURNONtime:INPUTLEVel?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the voltage level, in volts, from –500 V to 500 V.

Examples

`POWer:POWer12:TURNONtime:INPUTLEVel -350` sets the input voltage level of Turn On Time measurement 12 to –350 V.

`POWer:POWer4:TURNONtime:INPUTLEVel?` might return `POWer:POWer4:TURNONtime:INPUTLEVel 230`, indicating that the input voltage level value of Turn ON Time measurement 4 is set to 230 V.

POWer:POWer<x>:TURNONtime:INPUTSOurce

This command sets or queries the input source of the specified Turn On Time measurement.

Group Power

Syntax

```
POWer:POWer<x>:TURNONtime:INPUTSOurce
POWer:POWer<x>:TURNONtime:INPUTSOurce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` is the channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` is the Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> is the Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer1:TURNONtime:INPUTSOurce CH2 sets the input source of Turn On Time measurement 1 to Channel 2.

POWer:POWer2:TURNONtime:INPUTSOurce? might return POWer:POWer2:TURNONtime:INPUTSOurce MATH6, indicating that the input source of Turn On Time measurement 2 is Math waveform 6.

POWer:POWer<x>:TURNONtime:MAXTIMe

This command sets or returns the maximum turn on time of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:MAXTIMe <NR3>
POWer:POWer<x>:TURNONtime:MAXTIMe?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR3> is a floating point number that represents the maximum time value, in seconds, in the range 1 second to 500 seconds.

Examples

POWer:POWer3:TURNONtime:MAXTIMe 30 sets the maximum time value of Turn On Time measurement 3 to 30 seconds.

POWer:POWer5:TURNONtime:MAXTIMe? might return POWer:POWer5:TURNONtime:MAXTIMe 120, indicating that the maximum time value of Turn On Time measurement 5 is set to 120 seconds.

POWer:POWer<x>:TURNONTIME:MAXVoltage

This command sets or returns the maximum voltage setting of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNONTIME:MAXVoltage <NR3>
POWer:POWer<x>:TURNONTIME:MAXVoltage?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR3>` is a floating point number that represents the maximum voltage in the range 1 V to 500 V.

Examples `POWer:POWer1:TURNONtime:MAXVoltage 50` sets the maximum voltage value of Turn On Time measurement 1 to 50 volts.

`POWer:POWer4:TURNONtime:MAXVoltage?` might return `POWer:POWer4:TURNONtime:MAXVoltage 12`, indicating that the maximum voltage value of Turn On Time measurement 4 is set to 12 volts.

POWer:POWer<x>:TURNONtime:NUMOUTputs

This command sets or queries the number of outputs for the specified Turn On Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNONtime:NUMOUTputs
{ONE|TWO|THREE|FOUR|FIVE| SIX|SEVEN}
POWer:POWer<x>:TURNONtime:NUMOUTputs?
```

Related Commands POWer:POWer<x>:TURNONtime:INPUTSOurce

POWer:POWer<x>:TURNONtime:OUTPUT1SOURce

POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage

POWer:POWer<x>:TURNONtime:OUTPUT2SOURce

POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage

POWer:POWer<x>:TURNONtime:OUTPUT3SOURce

POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage

POWer:POWer<x>:TURNONtime:OUTPUT4SOURce

POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage

POWer:POWer<x>:TURNONtime:OUTPUT5SOURce

POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage

POWer:POWer<x>:TURNONtime:OUTPUT6SOURce

POWer:POWer<x>:TURNONtime:OUTPUT6VOLTage

POWer:POWer<x>:TURNONtime:OUTPUT7SOURce

POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`ONE` through `SEVEN` sets the number of outputs for the specified Turn On Time power measurement.

Examples

`POWer:POWer1:TURNONtime:NUMOUTputs TWO` sets the number of outputs of Turn On Time measurement 1 to two.

`POWer:POWer3:TURNONtime:NUMOUTputs?` might return `POWer:POWer3:TURNONtime:NUMOUTputs FOUR`, indicating that the number of outputs of Turn On Time measurement 3 is set to four inputs.

POWer:POWer<x>:TURNONtime:OUTPUT1SOURce

This command sets or queries the output 1 source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT1SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT1SOURce?
```

Arguments

`Power<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer5:TURNONtime:OUTPUT1SOURce CH2 sets the output 1 source of Turn On Time measurement 5 to Channel 2.

POWer:POWer1:TURNONTIME:OUTPUT1SOURce? might return POWer:POWer1:TURNONTIME:OUTPUT1SOURce CH4, indicating that the output 1 source of Turn On Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage

This command sets or queries the output 1 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT1VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs

POWer:POWer<x>:TURNONtime:OUTPUT1SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer1:TURNONtime:OUTPUT1VOLTage -2550 sets the output 1 voltage value of Turn On Time power measurement 1 to –2,550 volts.

POWer:POWer3:TURNONtime:OUTPUT1VOLTage? might return POWer:POWer3:TURNONtime:OUTPUT1VOLTage 1000, indicating that the output 1 voltage value of Turn On Time power measurement 3 is set to 1,000 volts.

POWer:POWer<x>:TURNONtime:OUTPUT2SOURce

This command sets or queries the output 2 source of the specified Turn On Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT2SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer2:TURNONtime:OUTPUT2SOURce CH2` sets the output 2 source of Turn On Time measurement 2 to Channel 2.

`POWer:POWer1:TURNONTIME:OUTPUT2SOURce?` might return `POWer:POWer1:TURNONTIME:OUTPUT2SOURce CH4`, indicating that the output 2 source of Turn On Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage

This command sets or queries the output 2 voltage level of the specified Turn On Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT2VOLTage?
```

Related Commands POWer:POWer<x>:TURNONtime:NUMOUTputs

POWer:POWer<x>:TURNONtime:OUTPUT2SOURce

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples POWer:POWer7:TURNONtime:OUTPUT2VOLTage -2550 sets the output 2 voltage value of Turn On Time power measurement 7 to –2,550 volts.

POWer:POWer1:TURNONtime:OUTPUT2VOLTage? might return POWer:POWer1:TURNONtime:OUTPUT2VOLTage 120, indicating that the output 2 voltage value of Turn On Time power measurement 1 is set to 120 volts.

POWer:POWer<x>:TURNONtime:OUTPUT3SOURce

This command sets or queries the output 3 source of the specified Turn On Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT3SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT3SOURce?
```

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples POWer:POWer3:TURNONtime:OUTPUT3SOURce CH2 sets the output 3 source of Turn On Time measurement 3 to Channel 2.

POWer:POWer13:TURNONTIME:OUTPUT3SOURce? might return POWer:POWer13:TURNONTIME:OUTPUT3SOURce REF3, indicating that the output 3 source of Turn On Time measurement 13 is Reference waveform 3.

POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage

This command sets or queries the output 3 voltage level of the specified Turn On Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT3VOLTage?
```

Related Commands POWer:POWer<x>:TURNONtime:NUMOUTputs

POWer:POWer<x>:TURNONtime:OUTPUT3SOURce

Arguments Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples POWer:POWer7:TURNONtime:OUTPUT3VOLTage -550 sets the output 3 voltage value of Turn On Time power measurement 7 to –550 volts.

POWer:POWer3:TURNONtime:OUTPUT3VOLTage? might return POWer:POWer3:TURNONtime:OUTPUT3VOLTage –200, indicating that the output 3 voltage value of Turn On Time power measurement 3 is set to –200 volts.

POWer:POWer<x>:TURNONtime:OUTPUT4SOURce

This command sets or queries the output 4 source of the specified Turn On Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT4SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT4SOURce?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

CH<x> A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNONtime:OUTPUT4SOURce CH2 sets the output 4 source of Turn On Time measurement 3 to Channel 2.

POWer:POWer1:TURNONTIME:OUTPUT4SOURce? might return POWer:POWer1:TURNONTIME:OUTPUT4SOURce MATH3, indicating that the output 4 source of Turn On Time measurement 1 is Math waveform 3.

POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage

This command sets or queries the output 4 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT4VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs

POWer:POWer<x>:TURNONtime:OUTPUT4SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

`POWer:POWer8:TURNONtime:OUTPUT4VOLTage 50` sets the output 4 voltage value of Turn On Time power measurement 8 to 50 volts.

`POWer:POWer2:TURNONtime:OUTPUT4VOLTage?` might return `POWer:POWer2:TURNONtime:OUTPUT4VOLTage 1000`, indicating that the output 4 voltage value of Turn On Time power measurement 2is set to 1,000 volts.

POWer:POWer<x>:TURNONtime:OUTPUT5SOURce

This command sets or queries the output 5 source of the specified Turn On Time measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT5SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT5SOURce?
```

Arguments

`POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

`POWer:POWer3:TURNONtime:OUTPUT5SOURce CH2` sets the output 5 source of Turn On Time measurement 3 to Channel 2.

`POWer:POWer1:TURNONTIME:OUTPUT5SOURce?` might return `POWer:POWer1:TURNONTIME:OUTPUT5SOURce CH4`, indicating that the output 5 source of Turn On Time measurement 1 is Channel 4.

POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage

This command sets or queries the output 5 voltage level of the specified Turn On Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT5VOLTage?
```

Related Commands POWer:POWer<x>:TURNONtime:NUMOUTputs

POWer:POWer<x>:TURNONtime:OUTPUT5SOURce

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`<NR2>` sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples `POWer:POWer2:TURNONtime:OUTPUT5VOLTage -255` sets the output 5 voltage value of Turn On Time power measurement 2 to –250 volts.

`POWer:POWer3:TURNONtime:OUTPUT5VOLTage?` might return `POWer:POWer3:TURNONtime:OUTPUT5VOLTage 100`, indicating that the output 5 voltage value of Turn On Time power measurement 3 is set to 100 volts.

POWer:POWer<x>:TURNONtime:OUTPUT6SOURce

This command sets or queries the output 6 source of the specified Turn On Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax
```
POWer:POWer<x>:TURNONtime:OUTPUT6SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT6SOURce?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

REF<x> A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

MATH<x> A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples

POWer:POWer3:TURNONtime:OUTPUT6SOURce CH2 sets the output 6 source of Turn On Time measurement 3 to Channel 2.

POWer:POWer8:TURNONTIME:OUTPUT6SOURce? might return POWer:POWer8:TURNONTIME:OUTPUT6SOURce CH4, indicating that the output 6 source of Turn On Time measurement 8 is set to Channel 4.

POWer:POWer<x>:TURNONtime:OUTPUT6VOLTage

This command sets or queries the output 6 voltage level of the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT6VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT6VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs

POWer:POWer<x>:TURNONtime:OUTPUT6SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer5:TURNONtime:OUTPUT6VOLTage 1550 sets the output 6 voltage value of Turn On Time power measurement 5 to 1,550 volts.

POWer:POWer2:TURNONtime:OUTPUT6VOLTage? might return POWer:POWer2:TURNONtime:OUTPUT6VOLTage –100, indicating that the output 6 voltage value of Turn On Time power measurement 2 is set to –100 volts.

POWer:POWer<x>:TURNONtime:OUTPUT7SOURce

This command sets or queries the output 7 source of the specified Turn On Time measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT7SOURce
{CH<x>|REF<x>|MATH<x>}
POWer:POWer<x>:TURNONtime:OUTPUT7SOURce?
```

Arguments `POWer<x>` is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

`CH<x>` A channel specifier in the range of 1 through 8 and is limited by the number of instrument input channels.

`REF<x>` A Reference waveform specifier ≥1. This is the equivalent of the number shown on a Reference waveform badge in the UI.

`MATH<x>` A Math waveform specifier ≥1. This is the equivalent of the number shown on a Math waveform badge in the UI.

Examples `POWer:POWer3:TURNONtime:OUTPUT6SOURce CH1` sets the output 7 source of Turn On Time measurement 3 to Channel 1.

`POWer:POWer7:TURNONTIME:OUTPUT6SOURce?` might return `POWer:POWer7:TURNONTIME:OUTPUT6SOURce REF4`, indicating that the output 7 source of Turn On Time measurement 7 is set to Reference waveform 4.

POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage

This command sets or queries the output 7 voltage level of the specified Turn On Time power measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage <NR2>
POWer:POWer<x>:TURNONtime:OUTPUT7VOLTage?
```

Related Commands

POWer:POWer<x>:TURNONtime:NUMOUTputs

POWer:POWer<x>:TURNONtime:OUTPUT7SOURce

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

<NR2> sets the output voltage value, in the range of –6,000 volts to +6,000 volts.

Examples

POWer:POWer12:TURNONtime:OUTPUT7VOLTage -2100 sets the output 7 voltage value of Turn On Time power measurement 12 to –2,100 volts.

POWer:POWer3:TURNONtime:OUTPUT7VOLTage? might return POWer:POWer3:TURNONtime:OUTPUT7VOLTage 450, indicating that the output 7 voltage value of Turn On Time power measurement 3 is set to 450 volts.

POWer:POWer<x>:TURNONtime:TYPE

This command sets or queries the type of AC/DC converter used in the specified Turn On Time power measurement.

Conditions

Requires option PWR or PS2.

Group

Power

Syntax

```
POWer:POWer<x>:TURNONtime:TYPE {DCDC|ACDC}
POWer:POWer<x>:TURNONtime:TYPE?
```

Arguments

Power<x> is the power measurement number. This is the equivalent of the number shown on a power measurement badge in the UI.

DCDC sets the measurement to use a DC to DC converter.

ACDC sets the measurement to use an AC to DC converter.

Examples

POWer:POWer3:TURNONtime:TYPE ACDC sets Turn On time measurement 3 to use an AC to DC converter.

POWer:POWer12:TURNONtime:TYPE? might return POWer:POWer12:TURNONtime:TYPE DCDC, indicating that the converter type for Turn On Time power measurement 12 is set to use a DC-DC converter.

POWer:POWer<x>:TYPe

This command sets or queries the measurement type of the specified power measurement number. If the measurement number does not exist, this command creates a new power measurement, assigns the specified measurement number to the new measurement, and then assigns the measurement type to the new measurement.

Conditions Requires option PWR or PS2.

Group Power

Syntax

```
POWer:POWer<x>:TYPe <Measurement Type>
POWer:POWer<x>:TYPe?
```

Arguments `Power<x>` is the power measurement number. This is the equivalent of the number shown in the UI for a power measurement badge.

`<Measurement Type>` = CYCLEAmp | CYCLEBase | CYCLEMAx | CYCLEMIn | CYCLEPKpk | CYCLETop | DIDT | DVDT | EFFICIENCY | FREQuency | HARMonics |IMPEDANCE| INDUCTANCE | INPUTCAP | INRUSHcurrent| | IVSINTEGRALV | LINERIpple | MAGNETICLOSS | MAGPROPERTY | NDUTYCycle | NPULSEWidth | PDUTYCycle | PERIod | POWERQUALity | PPULSEWidth | RDSON | SOA | SWITCHINGLOss | SWITCHINGRIpple | TURNOFFtime | TURNONtime| CLRESPONSE | PSRR

Examples `POWer:POWer6:TYPe INDUCTANCE` sets power measurement 6 to be an Inductance measurement.

`POWer:POWer1:TYPe?` might return `INPUTCAP`, indicating that power measurement 1 is an Input Capacitance measurement.

*PSC

This command sets and queries the power-on status flag that controls the automatic power-on handling of the DESER, SRER, and ESER registers. When *PSC is true, the DESER register is set to 255 and the SRER and ESER registers are set to 0 at power-on. When *PSC is false, the current values in the DESER, SRER, and ESER registers are preserved in nonvolatile memory when power is shut off and are restored at power-on.

Group Status and Error

Syntax

```
*PSC {<NR1>|OFF|ON}
*PSC?
```

Related Commands

DESE

*ESE

FACtory

*RST

*SRE

Arguments

<NR1> = 0 sets the power-on status clear flag to false, disables the power-on clear and allows the instrument to possibly assert SRQ after power-on; any other value sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.

OFF sets the power-on status clear flag to false, disables the power-on clear and allows the instrument to possibly assert SRQ after power-on.

ON sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.

Examples

*PSC 0 sets the power-on status clear flag to false.

*PSC? might return 1 to indicate that the power-on status clear flag is set to true.

*PUD

This command sets or queries a string of Protected User Data. This data is protected by the PASSWord command. You can modify it only by first entering the correct password. This password is not necessary to query the data.

Group

Status and Error

Syntax

```
*PUD {<Block>|<QString>}
*PUD?
```

Related Commands

PASSWord

Arguments

<Block> is a block containing up to 100 characters.

<QString> is a string containing up to 100 characters.

Examples `*PUD #229This instrument belongs to me` stores the string "This instrument belongs to me" in the user protected data area.

`*PUD?` might return `#221PROPERTY OF COMPANY X`.

RECAll:MASK (No Query Form)

This command recalls a saved mask definition from a Mask File. File suffixes can be xml or msk. If the specified mask test already exists the mask associated with that mask test will be replaced, otherwise a new mask test is created.

Conditions Requires option MTM.

Group Mask

Syntax `RECAll:MASK <source file>,MASK<x>`

Arguments `MASK<x>` is the destination mask.

`<source file>` is the source file. The file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or "" or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `RECALL:MASK "MaskWD17",MASK1` recalls the mask stored in the file named MASKWD17 and stores it in mask 1. If mask 1 already exists then its mask will be replaced.

RECAll:SESsion (No Query Form)

Restores the state of the instrument, including reference waveforms, from a saved session file.

Group Save and Recall

Syntax `RECAll:SESsion <file_path>`

Arguments `<file_path>` is the file path that specifies the location of the specified instrument session file.

If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a drive designator (such as C), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Returns Recalls the instrument session from the specified session file.

Examples `RECALL:SESSION "TEK00000.TSS"` recalls the setup from the file TEK00000.TSS in the current working directory.

RECAll:SETUp (No Query Form)

This command (no query form) returns stored or factory settings to the instrument from a copy of the settings stored in memory. This command performs the same function as selecting Recall from the File menu, and then choosing the Setup button.

Group Save and Recall

Syntax `RECAll:SETUp {FACtory|<file_path>}`

Related Commands FACtory

*RST

Arguments `FACtory` restores the factory setup. Performs the same operation as the :FACtory command.

`<file_path>` specifies a location for an instrument setup file. <file path> is a quoted string that defines the file name and path. If a file name or path is specified,

the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

`RECALL:SETUP FACTORY` recalls (and makes current) the instrument setup to its factory defaults.

`RECALL:SETUP "TEK00000.SET"` recalls the setup from the file TEK00000.SET in the default directory for setups.

RECAll:WAVEform (No Query Form)

This command recalls a stored waveform to a reference memory location.

Group

Save and Recall

Syntax

`RECAll:WAVEform <source file>,<destination>`

Arguments

`<source file>` is the source file. The file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

`<destination>` is REF<x> which specifies a reference to create from the recalled waveform data file.

Examples

`RECALL:WAVEFORM "TEK00000.ISF",REF1` recalls the waveform stored in the file named TEK00000.ISF from the current directory to reference 1.

REF:ADDNew (No Query Form)

This command adds the specified reference. Argument is of the form "REF<NR1> ", where NR1 ≥ 1.

Group Vertical

Syntax `REF:ADDNew <QString>`

Arguments <QString> is the specified reference. Argument is of the form "REF<NR1> ", where NR1 ≥ 1.

Examples `REF:ADDNEW` "REF2" adds reference 2 to the display.

REF:DELete (No Query Form)

Deletes the specified reference. Argument is of the form "REF<NR1>", where NR1 ≥ 1.

Conditions Vertical

Group Vertical

Syntax `REF:DELete <QString>`

Arguments <QString> is the specified reference. Argument is of the form "REF<NR1>", where NR1 ≥ 1.

Examples `REF:DELETE` “REF2” deletes reference 2 from the display.

REF:LIST? (Query Only)

This command returns a comma separated list of all currently defined references.

Group Vertical

Syntax `REF:LIST?`

Returns All currently defined references.

Examples `REF:LIST?` might return `:REF:LIST REF1,REF2` indicating references 1 and 2 are defined.

REF:REF<x>:DESKew

This command sets or queries the deskew value used for the specified reference.

Group Vertical

Syntax `REF:REF<x>:DESKew <NR3>`

Arguments `<NR3>` is the deskew value used for the specified reference.

Examples `REF:REF1:DESKew -1.5e-9` sets the deskew value to -1.5 ns.

`REF:REF1:DESKew?` might return `:REF:REF1:DESKEW 1.5200E-9` indicating the deskew value is 1.52 ns.

REF:REF<x>:LABel:COLor

This command sets or queries the color of the specified ref label.

Group Vertical

Syntax `REF:REF<x>:LABel:COLor <QString>`

Arguments `<QString>` is the label. To return the color to the default color, send an empty string as in this example: `:REF:REF1:LABEL:COLOR ""`.

Examples `REF:REF1:LABel:COLor "#FFFF00"` sets the font color to yellow.

`REF:REF1:LABel:COLor?` might return `:REF:REF1:LABEL:COLOR "#FF0000"` indicating the font color is red.

REF:REF<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the specified reference label.

Group Vertical

Syntax `REF:REF<x>:LABel:FONT:BOLD {<NR1>|OFF|ON}`

Arguments `<NR1>` = 0 disables bold font; any other value turns this feature on.

`OFF` disables bold font.

`ON` enables bold font.

Examples `REF:REF1:LABel:FONT:BOLD ON` turns on the bold font.

`REF:REF1:LABel:FONT:BOLD?` might return `:REF:REF1:LABEL:FONT:BOLD 0` indicating the bold font is off.

REF:REF<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the specified reference label.

Group Vertical

Syntax `REF:REF<x>:LABel:FONT:ITALic {<NR1>|OFF|ON}`

Arguments `<NR1>` = 0 disables italic font; any other value turns this feature on.

`OFF` disables italic font.

`ON` enables italic font.

Examples `REF:REF1:LABel:FONT:ITALic ON` turns on the italic font.

`REF:REF1:LABel:FONT:ITALic?` might return `:REF:REF1:LABEL:FONT:ITALIC 0` indicating the italic font is off.

REF:REF<x>:LABel:FONT:SIZE

This command sets or queries the font size of the specified reference label.

Group Vertical

Syntax `REF:REF<x>:LABel:FONT:SIZE <NR1>`

Arguments `<NR1>` is the font size of the label.

Examples `REF:REF1:LABel:FONT:SIZE 20` sets the font size to 20 points.

`REF:REF1:LABel:FONT:SIZE?` might return `:REF:REF1:LABEL:FONT:SIZE 14` indicating that the font size is 14 points.

REF:REF<x>:LABel:FONT:TYPE

This command sets or queries the font type of the specified reference label, such as Arial or Times New Roman.

Group Vertical

Syntax `REF:REF<x>:LABel:FONT:TYPE <QString>`

Arguments `<QString>` is the font type.

Examples `REF:REF1:LABel:FONT:TYPE "Monospace"` specifies a mono spaced font.

`REF:REF1:LABel:FONT:TYPE?` might return `:REF:REF1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman"`.

REF:REF<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified reference label.

Group Vertical

Syntax `REF:REF<x>:LABel:FONT:UNDERline {<NR1>|OFF|ON}`

Arguments `<NR1>` = 0 disables underline font; any other value turns this feature on.

`OFF` disables underline font.

ON enables underline font.

Examples REF:REF1:LABel:FONT:UNDERline ON turns on the underline font.

REF:REF1:LABel:FONT:UNDERline? might return :REF:REF1:LABEL:FONT:UNDERLINE 0 indicating that underline is off.

REF:REF<x>:LABel:NAMe

This command sets or queries the label of the specified reference. The reference waveform is specified by x.

Group Vertical

Syntax
```
REF:REF<x>:LABel:NAMe <QString>
REF:REF<x>:LABel:NAMe?
```

Arguments <QString> is the character string that will be used for the reference waveform label name.

Examples REF:REF4:LABEL:NAME "My Reference" sets the label name of Reference 4 waveform to "My Reference".

REF:REF3:LABEL:NAME? might return :REF:REF3:LABEL:NAME "Signal2", indicating that the label name for Reference 3 waveform is currently set to "Signal2".

REF:REF<x>:LABel:XPOS

This command sets or queries the X-position at which the label (attached to the displayed waveform of the specified reference) is displayed, relative to the left edge of the waveview. The reference waveform is specified by x.

Group Vertical

Syntax
```
REF:REF<x>:LABel:XPOS <NR1>
REF:REF<x>:LABel:XPOS?
```

Arguments <NR1> is the location (control in divisions) where the waveform label for the selected reference is displayed, relative to the left edge of the screen.

Examples `REF:REF4:LABEL:XPOS 10` moves the waveform label for the Reference 3 waveform, so that it begins 10 divisions to the right of the left edge of the screen.

`REF:REF2:LABEL:XPOS?` might return `:REF:REF2:LABEL:XPOS 1.5`, indicating that the x-axis for the Reference 2 waveform is currently 1.5 divisions to the right of the left edge of the screen.

REF:REF<x>:LABel:YPOS

This command sets or queries the Y-position of the label (attached to the displayed waveform of the specified reference), relative to the baseline of the waveform. The reference waveform is specified by x.

Group Vertical

Syntax
```
REF:REF<x>:LABel:YPOS <NR1>
REF:REF<x>:LABel:YPOS?
```

Arguments `<NR1>` is the location where the waveform label for the selected reference is displayed, relative to the baseline of the waveform.

Examples `REF:REF3:LABEL:YPOS -10` moves the waveform label for the Reference 3 waveform 10 vertical units below the baseline of the waveform.

`REF:REF2:LABEL:YPOS?` might return `:REF:REF2:LABEL:YPOS 0`, indicating that the waveform label for the Reference 2 waveform is currently located at the baseline of the waveform.

REF:REF<x>:SOUrce

This command sets or queries the filename used by the given reference.

Group Vertical

Syntax
```
REF:REF<x>:SOUrce <QString>
```

Arguments `<QString>` is the reference file name.

Examples
```
REF:REF1:SOUrce
"/home/guest/.local/share/Tektronix/TekScope/
```

FirstRecalledSession/161012_132000_000.wfm" sets the source of the reference.

REF:REF1:SOUrce? might return :REF:REF1:SOURCE "/home/guest/.local/share/Tektronix/TekScope/LastRecalledSession/161012_132039_000.wfm".

REF<x>_DALL:LABel:COLor

This command sets or queries the color of the specified digital group. The reference is specified by x.

Group Digital

Syntax REF<x>_DALL:LABel:COLor <QString>

Arguments <QString> is the color of the digital group label. To return the color to the default color, send an empty string as in this example: :REF5_DALL:LABEL:COLOR "".

Examples REF1_DALL:LABel:COLor "#FF0000" sets the font color to red.

REF1_DALL:LABel:COLor? might return :REF1_DALL:LABEL:COLOR "#FFFF00" indicating the font color is yellow.

REF<x>_DALL:LABel:FONT:BOLD

This command sets or queries the bold state of the specified digital group. The reference is specified by x.

Group Digital

Syntax REF<x>_DALL:LABel:FONT:BOLD {ON|OFF|<NR1>}

Arguments OFF argument turns off bold font.

ON argument turns on bold font.

<NR1> = 0 turns off bold font; any other value turns on bold font.

Examples `REF1_DALL:LABel:FONT:BOLD ON` sets the font to bold.

`REF1_DALL:LABel:FONT:BOLD?` might return `:REF1_DALL:LABEL:FONT:BOLD 0` indicating the font is not bold.

REF<x>_DALL:LABel:FONT:ITALic

This command sets or queries the italic state of the specified digital group. The reference is specified by x.

Group Digital

Syntax `REF<X>_DALL:LABel:FONT:ITALic {ON|OFF|<NR1>}`

Arguments `OFF` argument turns off italic font.

`ON` argument turns on italic font.

`<NR1>` = 0 turns off italic font; any other value turns on italic font.

Examples `REF1_DALL:LABel:FONT:ITALic 1` turns on italic font.

`REF1_DALL:LABel:FONT:ITALic?` might return `:REF1_DALL:LABEL:FONT:ITALIC 0` indicating the font is not italic.

REF<x>_DALL:LABel:FONT:SIZE

This command sets or queries the font size of the specified digital group. The reference is specified by x.

Group Digital

Syntax `REF<X>_DALL:LABel:FONT:SIZE <NR1>`

Arguments `<NR1>` is the font size.

Examples `REF1_DALL:LABel:FONT:SIZE 16` sets the font size to 16 points.

`REF1_DALL:LABel:FONT:SIZE?` might return `:REF1_DALL:LABEL:FONT:SIZE 20` indicating the font size is 20 points.

REF<x>_DALL:LABel:FONT:TYPE

This command sets or queries the font type of the specified digital group, such as Arial or Times New Roman. The reference is specified by x.

Group Digital

Syntax `REF<x>_DALL:LABel:FONT:TYPE <QString>`

Arguments `<QString>` is the font type.

Examples `REF1_DALL:LABel:FONT:TYPE "Monospace"` sets the font to a monospace font.

`REF1_DALL:LABel:FONT:TYPE?` might return `:REF1_DALL:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman"`.

REF<x>_DALL:LABel:FONT:UNDERline

This command sets or queries the underline state of the specified digital group. The reference is specified by x.

Group Digital

Syntax `REF<x>_DALL:LABel:FONT:UNDERline {ON|OFF|<NR1>}`

Arguments `OFF` argument turns off underline font.

`ON` argument turns on underline font.

`<NR1>` = 0 turns off underline font; any other value turns on underline font.

Examples `REF1_DALL:LABel:FONT:UNDERline ON` specifies an underlined font.

`REF1_DALL:LABel:FONT:UNDERline?` might return `:REF1_DALL:LABEL:FONT:UNDERLINE 0` indicating underline is off.

REF<x>_DALL:LABel:NAMe

This command sets or queries the label of the specified digital group. The reference is specified by x.

Group Digital

Syntax `REF<x>_DALL:LABel:NAMe <QString>`

Arguments `<QString>` is the name of the group.

Examples `REF1_DALL:LABel:NAMe “Clock Out”` sets the label name to Clock Out.

`REF1_DALL:LABel:NAMe?` might return `:REF1_DALL:LABEL:NAME "This is the digital name"`.

REF<x>_DALL:LABel:XPOS

This command sets or queries the x-position of the label of the specified digital group. The reference is specified by x.

Group Digital

Syntax `REF<x>_DALL:LABel:XPOS <NR3>`

Arguments `<NR3>` is the x-position, in pixels relative to the left edge of the display, of the group.

Examples `REF1_DALL:LABel:XPOS 90.0e0` sets the x position of the label to 90.

`REF1_DALL:LABel:XPOS?` might return `:REF1_DALL:LABEL:XPOS 45.0000` indicating the x position of the label is at 45 pixels to the right of the left edge of the display.

REF<x>_DALL:LABel:YPOS

This command sets or queries the y-position of the label of the specified digital group. The reference is specified by x.

Group Digital

Syntax `REF<x>_DALL:LABel:YPOS <NR3>`

Arguments `<NR3>` is the y-position, in pixels relative to the baseline of the waveform, of the group.

Examples `REF1_DALL:LABel:YPOS 50` sets the y position to 50.

`REF1_DALL:LABel:YPOS?` might return `:REF1_DALL:LABEL:YPOS 0.0E+0` indicating the y position of the label is at the baseline of the waveform.

REF<x>_D<x>:LABel:COLor

This command sets or queries the color of the label of the specified digital bit. The reference is specified by x.

Group Digital

Syntax `REF<x>_D<x>:LABel:COLor <QString>`

Arguments `<QString>` is the label color. To return the color to the default color, send an empty string as in this example: `:REF5_D1:LABEL:COLOR ""`.

Examples `REF1_D1:LABel:COLor "#FF0000"` sets the color to red.

`REF1_D1:LABel:COLor?` might return `:REF1_D1:LABEL:COLOR "#FFFF00"` indicating the color is yellow.

REF<x>_D<x>:LABel:FONT:BOLD

This command sets or queries the bold state of the label of the specified digital bit. The reference is specified by x.

Group Digital

Syntax `REF<x>_D<x>:LABel:FONT:BOLD {ON|OFF|<NR1>}`

Arguments OFF argument turns off bold font.

ON argument turns on bold font.

<NR1> = 0 turns off bold font; any other value turns on bold font.

Examples REF1_D1:LABel:FONT:BOLD ON sets the font to bold.

REF1_D1:LABel:FONT:BOLD? might return :REF1_D1:LABEL:FONT:BOLD 0 indicating the font is not bold.

REF<x>_D<x>:LABel:FONT:ITALic

This command sets or queries the italic state of the label of the specified digital bit. The reference is specified by x.

Group Digital

Syntax REF<X>_D<X>:LABel:FONT:ITALic {ON|OFF|<NR1>}

Arguments OFF argument turns off italic font.

ON argument turns on italic font.

<NR1> = 0 turns off italic font; any other value turns on italic font.

Examples REF1_D1:LABel:FONT:ITALic OFF turns off italic font.

REF1_D1:LABel:FONT:ITALic? might return :REF1_D1:LABEL:FONT:ITALIC 1 indicating the font is italic.

REF<x>_D<x>:LABel:FONT:SIZE

This command sets or queries the font size of the label of the specified digital bit. The reference is specified by x.

Group Digital

Syntax REF<X>_D<X>:LABel:FONT:SIZE <NR1>

Arguments <NR1> is the font size.

Examples `REF1_D1:LABel:FONT:SIZE 16` sets the font size to 16 points.

`REF1_D1:LABel:FONT:SIZE?` might return `:REF1_D1:LABEL:FONT:SIZE 20` indicating the font size is 20 points.

REF<x>_D<x>:LABel:FONT:TYPE

This command sets or queries the font type of the label of the specified digital bit, such as Arial or Times New Roman. The reference is specified by x.

Group Digital

Syntax `REF<x>_D<x>:LABel:FONT:TYPE <QString>`

Arguments `<QString>` is the font type of the label.

Examples `REF1_D1:LABel:FONT:TYPE "Monospace"` sets the font to Monospace.

`CHREF1_D1:LABel:FONT:TYPE?` might return `:REF1_D1:LABEL:FONT:TYPE "Frutiger LT Std 55 Roman"`.

REF<x>_D<x>:LABel:FONT:UNDERline

This command sets or queries the underline state of the label of the specified digital bit. The reference is specified by x.

Group Digital

Syntax `REF<x>_D<x>:LABel:FONT:UNDERline {ON|OFF|<NR1>}`

Arguments `OFF` argument turns off underline font.

`ON` argument turns on underline font.

`<NR1>` = 0 turns off underline font; any other value turns on underline font.

Examples `REF1_D1:LABel:FONT:UNDERline ON` turns on underline font.

`REF1_D1:LABel:FONT:UNDERline?` might return `:REF1_D1:LABEL:FONT:UNDERLINE 0` indicating the underline font is off.

REF<x>_D<x>:LABel:NAMe

Sets or queries the label of the specified digital bit. The channel is specified by x.

Group Digital

Syntax `REF<x>_D<x>:LABel:NAMe <QString>`

Arguments `<QString>` is the label.

Examples `REF1_D1:LABel:NAMe "Clock in"` sets the name to Clock in.

`REF1_D1:LABel:NAMe?` might return `:REF1_D1:LABEL:NAME "Digital 1"`.

REF<x>_D<x>:LABel:XPOS

This command sets or queries the x-position of the label of the specified digital bit. The reference is specified by x.

Group Digital

Syntax `REF<x>_D<x>:LABel:XPOS <NR3>`

Arguments `<NR3>` is the x-position, in pixels relative to the left edge of the display, of the label.

Examples `REF1_D1:LABel:XPOS 90` sets the x position to 90.

`REF1_D1:LABel:XPOS?` might return `:REF1_D1:LABEL:XPOS 45.0000` indicating the position is 45 pixels to the right of the left edge of the waveform.

REF<x>_D<x>:LABel:YPOS

This command sets or queries the y-position of the label of the specified digital bit. The channel is specified by x.

Group Digital

Syntax `REF<x>_D<x>:LABel:YPOS <NR3>`

Arguments <NR3> is the y-position, in pixels relative to the baseline of the waveform, of the label.

Examples `REF1_D1:LABel:YPOS 10` sets the y position to 10.

`REF1_D1:LABel:YPOS?` might return `:REF1_D1:LABEL:YPOS 0.0E+0` indicating the y position of the label is at the baseline of the waveform.

REM (No Query Form)

This command (no query form) embeds a comment within programs as a means of internally documenting the programs. This is how to embed comments in a .set file. The instrument ignores these embedded comment lines.

Group Miscellaneous

Syntax `REM <QString>`

Arguments <QString> is a string that can contain a maximum of 80 characters.

Examples `REM "This is a comment"` is a comment string that the instrument will ignore.

ROSc:SOUrce

This command sets or queries the selected source for the time base reference oscillator. The reference oscillator locks to this source. Depending on the command argument that you specify, you can use an external reference or use the internal crystal oscillator as the time base reference.

Group Miscellaneous

Syntax
```
ROSc:SOUrce {INTERnal|EXTernal}
ROSc:SOUrce?
```

Related Commands ROSc:STATE?

Arguments

`INTERnal` specifies the internal 10 MHz crystal oscillator as the time base reference.

`EXTernal` specifies the user-supplied external signal as the time base reference.

Examples

`ROSC:SOURCE INTERNAL` specifies the internal 10 MHz crystal oscillator as the time base reference.

`ROSC:SOURCE?` might return `:ROSC:SOURCE INTERNAL`, indicating that the 10 MHz crystal oscillator is being used as the time base reference.

ROSc:STATE? (Query Only)

This query-only command returns whether the time base reference oscillator is locked. This command will return either LOCKED or UNLOCKED.

Group

Miscellaneous

Syntax

`ROSc:STATE?`

Returns

`LOCKED` indicates the reference oscillator is locked.

`UNLOCKED` indicates the reference oscillator is not locked.

Examples

`ROSC:STATE?` might return `:ROSC:STATE LOCKED`, indicating that the time base reference is locked.

*RST (No Query Form)

This command (no query form) resets the instrument to the factory default settings.

This command does the following:

- Recalls the default instrument setup.
- Clears the current *DDT command.
- Disables aliases (:ALIAS:STATE 0).
- Disables the user password (for the *PUD command).

The *RST command does not change the following:

- The current working directory (`:FILESystem:CWD` command).
- The state of command headers (`:HEADer` command).
- The state of keyword and enumeration verbosity (`:VERBose` command).
- The Power-on Status Clear Flag (`*PSC` command).
- The Event Status Enable Register (`*ESE` command).
- The Service Request Enable Register (`*SRE` command).
- The Device Event Status Enable Register (`DESE` command).
- The user password for protected user data (`:PASSWord` command).
- The content of protected user data (`*PUD` command).
- The enabled state of the socket server (`:SOCKETServer:ENAble` command).
- The socket server port number (`:SOCKETServer:PORT` command).
- The socket server protocol (`:SOCKETServer:PROTOCol` command).
- The USBTMC port configuration (`:USBDevice:CONFigure` command).
- The destination reference waveform or file path for the `:CURVe` command (`:DATa:DESTination` command).
- The source waveform for the `:CURVe?` or `:WAVFrm?` queries (`:DATa:SOUrce` command).
- The waveform data encoding for the `:CURVe` command or query or the `:WAVFrm?` query (`:DATa:ENCdg` command).
- The starting point for `:CURVe?` queries (`:DATa:STARt` command).
- The ending point for `:CURVe?` queries (`:DATa:STOP` command).
- All settings associated the `:WFMInpre` commands.
- All user settable settings associated with the `WFMOutpre` commands.

*RST only resets the programmable interface settings, it does not change the user interface settings.

Group Status and Error

Syntax `*RST`

Related Commands FACtory

RECAll:SETUp

Arguments None

Examples `*RST` resets the instrument settings to factory defaults.

SAVe:EVENTtable:BUS (No Query Form)

This command saves bus results table to the specified file.

Group Save and Recall

Syntax `SAVe:EVENTtable:BUS <QString>`

Arguments <QString> is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified.

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `SAVe:EVENTtable:BUS "TEK000.CSV"` saves the bus decode event table in the file named TEK000.CSV.

SAVe:EVENTtable:CUSTom (No Query Form)

This command saves the results table to the specified file path and name.

Group Save and Recall

Syntax `SAVe:EVENTtable:CUSTom <QString>`

Arguments <QString> is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified.

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `SAVe:EVENTtable:CUSTom "C:/TEK000.CSV"` sets the instrument to save the results table in the top level C: directory in a file named TEK000.CSV.

SAVe:EVENTtable:CUSTom:COMMents

This command sets or queries comments to be included in saved results table files.

Group Save and Recall

Syntax
```
SAVe:EVENTtable:CUSTom:COMMents <Qstring>
SAVe:EVENTtable:CUSTom:COMMents?
```

Arguments `<Qstring>` sets the instrument to save the quoted string as a comment in the saved results table file.

Examples `SAVe:EVENTtable:CUSTom:COMMents ”PSRR batch 2 testing.”` sets the instrument to save the quoted string in all following saved results table files.

`SAVe:EVENTtable:CUSTom:COMMents?` might return "", indicating that the there is no comment set to be saved in results table files.

SAVe:EVENTtable:CUSTom:DATAFormat

This command sets or queries the data format to use for saving results table data.

Group Save and Recall

Syntax `SAVe:EVENTtable:CUSTom:DATAFormat [SCIentific|ENGineering]`

Arguments `SCIentific` sets the instrument to save results tables data in scientific notation (for example, 5.0100E-12).

ENGineering sets the instrument to save results tables data in engineering notation (for example, 5.0100ps).

Examples

SAVe:EVENTtable:CUSTom:DATAFormat SCIentific sets the instrument to save results tables data in scientific notation.

SAVe:EVENTtable:CUSTom:DATAFormat SCIentific might return ENGINEERING, indicating that the instrument is set to save results tables data in engineering notation.

SAVe:EVENTtable:CUSTom:INCLUDEREFs

This command sets or queries whether to include displayed reference waveforms with saved results table files.

Group

Save and Recall

Syntax

```
SAVe:EVENTtable:CUSTom:INCLUDEREFs {1|0}
SAVe:EVENTtable:CUSTom:INCLUDEREFs?
```

Arguments

1 sets the instrument to save all displayed reference waveforms as part of a saved results table file.

0 sets the instrument to not save all displayed reference waveforms as part of a saved results table file.

Examples

SAVe:EVENTtable:CUSTom:INCLUDEREFs 1 sets the instrument to save all displayed reference waveforms as part of a saved results table file.

SAVe:EVENTtable:CUSTom:INCLUDEREFs? might return 0, indicating that the instrument will not save all displayed reference waveforms as part of a saved results table file.

SAVe:EVENTtable:MEASUrement (No Query Form)

This command saves data (measurement) results to the specified file.

Group

Save and Recall

Syntax

```
SAVe:EVENTtable:MEASUrement <QString>
```

Arguments <QString> is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `SAVe:EVENTtable:MEASUrement "TEK000.CSV"` saves the measurement in the file named TEK000.CSV.

SAVe:EVENTtable:SEARCHTable (No Query Form)

This command saves a search results table to the specified file.

Group Save and Recall

Syntax `SAVe:EVENTtable:SEARCHTable <QString>`

Arguments <QString> is the specified file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `SAVe:EVENTtable:SEARCHTable "C:\Users\Tek_Local_Admin\Tektronix\TekScope\Results Tables\Tek002.csv"` saves the search results table in the file named TEK002.CSV.

SAVe:IMAGe (No Query Form)

Saves a capture of the screen contents to the specified image file. Supported image formats are PNG, Windows Bitmap, and JPEG.

Group Save and Recall

Syntax

```
SAVe:IMAGe <QString>
```

Arguments <QString> is the file name and location used to store the image file.

When specifying the file name with this command, use the correct file extension (".png" for PNG format, ".bmp" for BMP format, or ".jpg" for JPEG format). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by :FILESystem:CWDFILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples SAVE:IMAGE "C:/Dut12–tests.png" saves the image at the location specified.

SAVe:IMAGe:COMPosition

Sets or queries the color mode for saved images (normal or inverted).

Group Save and Recall

Syntax

```
SAVe:IMAGe:COMPosition {NORMal|INVErted}
SAVe:IMAGe:COMPosition?
```

Arguments NORMal Sets the saved screen capture to Normal colors.

INVErted sets the saved screen capture to Inverted colors.

Examples SAVE:IMAGE:COMPOSITION NORMal saves the screen image using the current normal color settings.

SAVE:IMAGE:COMPOSITION? might return INVERTED, indicating that captured screen images will be saved using the inverted color settings.

SAVe:IMAGe:VIEWTYpe

Sets or queries the view type for saved images. Currently only FULLScreen is supported.

Group Save and Recall

Syntax
```
SAVe:IMAGe:VIEWTYpe {FULLScreen}
SAVe:IMAGe:VIEWTYpe?
```

Arguments FULLScreen sets the screen capture mode to capture the full screen.

Examples SAVE:IMAGE:VIEWTYPE FULLScreen sets the screen capture mode to capture the full screen.

SAVE:IMAGE:VIEWTYPE? might return FULLScreen, indicating that the screen capture mode is set to full screen.

SAVe:PLOTData (No Query Form)

Saves the plot data of the currently selected plot to a specified file. Supported file format is CSV.

Group Save and Recall

Syntax
```
SAVe:PLOTData <Qstring>
```

Related Commands DISplay:SELect:VIEW

FILESystem:CWD

Arguments Qstring sets the file name and location used to store the plot data. When specifying the file name with this command, use the correct file extension (.CSV). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified.

If the file argument begins with a drive designator (such as C:), then the file name is interpreted as a full path. If the file argument begins with "." or "", or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

To export an eye diagram plot data to a .csv file, the prerequisite command is `MEASUrement:ADDMEAS TIE`

Examples

`SAVE:PLOTDATA "plot1.csv"` saves the plot to the designated file in the current working directory.

The following is the example of exporting an XY plot data to a .csv file:

```
PLOT:PLOT1:TYPe XY
DISplay:SELect:VIEW PLOTVIEW1
SAVe:PLOTData "C:/plot1.csv" (On standard instruments)
SAVe:PLOTData "C:\Temp\plot1.csv" (On instruments with optional Windows 10)
```

SAVe:REPOrt (No Query Form)

This command saves a report to the specified file. Supported report formats are PDF and MHT (web page archive file).

Group

Save and Recall

Syntax

`SAVe:REPOrt <QString>`

Related Commands

SAVe:REPOrt:COMMents

Arguments

`<QString>` is the complete path specification. When specifying the file name with this command, use the correct file extension (.pdf for PDF format, or .mht for MHT format).

If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples SAVE:REPORT "report.pdf" creates a report in PDF format, in the location specified.

SAVe:REPOrt:COMMents

This command sets or queries the comments to be included in saved report files.

Group Save and Recall

Syntax
```
SAVe:REPOrt:COMMents <QString>
SAVe:REPOrt:COMMents?
```

Arguments <QString> is the comments to be included in saved report files.

Examples SAVE:REPORT:COMMENTS "Test 3" adds comments to the report.

SAVE:REPORT:COMMENTS? might return :SAVE:REPORT:COMMENTS "High Temp Test 1".

SAVe:SESsion (No Query Form)

Saves the state of the instrument, including reference waveforms, to a saved session file.

Group Save and Recall

Syntax SAVe:SESsion <QString>

Arguments <QString> is the file path that specifies the location to save the specified instrument session file. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWDFILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `SAVE:SESSION` “c:/rose_was_here.tss” saves the instrument state in the specified file.

SAVe:SETUp (No Query Form)

Saves the current instrument state to the specified file.

Group Save and Recall

Syntax

```
SAVe:SETUp <QString>
```

Related Commands SAVe:SETUp:INCLUDEREFs

Arguments `<QString>` is a quoted string that is the complete path specification. If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a Windows drive designator such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples `SAVE:SETUP “c:/rose_was_here.set”` saves the instrument setup in the specified file.

SAVe:SETUp:INCLUDEREFs

This command sets or queries whether displayed reference waveforms are to be included in saved setups.

Group Save and Recall

Syntax

```
SAVe:SETUp:INCLUDEREFs {OFF|ON|0|1}
SAVe:SETUp:INCLUDEREFs?
```

Arguments

OFF specifies not including displayed reference waveforms in saved setups.

ON specifies including displayed reference waveforms in saved setups.

0 specifies not including displayed reference waveforms in saved setups.

1 specifies including displayed reference waveforms in saved setups.

Examples

SAVE:SETUP:INCLUDEREFS 0 sets reference waveforms not to be included in saved setups.

SAVE:SETUP:INCLUDEREFS? might return :SAVE:SETUP:INCLUDEREFS 1 indicating that reference waveforms are to be included in saved setups.

SAVe:WAVEform (No Query Form)

This command saves the specified waveform(s) to the specified destination file. The waveform source or sources must be active (turned on) to save data to a file.

Conditions

_MAG_VS_TIME, _FREQ_VS_TIME, _PHASE_VS_TIME, and _SV_BASEBAND_IQ require option SV-RFVT.

Option SV-RFVT on the 4 Series MSO instruments requires option SV-BAS.

Group

Save and Recall

Syntax

```
SAVe:WAVEform
{CH<x>[_DALL|_SV_NORMal|_SV_AVErage|_SV_MAXHold|
_SV_MINHold|_MAG_VS_TIME|_FREQ_VS_TIME|
_PHASE_VS_TIME]|MATH<x>|REF<x>|ALL| },<QString>
```

Related Commands

FILESystem:CWD

SAVe:WAVEform:SOURCELIst?

Arguments

<x> is the number of the analog channel, math, or reference waveform source used to save the waveform data.

_DALL saves the digital channel waveform data of the specified channel. This argument is required if the channel specified is a digital channel.

_SV_NORMal saves the Normal Spectrum view waveform of the specified channel.

_SV_AVErage saves the Average Spectrum view waveform of the specified channel.

_SV_MAXHold saves the Maximum Hold Spectrum view waveform of the specified channel.

_SV_MINHold saves the Minimum Hold Spectrum view waveform of the specified channel.

_MAG_VS_TIME saves the Magnitude vs. Time waveform of the specified channel.

_FREQ_VS_TIME saves the Freuency vs. Time waveform of the specified channel.

_PHASE_VS_TIME saves the Phase vs. Time waveform of the specified channel.

_SV_BASEBAND_IQ saves the baseband I & Q data of the specified channel. The data is saved in Tektronix TIQ format using a .TIQ file extension. You can import .TIQ files into Tektronix SignalVu-PC software (PC based), SignalVu software (oscilloscope based), RSAVu software (PC based) or into a Tektronix real-time spectrum analyzer for pulse analysis and demodulation analysis.

ALL saves all displayed analog, math, and reference waveforms to individual files. Each file name created includes the name of the source (ch1, math3, and so on) used to create that file.

<Qstring> is a quoted string that defines the path and file name to use to save the specified file, in the format '[<path>]<filename.ext>'. Specifying a path is optional. If no path is entered, the file is saved to the current working directory as set in FILESystem:CWD.

<path> uses the form '<drive>/<dir>.../'. You can specify a relative path or a complete path:

- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.
- Complete path specification. If the file argument begins with a file path separator (forward slash character) or a drive designator (such as C:), then the file name is interpreted as a full path from the specified drive.

<filename> sets the file name to use to create the file. A file can have up to 125 characters. When using the ALL agument to save multiple files, each filename has the filename appended with the source used to create that file. For example, a filename of QualTest can create QualTest_ch1.xxx, QualTest_ref1.xxx, and so on.

<.ext> sets the file format to which to save the data:

- Use the .wfm extension to save waveform data to a Tektronix Internal format.
- Use the .csv extension to save waveform data to a comma separated values spreadsheet format.
- Use the .mat extension to save waveform data to a matlab compatible file format.

Examples SAVE:WAVEFORM MATH1,"TEK0000.WFM" saves the Math1 waveform to the file TEK00000.WFM in the current working directory.

SAVe:WAVEform:GATing

This command specifies the method to save a specified part of the waveform data or the entire waveform.

Group Save and Recall

Syntax
```
SAVe:WAVEform:GATing {NONe|CURSors|SCREEN|RESAMPLE}
SAVe:WAVEform:GATing?
```

Arguments

NONe saves the full waveform data.

CURSors saves the waveform data located between the vertical cursors.

SCREEN saves the waveform data that is on the screen. Nothing outside the waveform will be saved.

RESAMPLE saves the waveform data at a sample interval set by the user. The resulting saved waveform is a resampled version of the original waveform with fewer data points.

Examples SAVE:WAVEFORM:GATING NONe saves the entire waveform.

SAVE:WAVEFORM:GATING? might return :SAVe:WAVEform:GATing SCREEN, indicating the save waveform operation is set to save the waveform data on the screen.

SAVe:WAVEform:GATing:RESAMPLErate

This command saves the waveform data at a sample interval. The resulting saved waveform is a resampled version of the original waveform with fewer data points.

Group Save and Recall

Syntax
```
SAVe:WAVEform:GATing:RESAMPLErate <NR1>
SAVe:WAVEform:GATing:RESAMPLErate?
```

Arguments <NR1> specifies the resample interval.

Examples

`SAVE:WAVEFORM:GATING:RESAMPLERATE 3` sets to save waveform data every third data point.

`SAVE:WAVEFORM:GATING:RESAMPLERATE?` might return `:SAVe:WAVEform:GATing:RESAMPLErate 2`, indicating the saved waveform will have half as many data points. This command saves the first data point and then every other following data point.

SAVe:WAVEform:SOURCELIst? (Query Only)

This query returns a list of the available waveforms that can be specified as the source for the SAVe:WAVEform command. Source waveforms must have their display mode set to On to appear in this list and to be saved.

Group Save and Recall

Syntax

```
SAVe:WAVEform:SOURCELIst?
```

Examples

`SAVE:WAVEFORM:SOURCELIST?` might return `ALL,CH1_DALL,CH2,CH3,CH8,MATH1,MATH2,REF2,REF4`.

SAVEON:FILE:DEST

This command sets or queries the location where files are saved when SAVEON:TRIGGER is ON and SAVEON:WAVEFORM is ON. You can save the files to a local drive or network path by entering the desired location in <QString>. You can also select to save the files to a USB drive.

Group Save On

Syntax

```
SAVEON:FILE:DEST <QString>
SAVEON:FILE:DEST?
```

Related Commands SAVEON:FILE:NAME

Arguments <QString> specifies the location to store files.

Examples

`SAVEON:FILE:DEST "C:\users\username\Tektronix\TekScope\SaveOnTrigger"` sets this

as the location to save files (named by the SAVEON:FILE:NAME command), when there is a trigger.

SAVEON:FILE:DEST? might return :SAVEON:FILE:DEST "C:\users\username\Tektronix\TekScope\SaveOnEvent", indicating the drive location where files will be saved when there is a trigger.

SAVEON:FILE:NAME

Sets or queries the file name to use when SAVEON:TRIGer is ON.

Group Save On

Syntax

```
SAVEON:FILE:NAME <QString>
SAVEON:FILE:NAME?
```

Related Commands SAVEON:FILE:DEST

Arguments <QString> is the file name you want to use.

Examples SAVEON:FILE:NAME "MaskFailure" sets the name of the file to MaskFailure.

SAVEON:FILE:NAME? might return ":SAVEON:FILE:NAME MaskFailure5", indicating the name you set for the instrument to use, with the autoincrement number (5) appended.

SAVEON:IMAGe

This command sets or queries whether to save a screen capture when a trigger occurs and SAVEON:TRIGer is ON and SAVEON:IMAGE is ON.

Group Save On

Syntax

```
SAVEON:IMAGe {<NR1>|OFF|ON}
SAVEON:IMAGe?
```

Related Commands SAVEON:FILE:DEST

SAVEON:FILE:NAME

SAVEON:TRIGger

SAVEON:IMAGe

Arguments

<NR1> = 0 disables Save On Image; any other value turns this feature on.

OFF disables Save On Image.

ON enables Save On Image.

Examples

SAVEON:IMAGE ON sets the instrument to save a screen capture on a specified trigger.

SAVEON:IMAGE? might return :SAVEON:IMAGE 1, indicating that the instrument will save a screen capture when the specified trigger occurs.

SAVEON:IMAGe:FILEFormat

This command sets or queries the file format to be used for saved image files when :SAVEON:IMAGe is set to 1.

Group

Save On

Syntax

```
SAVEON:IMAGe:FILEFormat {PNG|BMP|JPG}
SAVEON:IMAGe:FILEFormat?
```

Arguments

PNG specifies using PNG format for saved image files.

BMP specifies using BMP format for saved image files.

JPG specifies using JPEG format for saved image files.

When specifying the file name with this command, use the correct file extension (".png" for PNG format, ".bmp" for BMP format, or ".jpg" for JPEG format). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or ".." or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

`SAVEON:IMAGE:FILEFORMAT JPG` sets the image file format to JPEG.

`SAVEON:IMAGE:FILEFORMAT?` might return `:SAVEON:IMAGE:FILEFORMAT PNG` indicating that the file format is set to PNG.

SAVEON:TRIGger

Sets or queries whether to save a file when a trigger occurs. You can define the trigger using Trigger commands or the instrument user interface.

This command is longer necessary. Please see Act On Event commands for future development.

The trigger will cause the instrument to save an image or a waveform to a file, depending on what you specified. For example, if you have set SAVEON:IMAGe to On, and a trigger event occurs, the instrument will save a screen capture. You can set options for file storage (such as file name, file destination, and auto increment), using the SAVEON:FILE commands.

Use the instrument interface to select whether to save one or more analog channels, digital channels, or math waveforms

Analog and math waveforms are saved using one file per waveform. Digital waveforms are all saved to a single file.

Group

Save On

Syntax

```
SAVEON:TRIGger {<NR1>|ON|OFF}
SAVEON:TRIGger?
```

Related Commands

SAVEON:IMAGe

SAVEON:WAVEform

SAVEON:FILE:DEST

SAVEON:FILE:NAME

Arguments

`<NR1>` = 0 disables Save On Trigger; any other value turns this feature on.

`OFF` disables Save On Trigger.

`ON` enables Save On Trigger.

Examples

`SAVEON:TRIGGER ON` sets the instrument to save an image, measurement, and/or waveform when a trigger occurs.

SAVEON:TRIGGER? might return :SAVEON:TRIGGER ON, indicating that a file will be saved on triggering.

SAVEON:WAVEform

Sets or queries whether to save a waveform when a trigger occurs when SAVEON:TRIGger is ON.

The waveform will be saved to the file you selected with SAVEON:FILE:NAME, in the location that you selected using SAVEON:FILE:DEST. You can set options for file storage (such as file name, file destination, and autoincrement), using the SAVEON:FILE commands.

Group Save On

Syntax

```
SAVEON:WAVEform {<NR1>|ON|OFF}
SAVEON:WAVEform?
```

Related Commands SAVEON:FILE:DEST

SAVEON:FILE:NAME

SAVEON:TRIGger

Arguments <NR1> = 0 disables Save On Waveform; any other value turns this feature on.

OFF disables Save On Waveform.

ON enables Save On Waveform.

Examples SAVEON:WAVEFORM ON turns on the Save On Waveform feature, so that a waveform will be saved when a selected trigger occurs.

SAVEON:WAVEFORM? might return :SAVEON:WAVEFORM ON, indicating that a waveform will be saved when a selected trigger occurs.

SAVEON:WAVEform:FILEFormat

This command sets or queries the file format for saving waveforms when :SAVEON:WAVEform is set to 1.

Group Save On

Syntax

```
SAVEON:WAVEform:FILEFormat {INTERNal|SPREADSheet}
SAVEON:WAVEform:FILEFormat?
```

Arguments

`INTERNal` specifies saving the waveform in the instrument internal format.

`SPREADSheet` specifies saving the waveform in comma separated values format.

Examples

`SAVEON:WAVEFORM:FILEFORMAT SPREADSheet` sets the file format to spreadsheet.

`SAVEON:WAVEFORM:FILEFORMAT?` might return `:SAVEON:WAVEFORM:FILEFORMAT INTERNAL` indicating the file format is set to `INTERNAL`.

SAVEON:WAVEform:SOURce

This command sets or queries the sources for saving waveforms when SAVEON:TRIGger is ON.

Group

Save On

Syntax

```
SAVEON:WAVEform:SOURce {CH<x>|MATH<x>|REF<x>|ALL}
SAVEON:WAVEform:SOURce?
```

Arguments

Arguments are the available sources.

Examples

`SAVEON:WAVEform:SOURce MATH1` specifies MATH 1 as the save on source.

`SAVEON:WAVEform:SOURce?` might return `:SAVEON:WAVEFORM:SOURCE REF1` indicating the save on source is REF1.

SAVEONEVent:FILEDest

This command sets or queries the location where files are saved. This command replaces SAVEON:FILE:DEST (still valid command, but only an alias for this new command).

Group

Act On Event

Syntax

```
SAVEONEVent:FILEDest
SAVEONEVent:FILEDest?  <Qstring>
```

Arguments <QString> specifies the location to store files.

Examples SAVEONEVENT:FILEDEST "C:\users\username\Tektronix\TekScope\SaveOnTrigger" sets this as the location to save files (named by the SAVEONEVent:FILEName command), when there is a trigger.

SAVEONEVENT:FILEDEST? might return :SAVEONEVent:FILEDest "C:\users\username\Tektronix\TekScope\SaveOnTrigger", indicating the drive location where files will be saved when there is a trigger.

SAVEONEVent:FILEName

This command sets or queries the file name without the extension. This command replaces SAVEON:FILE:NAME (still valid command, but only an alias for this new command).

Group Act On Event

Syntax
```
SAVEONEVent:FILEName <QString>
SAVEONEVent:FILEName?
```

Arguments <QString> specifies the name of the file.

Examples SAVEONEVENT:FILENAME "MaskFailure" sets the name of the file to MaskFailure.

SAVEONEVENT:FILENAME? might return :SAVEONEVent:FILEName "MaskFailure5", indicating the name you set for the instrument to use. The autoincrement number of 5 is appended.

SAVEONEVent:IMAGe:FILEFormat

This command sets or returns the image file extension (png, jpg, bmp). This command replaces SAVEON:IMAGe:FILEFormat (still valid command, but only an alias for this new command).

Group Act On Event

Syntax
```
SAVEONEVent:IMAGe:FILEFormat {PNG|BMP|JPG}
SAVEONEVent:IMAGe:FILEFormat?
```

Arguments

`PNG` specifies using PNG format for saved image files.

`BMP` specifies using BMP format for saved image files.

`JPG` specifies using JPEG format for saved image files.

When specifying the file name with this command, use the correct file extension (".png" for PNG format, ".bmp" for BMP format, or ".jpg" for JPEG format). If a file name or path is specified, the file is expected to be located in a directory relative to the current working directory (specified by FILESystem:CWD) unless a complete path is specified:

- Complete path specification. If the file argument begins with drive designator (such as C:), then the file name is interpreted as a full path.
- Relative path specification. If the file argument begins with "." or "" or has a file path separator appearing anywhere other than the first character position, then the file name is treated as a path that is relative to the current working directory.

Examples

`SAVEONEVENT:IMAGE:FILEFORMAT JPG` sets the image file format to JPEG.

`SAVEONEVENT:IMAGE:FILEFORMAT?` might return `:SAVEON:IMAGE:FILEFORMAT PNG` indicating that the file format is set to PNG.

SAVEONEVent:WAVEform:FILEFormat

This command sets or returns the file extension (csv, wfm, mat). This command replaces SAVEON:WAVEform:FILEFormat (still valid command, but only an alias for this new command).

Group

Act On Event

Syntax

```
SAVEONEVent:WAVEform:FILEFormat
{INTERNal|SPREADSheet|MATlab}
SAVEONEVent:WAVEform:FILEFormat?
```

Arguments

`INTERNal` specifies saving the waveform in the instrument internal format.

`SPREADSheet` specifies saving the waveform in comma separated values format.

`MATlab` specifies saving the waveform in matlab compatible file format.

Examples

`SAVEONEVENT:WAVEFORM:FILEFORMAT SPREADSheet` sets the file format to spreadsheet (csv).

SAVEONEVENT:WAVEFORM:FILEFORMAT? might return :SAVEONEVent:WAVEform:FILEFormat INTERNAL, indicating the file format is set to INTERNAL.

SAVEONEVent:WAVEform:SOUrce

This command sets or returns the sources for saving waveforms when an event occurs. This command replaces SAVEON:WAVEform:SOURce (still valid command, but only an alias for this new command).

Group Act On Event

Syntax
```
SAVEONEVent:WAVEform:SOUrce {CH<x>|MATH<x>|REF<x>|ALL}
SAVEONEVent:WAVEform:SOUrce?
```

Arguments

CH<x> specifies an analog channel as the source waveform for saving.

Math<x> specifies a math waveform as the source waveform for saving.

REF<x> specifies a reference waveform as the source waveform for saving.

ALL specifies all analog, math, and reference waveforms as the source waveforms for saving.

Examples

SAVEONEVENT:WAVEFORM:SOURCE MATH1 specifies MATH 1 as the save on source.

SAVEONEVENT:WAVEFORM:SOURCE? might return :SAVEONEVent:IMAGe:FILEFormat REF1, indicating the save on source is REF 1.

SEARCH:ADDNew (No Query Form)

This command adds the specified search.

Group Search and Mark

Syntax
```
SEARCH:ADDNew <QString>
```

Arguments

<QString> is the specified search. The argument is of the form "SEARCH<NR1>", where <NR1> is ≥ 1.

Examples SEARCH:ADDNEW “SEARCH2” adds a new search named SEARCH 2.

SEARCH:DELete (No Query Form)

This command deletes the specified search.

Group Search and Mark

Syntax SEARCH:DELete <QString>

Arguments <QString> is the specified search. The argument is of the form "SEARCH<NR1>", where <NR1> is ≥ 1).

Examples SEARCH:DELETE “SEARCH3” deletes SEARCH 3.

SEARCH:DELETEALL (No Query Form)

This command deletes all the active instances of search definitions defined in the scope application.

Group Search and Mark

Syntax SEARCH:DELETEALL

Examples SEARCH:DELETEALL deletes all the active instances of search definitions.

SEARCH:LIST? (Query Only)

This command returns a comma separated list of all currently defined searches.

Group Search and Mark

Syntax SEARCH:LIST?

Returns All currently defined searches.

Examples SEARCH:LIST? might return :SEARCH:LIST SEARCH1,SEARCH2 indicating that Search 1 and Search 2 are defined.

SEARCH:SEARCH<x>:COPy (No Query Form)

This command (no query form) copies the search criteria to or from the trigger. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:COPy {SEARCHtotrigger|TRIGgertosearch}

Arguments SEARCHtotrigger copies the search criteria to the trigger.

TRIGgertosearch copies the trigger criteria to the search.

Examples SEARCH:SEARCH1:COPY TRIGGERTOSEARCH copies the trigger criteria to the search 1 criteria.

SEARCH:SEARCH1:COPY SEARCHTOTRIGGER copies the search criteria to the trigger.

SEARCH:SEARCH<x>:NAVigate (No Query Form)

This command sets the navigation action for search marks. The NONE action is the default setting when no action is being taken. The search number is specified by x.

Group Search and Mark

Syntax SEARCH:SEARCH<x>:NAVigate {NEXT|PREVious|MIN|NONE|MAX}

Arguments NEXT goes to the next search mark.

PREVious goes to the previous search mark.

MIN goes to the first search mark.

NONE is the default setting when no action is being taken.

MAX goes to the last search mark.

Examples SEARCH:SEARCH1:NAVigate NEXT goes to the next search mark.

SEARCH:SEARCH<x>:TOTAL? (Query Only)

This query-only command returns the total number of found search marks for this search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TOTAL?
```

Examples `SEARCH:SEARCH1:TOTAL?` might return `SEARCH:SEARCH1:TOTAL 7`, indicating that there are 7 matches for search 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:CONDition

This command specifies a field or condition for an ARINC429 bus to search on. The search number is specified by x.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:CONDition
{SOW|LABel|DATa|LABELANDDATA|EOW|ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:CONDition?
```

Arguments `SOW` specifies a search for the first bit of a word.

`LABel` specifies a search for a matching label.

`DATa` specifies a search for matching packet data fields.

`LABELANDDATA` specifies a search for a matching label and matching packet data field(s).

`EOW` specifies a search for the 32nd bit of a word.

`ERRor` specifies a search for a specified error condition.

NOTE. *The type of error searched for is specified by* *`SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:ERRTYPe`*.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:CONDITION DATA` specifies finding packets that contain matching data field(s).

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:CONDITION?` might return SOW, indicating that the bus is being searched for the first bit of each word in the packet.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:HIVALue

This command sets or queries the high value when searching on an ARINC429 data field. The search number is specified by x. The search condition must be set to DATa or LABELANDDATA, and the data qualifier must be INrange or OUTrange.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:HIVALue?
```

Arguments `<QString>` is the label value.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:HIVALUE "XXXXXXXXXXXXXXX1000"` sets the value to XXXXXXXXXXXXXXX1000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:HIVALUE?` might return "XXXXXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:QUALifier

This command sets or queries the qualifier to be used when searching on data in the DATA field for an ARINC429 bus signal. The search number is specified by x. The search condition must be set to DATa or LABELANDDATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:
QUALifier {EQual|UNEQual|LESSthan|MOREthan
|LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:QUALifier?
```

Arguments Arguments are the available data qualifiers.

NOTE. *The search qualifier only applies to the bits defined as the data field via the bus data field format specifier (using* `BUS:B<x>:ARINC429A:DATAFORmat`*).*

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:QUALIFIER LESSTHAN` sets the data qualifier to less than.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:QUALIFIER?` might return EQUAL, indicating that the data qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:VALue

This command sets or queries the low value when searching on an ARINC429 data field. The search number is specified by x. The search condition must be set to DATa or LABELANDDATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:VALue?
```

Arguments `<QString>` is the label value.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:VALUE "XXXXXXXXXXXXXXX1000"` sets the value to XXXXXXXXXXXXXXX1000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:DATA:VALUE?` might return "XXXXXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:ERRTYPe

This command sets or queries the error type when searching on an ARINC429 bus signal. The search number is specified by x. The search condition must be set to ERRor.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:ERRTYPe
{ANY|PARity|WORD|GAP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:ERRTYPe?
```

Arguments

`ANY` sets the error type to match any of the other available error types.

`PARity` sets the error type to match on parity errors (parity value results in even parity count for a word).

`WORD` sets the error type to match on word errors (any unframed or unknown decode data).

`GAP` sets the error type to match on gap violations (less than 4 bits idle time between two packets on the bus).

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:ERRTYPE PARITY` sets the error type to match on parity errors.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:ERRTYPE?` might return ANY, indicating that any error condition will produce a match.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue

This command sets or queries the high value when searching on an ARINC429 label field. The search number is specified by x. The search condition must be set to LABel, and the label qualifier must be INrange or OUTrange.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue?
```

Arguments

<QString> is the label value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:HIVALUE "XXXX1010" sets the value to XXXX1010.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:HIVALUE? might return "XXXXXXXX", indicating that the value is XXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:QUALifier

This command sets or queries the qualifier to be used when searching on label data for an ARINC429 bus signal. The search number is specified by x. The search condition must be set to LABel or LABELANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:
QUALifier {EQual|UNEQual|LESSthan|MOREthan
|LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:QUALifier?
```

Arguments

Arguments are the available data qualifiers.

NOTE. *If the search condition is set to LABELANDDATA, the label qualifier will be locked to Equal until the search condition is changed again.*

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:QUALIFIER LESSTHAN sets the label qualifier to less than.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:QUALIFIER? might return EQUAL, indicating that the label qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:VALue

This command sets or queries the low value when searching on an ARINC429 label field. The search number is specified by x. The search condition must be set to LABel or LABELANDDATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:VALue
<QString>
```

Arguments <QString> is the label value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:VALUE "XXXX1010" sets the value to XXXX1010.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:LABEL:VALUE? might return "XXXXXXXX", indicating that the value is XXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SDI:VALue

This command sets or queries the label when searching on an ARINC429 SDI field. The search number is specified by x. The search condition must be set to DATa or LABELANDDATA, and the data format must be set to DATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SDI:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SDI:VALue?
```

Arguments <QString> is the label value.

NOTE. *The SDI field is only present when the selected data field format is DATA (using BUS:B<x>:ARINC429A:DATAFORmat). Also, the stored QString is reset to its default value whenever the data field format is changed.*

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SDI:VALUE "X0" sets the value to X0.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SDI:VALUE? might return "XX", indicating that the value is XX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SSM:VALue

This command sets or queries the label when searching on an ARINC429 SSM field. The search number is specified by x. The search condition must be set to DATa or LABELANDDATA, and the data format must be set to DATA or SDIDATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SSM:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SSM:VALue?
```

Arguments <QString> is the label value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SSM:VALUE "X0" sets the value to X0.

SEARCH:SEARCH1:TRIGGER:A:BUS:ARINC429A:SSM:VALUE? might return "XX", indicating that the value is XX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition

This command sets or queries the condition (word select / frame sync, or matching data) to be used when searching on an audio bus signal. The search number is specified by x.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition {SOF|DATa}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition?
```

Arguments SOF specifies to search on a word select or start of frame (depending on Audio Type).

DATA specifies to search on matching data.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:CONDition SOF sets conditions to search on start of frame.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:CONDition? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:CONDITION DATA indicating settings for a search on DATA.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HITDMVALue

This command sets or queries the binary data string for the high data word to be used when searching on an TDM audio bus signal. The search condition must be set to DATa using SEARCH:SEARCH{x}:TRIGger:A:BUS:AUDio:CONDition.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HITDMVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HITDMVALue?
```

Arguments <QString> is the binary data string for the high data word to be used when searching on an TDM audio bus signal.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:HITDMVALue "XXXXXXXXXXXXXXXXXXXX1100" sets the high value to 1100.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:HITDMVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:HITDMVALUE "XXXXXXXXXXXXXXXXXXXX1010" indicating the high value is set to 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HIVALue

This command sets or queries the binary data string for the high data word to be used when searching on an audio bus signal. The search condition must be set to DATa using :SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HIVALue?
```

Arguments <QString> specifies the upper word value.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:HIVALue “XXXX” sets the HIVALUE to XXXX.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:HIVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:HIVALUE "TEST_001101", indicating the HIVALUE is set to TEST_001101.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:OFFSet

This command sets or queries the data offset value (TDM channel) to be used when searching on a TDM type audio bus signal. The search condition must be set to DATa using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:TDM:CONDition. The search number is specified by x.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:OFFSet?
```

Arguments <NR1> is the data offset value.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:OFFSet 2 sets the data offset value to 2.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:OFFSet? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:OFFSET 1 indicating data offset value is set to TDM channel 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:QUALifier

This command sets or queries the qualifier to be used when searching on an audio bus signal. The search condition must be set to DATa using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:{NONTdm|TDM}:CONDition. The search number is specified by x.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:
AUDio:DATa:QUALifier {EQual|UNEQual|
LESSthan|MOREthan|LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:QUALifier?
```

Arguments

LESSthan sets the qualifier to less than.

MOREthan sets the qualifier to greater than.

EQual sets the qualifier to equal.

UNEQual sets the qualifier to not equal.

LESSEQual sets the qualifier to less than or equal.

MOREEQual sets the qualifier to greater than or equal.

INrange sets the qualifier to in range.

OUTrange sets the qualifier to out of range.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:QUALifier LESSthan sets the qualifier to LESSthan.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:QUALifier? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:QUALIFIER EQUAL indicating the qualifier is set to EQUAL.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:TDMVALue

This command sets or queries the binary data string for the single or low data word to be used when searching on an TDM audio bus signal. The search condition must be set to DATA using SEARCH:SEARCH{x}:TRIGger:A:BUS:AUDio:CONDition.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:TDMVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:TDMVALue?
```

Arguments <QString> is the binary data string for the single or low data word to be used when searching on an TDM audio bus signal.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:TDMVALue "1100" sets the TDMVALUE to 1100.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:TDMVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:TDMVALUE "XXXXXXXXXXXXXXXXXXXX1010", indicating the TDM value is 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:VALue

This command sets or queries the binary data string for the single or low data word to be used when searching on an audio bus signal. The search condition must be set to DATA using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:{NONTdm|TDM}:CONDition. The search number is specified by x.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:VALue?
```

Arguments <QString> is the lower word value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:VALue “X0X011” sets the data value to X0X011.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:VALUE "XXXXXXXXXXXXXXXXXXXXXXXX", indicating the data value is set to XXXXXXXXXXXXXXXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:WORD

This command sets or queries the alignment of the data (left, right or either) to be used when searching on a non-TDM type audio bus signal. The search condition must be set to DATA using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:NONTdm:CONDition. The search number is specified by x.

Conditions

Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:WORD
{EITher|LEFt|RIGht}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:WORD?
```

Arguments

EITher aligns the data to either left or right.

LEFt aligns the data to the left.

RIGht aligns the data to the right.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:WORD LEFT aligns the data to the left.

SEARCH:SEARCH1:TRIGger:A:BUS:AUDio:DATa:WORD? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUDIO:DATA:WORD EITHER indicating the data is aligned either to the left or right.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

This command sets or queries the field or condition for which to search an AutoEthernet.

Conditions Requires option SR-AUTOENET

Most of the other SEARCH:SEARCH<x>:TRIGger:A:BUS:B:AUTOETHERnet commands are impacted by the setting of this command.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:
CONDition {SSD|SFD|MACADDRess|MACLENgth|IPHeader|
TCPHeader|DATa|EOP|FCSError|QTAG}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition?
```

Arguments `Search<x>` is the Search number.

`SSD` specifies to search for Start of frame.

`SFD` specifies to search for Start of frame delimiter.

`MACADDRess` specifies to search for MAC addresses field.

`MACLENgth` specifies to search for MAC length/type field.

`IPHeader` specifies to search for IP header field.

`TCPHeader` specifies to search for TCP header field.

`DATa` specifies to search for TCP/IPv4 or MAC protocol client data field.

`EOP` specifies to search for End of Packet field.

`FCSError` specifies to search for Frame Check Sequence Error (CRC) field.

`QTAG` specifies to search for IEEE 802.1Q (VLAN) control information field.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:CONDITION MACADDRESS` specifies MACADDRess as the field within an AutoEthernet frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:CONDITION?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:CONDITION DATA`, indicating that DATA is the currently specified field within an AutoEthernet frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue

This command sets or queries the binary data value to be used when searching on an AutoEthernet bus signal. The search condition must be set to DATA and the data qualifier to inside or outside range.

Conditions Requires option SR-AUTOENET

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue?
```

Arguments `Search<x>` is the Search number.

`QString` specifies the binary data value to be used when searching on an AutoEthernet bus signal.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue "101011"` sets the data HIVALue to 101011.

`SEARCH:SEARCH1:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue?` might return`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:HIVALUE "10110"`.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:OFFSet

This command sets or queries the data offset value, in bytes, to use when searching on the AutoEthernet data field. The search condition needs to be set to DATa.

Conditions Requires option SR-AUTOENET

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:OFFSet?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SOUrceport:VALue

Arguments Search<x> is the Search number.

NR1 sets the data offset value, in bytes. It is an integer value whose minimum and default values are -1 (don't care) and maximum is 1,499.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:OFFSEt 36 sets the data offset to 36 bytes.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:OFFSET? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:OFFSET -1, indicating that the data offset value is the default value (-1), meaning "don't care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to use when searching on the AutoEthernet bus signal. The search condition needs to be set to DATa.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to DATa.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments Search<x> is the Search number.

NR1 sets the data length, in bytes. It is an integer value whose minimum and default values are 1, and the maximum value is 16.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:SIZE 4 specifies 4 as the number of contiguous TCP/IPv4/MAC client bytes to use when searching on the AutoEthernet data field.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:SIZE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:SIZE 8.

SEARCH:SEARCHx>:TRIGger:A:BUS:AUTOETHERnet:DATa:VALue

This command sets or queries the binary value to use when searching on the AutoEthernet bus signal. The search condition needs to be set to DATa.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to DATa.

Group Search and Mark

Syntax

```
SEARCH:SEARCHx>:TRIGger:A:BUS:AUTOETHERnet:DATa:VALue
<QString>
SEARCH:SEARCHx>:TRIGger:A:BUS:AUTOETHERnet:DATa:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:OFFSet

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe

Arguments `Search<x>` is the Search number.

`QString` is a quoted string where the allowable characters are 0, 1, and X. The allowable number of characters depends on the setting for size (using SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe). The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

```
SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:VALUE
"00001000"
```

specifies 00001000 as the value to use when triggering on the AutoEthernet binary data field, assuming that SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:SIZe is set to 1 byte.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:DATA:VALUE "00001000"`, indicating 0000100 is the currently specified value used when searching on the AutoEthernet binary data field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:DESTinationaddr:VALue

This command sets or queries the binary destination address value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to IPHeader.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to IPHeader.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:
DESTinationaddr:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:
DESTinationaddr:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments `Search<x>` is the Search number.

`QString` is a quoted string of up to 32 characters that sets the IP address, where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:DESTINATIONADDR:VALUE "11000000101010000000000000000001"` specifies 192.168.0.1 as the value to use when searching on the AutoEthernet IPv4 header address destination field.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:DESTINATIONADDR:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:DESTINATIONADDR:VALUE "11000000101010000000000000000001"`, indicating that 192.168.0.1 is the currently specified value used when searching on the AutoEthernet IPv4 header address destination field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:PROTOcol:VALue

This command sets or queries the binary protocol value to use when searching on the Ethernet bus signal. The search condition needs to be set to IPHeader.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to IPHeader.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:
PROTOcol:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:
PROTOcol:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments `Search<x>` is the Search number.

`QString` is a quoted string of up to eight (8) characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:PROTOCOL:VALUE "01010010"` specifies 01010010 as the value to use when searching on the AutoEthernet IP header protocol field.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:PROTOCOL:VALUE?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:PROTOCOL:VALUE "XXXXXXXX"`, indicating that the trigger value has been set to "don't care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:SOUrceaddr:VALue

This command sets or queries the binary source address value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to IPHeader.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to IPHeader.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:
SOUrceaddr:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:
SOUrceaddr:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments

Search<x> is the Search number.

QString is a quoted string of up to 32 characters, where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:SOURCEADDR:VALUE "11000000101010000000000000000001" specifies to use the value of 192.168.0.1 when searching on the AutoEthernet IP header address source field.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:SOURCEADDR:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:IPHEADER:SOURCEADDR:VALUE "11000000101010000000000000000001", indicating that the search value has been set to 192.168.0.1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:DESTination:VAl

This command sets or queries the binary MAC address destination value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to MACADDRess.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to MACADDRess.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:
DESTination:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:
DESTination:VALue ?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments

Search<x> is the Search number.

QString is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

```
SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:
DESTINATION:VALUE
```

"XXXXXXXX00110101111111000000011110101011001000" specifies to use the value of XX:35:FC:07:AA:C8 hex when searching on the Ethernet MAC address destination field.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:DESTINATION:VALUE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:DESTINATION:VALUE
"XXXXXXXX00110101111111000000011110101011001000", indicating a MAC address destination field value of XX:35:FC:07:AA:C8 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:SOUrce:VALue

This command sets or queries the binary MAC address source value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to MACADDRess.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to MACADDRess.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:SOUrce:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:SOUrce:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments Search<x> is the Search number.

QString is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples
```
SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:SOURCE:VALUE "XXXXXXXX00110101111111000000011110101011001000"
```
specifies to use the value of XX:35:FC:07:AA:C8 hex when searching on the Ethernet MAC address source field.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:SOURCE:VALUE? might return
SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:ADDRESS:SOURCE:VALUE

”XXXXXXXX00110101111111000000011110101010011001000”, indicating a MAC address source field value of XX:35:FC:07:AA:C8 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:HIVALue

This command sets or queries the binary MAC length high value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to MACADDRess.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to MACADDRess.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:
HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:
HIVALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments `Search<x>` is the Search number.

`QString` is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE "XXXXXXXX00001000"` specifies to use the hexadecimal value XX08 when searching on the AutoEthernet MAC length.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE "XXXXXXXX00001000"`, indicating an AutoEthernet MAC length value of XX08 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:VALue

This command sets or queries the MAC length value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to MACADDRess.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:
VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments `Search<x>` is the Search number.

`QString` is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE "XXXXXXXX00001000"` specifies to use the hexadecimal value XX08 when searching on the AutoEthernet MAC length.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE?` might return
`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:MAC:LENGTH:HIVALUE "XXXXXXXX00001000"`, indicating an AutoEthernet MAC length value of XX08 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QTAG:VALue

This command sets or queries the binary Q-tag information to use when searching on an AutoEthernet bus signal. The search condition needs to be set to QTAG.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to QTAG.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QTAG:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QTAG:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments Search<x> is the Search number.

QString is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QTAG:VALUE "XXXXXXXXXXXXXXXXXXXX010010001010" specifies to use the value of hexadecimal XXXXX48A when searching on the AutoEthernet Q-Tag field.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QTAG:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QTAG:VALUE "XXXXXXXXXXXXXXXXXXXX010010001010", indicating that hexadecimal XXXXX48A has been set as the AutoEthernet Q-Tag field search value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QUALifier

This command sets or queries the qualifier to be used when searching on an AutoEthernet bus signal. The search condition must be set to Client Data or MAC Length/Type.

Conditions Requires option SR-AUTOENET

The search condition must be set to Client Data or MAC Length/Type.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QUALifier
{LESSthan|MOREthan|EQual|UNEQual| LESSEQual|MOREEQual|
INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QUALifier?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments Search<x> is the Search number.

LESSthan sets the qualifier to less than.

MOREthan sets the qualifier to greater than.

EQual sets the qualifier to equal.

UNEQual sets the qualifier to not equal.

LESSEQual sets the qualifier to less than or equal.

MOREEQual sets the qualifier to greater than or equal.

INrange sets the qualifier to in range.

OUTrange sets the qualifier to out of range.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QUALIFIER LESSTHAN sets the qualifier to "less than".

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QUALIFIER? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:QUALIFIER EQUAL, indicating that the qualifier is set to EQUAL.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:ACKnum:VALue

This command sets or queries the binary ack number value to use when searching on an AutoEthernet bus signal. The default is all X's (don't care). The search condition needs to be set to TCPHeader.

Conditions

Requires option SR-AUTOENET

The search condition needs to be set to TCPHeader.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:
ACKnum:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:
ACKnum:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments

Search<x> is the Search number.

QString is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:ACKNUM:VALUE "XXXXXXXXXXXXXXXXXXXX00001000" specifies hexadecimal

XXXXXX08 as the value to use when searching on the AutoEthernet TCP header acknowledgement number.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:ACKNUM:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:ACKNUM:VALUE "XXXXXXXXXXXXXXXXXXXX00001000", indicating that hexadecimal XXXXXX08 has been specified as the value to use when searching on the Auto Ethernet TCP header acknowledgement number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:DESTinationport:VA

This command sets or queries the binary destination port value to use when searching on the AutoEthernet TCP header destination port number. The search condition needs to be set to TCPHeader.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to TCPHeader.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:
DESTinationport:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:
DESTinationport:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments Search<x> is the Search number.

QString is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:DESTINATIONPORT:VALUE "XXXXXXXX00100010" specifies to use the value of hexadecimal XX22 when searching on the AutoEthernet TCP header destination port number.

SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:DESTINATIONPORT:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:DESTINATIONPORT:VALUE "XXXXXXXX00100010", indicating that

hexadecimal XX22 has been set as the value to use when searching on the AutoEthernet TCP header destination port number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SEQnum:VALue

This command sets or queries the TCP header sequence number value to use when searching on the AutoEthernet TCP header destination port number. The search condition needs to be set to TCPHeader.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to TCPHeader.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:
SEQnum:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:
SEQnum:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments `Search<x>` is the Search number.

`QString` is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SEQNUM:VALUE "XXXXXXXXXXXXXXXXXXXX000100010001" specifies to use the value of hexadecimal XXXXX111 when searching on the AutoEthernet TCP header sequence number.

`SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SEQNUM:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SEQNUM:VALUE "XXXXXXXXXXXXXXXXXXXX0010010100"`, indicating that hexadecimal XXXXX111 has been specified as the value to use when searching on the AutoEthernet TCP header sequence number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SOUrceport:VALue

This command sets or queries the binary source port value to use when searching on an AutoEthernet bus signal. The search condition needs to be set to TCPHeader.

Conditions Requires option SR-AUTOENET

The search condition needs to be set to TCPHeader.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:
SOUrceport:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:
SOUrceport:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition

Arguments `Search<x>` is the Search number.

`QString` is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits (LSB), leaving any unspecified upper bits unchanged.

Examples

```
SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SOURCEPORT:VALUE
"XXXX000010100110"
```

specifies to use the value of hexadecimal X0A6 when searching on the AutoEthernet TCP header source port number.

```
SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SOURCEPORT:VALUE
```

might return

```
SEARCH:SEARCH1:TRIGGER:A:BUS:AUTOETHERnet:TCPHEADER:SOURCEPORT:VALUE
"XXXXX01001010110"
```

, indicating that hexadecimal X0A6 has been specified as the value to use when searching on the Ethernet TCP header source port number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition

This command sets or queries the search condition for a CAN bus. The search number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition
{SOF|FRAMEtype|IDentifier|DATa|IDANDDATA|EOF|ERRor|FDBITS}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition?
```

Arguments Arguments specify the CAN bus trigger condition.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:CAN:CONDition EOF` sets the CAN bus trigger condition to end of frame.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:CONDition?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:CONDITION SOF`, indicating the CAN bus trigger condition is set to start of frame.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:DIRection

This command specifies the CAN search type to be valid on a Read, Write, or Either condition. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:DIRection
{READ|WRITE|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:DIRection?
```

Arguments `READ` specifies the read direction.

`WRITE` specifies the write direction.

`NOCARE` specifies either data direction.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:DIRection READ` sets the data direction to `READ`.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:DIRection?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:DIRECTION NOCARE`, indicating the data direction is set to either data direction.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:OFFSet

This command sets or queries the data offset value, in bytes, to use when searching on the CAN data field. The search number is specified by x. The search condition must be set to DATA or IDANDDATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:OFFSet?
```

Related Commands BUS:B<x>:CAN:STANDard

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:SIZe

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue

Arguments <NR1> is an integer whose minimum and default values are -1 (don't care) and maximum is up to 7 (for CAN 2.0) or up to 63 (for ISO CAN FD and Non-ISO CAN FD). The maximum is dependent on the number of bytes being matched and the CAN standard selected. Its value is calculated as [Absolute Maximum] - [Data Match Size]. For CAN 2.0, the absolute maximum is 8 bytes. For ISO CAN FD and Non-ISO CAN FD, the absolute maximum is 64 bytes. The minimum data match size is 1 byte, which produces the ranges listed above. Increasing the data match size above 1 byte will adjust the range of valid data offset values accordingly.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:OFFSET 5` sets the CAN data offset to 5 bytes.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:OFFSET?` might return 7, indicating the CAN data offset is 7 bytes. If the CAN standard is set for CAN 2.0 and the search data size is set to 3, the maximum value for the data offset will be 5 (8 - 3 = 5). If the CAN standard is set for ISO CAN FD or Non-ISO CAN FD and the search data size is set to 8, the maximum value for the data offset will be 56 (64 - 8 = 56).

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:QUALifier

This command sets or queries the CAN bus trigger data qualifier to be used when searching on a CAN bus signal. The search number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:QUALifier
{EQUal|LESSEQual|MOREEQua|UNEQual|LESSthan|MOREthan}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:QUALifier?
```

Arguments Arguments are the data qualifier types.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:QUALifier UNEQual` sets the data qualifier to unequal.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:QUALifier?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:QUALIFIER EQUAL`, indicating that the data qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to be used when searching on a CAN bus signal. The search condition must be set to IDANDDATA or DATA. The search number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:SIZe?
```

Arguments `<NR1>` specifies the data size.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:SIZe 1` sets the data size to 1.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:SIZe?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:SIZE 1`, indicating the data size is set to 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue

This command sets or queries the binary data value to be used when searching on a CAN bus signal. The search condition must be set to IDANDDATA OR DATA.

Group Search and Mark

Syntax `SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue <QString>`

Arguments `<QString>`

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:VALue "1111"` sets the data value to 1111.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:DATa:VALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:DATA:VALUE "1010"` indicating the data value is 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType

This command sets or queries the type of error condition for a CAN bus to search on. The search number is specified by x. The search condition must be set to ERRor.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType
{ACKMISS|BITSTUFFing|FORMERRor |ANYERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType?
```

Arguments

ACKMISS specifies a search based on a missing ACK field.

BITSTUFFing specifies a search based on a bit stuffing error.

FORMERRor specifies a search based on a CAN FD form error. To use this option, the CAN standard must be set to FDISO or FDNONISO.

ANYERRor specifies a search based on any error type.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:ERRTYPE ACKMISS specifies searching for any missing ACK fields.

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:ERRTYPE? might return ANYERROR, indicating that the bus is being searched for all error types.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit

This command sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to search on. The search number is specified by x. The search condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit?
```

Arguments

ONE filters CAN FD packets to only match those where the BRS bit has a value of 1 (fast data enabled).

ZERo filters CAN FD packets to only match those where the BRS bit has a value of 0 (fast data disabled).

NOCARE disables filtering of CAN FD packets on the BRS bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:BRSBit ONE specifies filtering CAN FD packets for those where the BRS bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:BRSBIT? might return NOCARE, indicating that CAN FD packets are not being filtered based on the BRS bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit

This command sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to search on. The search number is specified by x. The search condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit?
```

Arguments

`ONE` filters CAN FD packets to only match those where the ESI bit has a value of 1 (recessive).

`ZERo` filters CAN FD packets to only match those where the ESI bit has a value of 0 (dominant).

`NOCARE` disables filtering of CAN FD packets on the ESI bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:ESIBit ONE` specifies filtering CAN FD packets for those where the ESI bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FD:ESIBIT?` might return NOCARE, indicating that CAN FD packets are not being filtered based on the ESI bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEtype

This command sets or queries CAN bus trigger frame type to be used when searching on a CAN bus signal. The search condition must be set to FRAMEtype. The search number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEtype
{DATa|ERRor|OVERLoad|REMote}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEtype?
```

Arguments

Arguments are the available frame types.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:FRAMEtype ERROR` sets the frame type to error.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:FRAMEtype?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:FRAMETYPE DATA`, indicates the frame type is set to data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODe

This command sets or queries the CAN bus trigger identifier (address) mode to be used when searching on a CAN bus signal. The search number is specified by x. The search condition must be set to IDANDDATA or DATA.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODe
{EXTENDed|STandard}
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODe?
```

Arguments

`EXTENDed` specifies the extended identifier mode.

`STandard` specifies the standard identifier mode.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDentifier:MODe EXTEND` sets the identifier mode to extended.

`SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDentifier:MODe?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:IDENTIFIER:MODE STANDARD`, indicating the identifier mode is set to standard.

SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue

This command sets or queries CAN bus trigger identifier (address) value to be used when searching on a CAN bus signal. The search number is specified by x. The search condition must be set to IDANDDATA or DATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue?
```

Arguments <QString> is the identifier value.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDentifier:VALue "1010" sets the identifier value to 1010.

SEARCH:SEARCH1:TRIGger:A:BUS:CAN:IDentifier:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:CAN:IDENTIFIER:VALUE "101011", indicating the identifier value is 101011.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:BLUe:VALue

This command specifies the blue data string used for DPHY triggering if the trigger condition is on RGB packet.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:BLUe:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:BLUe:VALue?
```

Arguments <QString> is the identifier value. The default blue value is XXXXXXXXXX.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:BLUe:VALue "BE4" specifies BE4 as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:BLUe:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:BLUe:VALue "BE4", indicating that BE4 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:CONDition

This command sets or queries the trigger condition for an DPHY bus. The default trigger on condition is SOT.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:CONDition {SOT| EOT|
DATa| EOTPDATa| OPTIONALPARam| SCRambling| COMPression|
PACKets| BUSTURNAROUnd| ESCAPEMODe| STop| ERRors| MODe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:CONDition?
```

Arguments

SOT specifies SOT as the field within a DPHY frame to search on.

EOT specifies EOT as the field within a DPHY frame to search on.

DATa specifies DATa as the field within a DPHY frame to search on

EOTPDATa specifies EOTPDATa as the field within a DPHY frame to search on.

OPTIONALPARam specifies OPTIONALPARam as the field within a DPHY frame to search on.

SCRambling specifies SCRambling as the field within a DPHY frame to search on.

COMPression specifies COMPression as the field within a DPHY frame to search on.

PACKets specifies PACKets as the field within a DPHY frame to search on.

BUSTURNAROUnd specifies BUSTURNAROUnd as the field within a DPHY frame to search on.

ESCAPEMODe specifies ESCAPEMODe as the field within a DPHY frame to search on.

STop specifies STop as the field within a DPHY frame to search on.

ERRors specifies ERRors as the field within a DPHY frame to search on.

MODe specifies MODe as the field within a DPHY frame to search on.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:CONDition SOT` specifies SOT as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:CONDition?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:CONDition SOT`, indicating that SOT is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:SIZe

This command specifies the length of the data string in bytes to be used for an DPHY trigger if the trigger condition is DATA. The default data size is 1 byte and ranges between 1 and 5. The search number is specified by x.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:SIZe?
```

Arguments `<NR1>` specifies the data size.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:DATa:SIZe 2` specifies 2 as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:DATa:SIZe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:DATa:SIZe 2`, indicating 2 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:VALue

This command specifies the binary data string used for DPHY triggering if the trigger condition is DATA.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:VALue?
```

Arguments <QString> is the identifier value. The default value is XXXXXXXX.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:DATa:VALue "11110000" specifies 11110000 as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:DATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:DATa:VALue "11110000", indicating that 11110000 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ERRor:TYPe

This command sets or queries the error type for DPHY bus. The default error type condition is ANY.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ERRor:TYPe {ANY|ECC|CRC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ERRor:TYPe?
```

Arguments ANY specifies ANY as the field within a DPHY frame to search on.

ECC specifies ECC as the field within a DPHY frame to search on.

CRC specifies CRC as the field within a DPHY frame to search on

Examples SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:ERRor:TYPe ECC specifies ECC as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:ERRor:TYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:ERRor:TYPe ECC, indicating that ECC is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand

This command sets or queries the escape mode command type for DPHY bus.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand
{LPDT|ULPS|RESETTRIGger|ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand?
```

Arguments `LPDT` specifies LPDT as the field within a DPHY frame to search on.

`ULPS` specifies ULPS as the field within a DPHY frame to search on.

`RESETTRIGger` specifies RESETTRIGger as the field within a DPHY frame to search on

`ANY` specifies ANY as the field within a DPHY frame to search on. ANY is the default value.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand ULPS` specifies ULPS as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand ULPS`, indicating that ULPS is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:GREen:VALue

This command specifies the green data string used for DPHY triggering if the trigger condition is on RGB packet.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:GREen:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:GREen:VALue?
```

Arguments `<QString>` is the identifier value. The default green value is XXXXXXXXXX.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:GREen:VALue "243"` specifies 243 as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:GREen:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:GREen:VALue "243", indicating that 243 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:MODe:TYPe

This command sets or queries the mode type for DPHY bus. The default mode type condition is HS.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:MODe:TYPe {HS|LP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:MODe:TYPe?
```

Arguments HS specifies HS as the mode within a DPHY frame to search on. HS is default mode type condition.

LP specifies LP as the mode within a DPHY frame to search on.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:MODe:TYPe LP specifies LP as the mode within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:MODe:TYPe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:MODe:TYPe LP, indicating that LP is the currently specified mode within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:LIST

This command sets or queries the packet list for DPHY bus. The default packet list condition is VSYNCSTART.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:LIST
{VSYNCSTART| VSYNCEND| HSYNCSTART| HSYNCEND| EOTP| COLOROFF|
COLORON| SHUTDOWN| TURNON| GSW| GR| COMPRESSION| SCRAMBLING|
EXECUTEQUEue| DCSSW| DCSR| GLONGWRITE| DSINULL| DSIBLANk|
```

```
MAXRETSIZE| DCSGLONGWRITE| PICPARameter| COMPRESSEDPSTReam|
PPS101010| PPS121212| PPS565| PPS666| LPS666| PPS888|
YCBCR24| LPYCBCR24| YCBCR16| YCBCR12| ACKNowledge|
ERRORREPort| GSRR| GLONGREAD| DCSLONGREAD| DCSSRR|
FRAMESTARt| FRAMEEND| LINESTARt| LINEEND| GSP| CSINULL|
CSIBLANk| EMBEDDED| USEREIGHTB| RGB565| RGB555| RGB444|
RGB666| RAW6| RAW7| RAW8| RAW10| RAW12| RAW14| RAW16| RAW20|
RGB888| YUV420L8| YUV8BIT| YUV422B10| YUV420B10}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:LIST?
```

Arguments

VSYNCSTART specifies VSYNCSTART as the packet within DPHY frame to search on.

VSYNCEND specifies VSYNCEND as the packet within DPHY frame to search on.

HSYNCSTART specifies HSYNCSTART as the packet within DPHY frame to search on.

HSYNCEND specifies HSYNCEND as the packet within DPHY frame to search on.

EOTP specifies EOTP as the packet within DPHY frame to search on.

COLOROFF specifies COLOROFF as the packet within DPHY frame to search on.

COLORON specifies COLORON as the packet within DPHY frame to search on.

SHUTDOWN specifies SHUTDOWN as the packet within DPHY frame to search on.

TURNON specifies TURNON as the packet within DPHY frame to search on.

GSW specifies GSW as the packet within DPHY frame to search on.

GR specifies GR as the packet within DPHY frame to search on.

COMPRESSION specifies COMPRESSION as the packet within DPHY frame to search on.

SCRAMBLING specifies SCRAMBLING as the packet within DPHY frame to search on.

EXECUTEQUEue specifies EXECUTEQUEue as the packet within DPHY frame to search on.

DCSSW specifies DCSSW as the packet within DPHY frame to search on.

DCSR specifies DCSR as the packet within DPHY frame to search on.

GLONGWRITE specifies GLONGWRITE as the packet within DPHY frame to search on.

DSINULL specifies DSINULL as the packet within DPHY frame to search on.

DSIBLANk specifies DSIBLANk as the packet within DPHY frame to search on.

MAXRETSIZE specifies MAXRETSIZE as the packet within DPHY frame to search on.

DCSGLONGWRITE specifies DCSGLONGWRITE as the packet within DPHY frame to search on.

PICPARameter specifies PICPARameter as the packet within DPHY frame to search on.

COMPRESSEDPSTReam specifies COMPRESSEDPSTReam as the packet within DPHY frame to search on.

PPS101010 specifies PPS101010 as the packet within DPHY frame to search on.

PPS121212 specifies PPS121212 as the packet within DPHY frame to search on.

PPS565 specifies PPS565 as the packet within DPHY frame to search on.

PPS666 specifies PPS666 as the packet within DPHY frame to search on.

LPS666 specifies LPS666 as the packet within DPHY frame to search on.

PPS888 specifies LPS666 as the packet within DPHY frame to search on.

YCBCR24 specifies YCBCR24 as the packet within DPHY frame to search on.

LPYCBCR24 specifies LPYCBCR24 as the packet within DPHY frame to search on.

YCBCR16 specifies YCBCR16 as the packet within DPHY frame to search on.

YCBCR12 specifies YCBCR12 as the packet within DPHY frame to search on.

ACKNowledge specifies ACKNowledge as the packet within DPHY frame to search on.

ERRORREPort specifies ERRORREPort as the packet within DPHY frame to search on.

GSRR specifies GSRR as the packet within DPHY frame to search on.

GLONGREAD specifies GLONGREAD as the packet within DPHY frame to search on.

DCSLONGREAD specifies DCSLONGREAD as the packet within DPHY frame to search on.

DCSSRR specifies DCSSRR as the packet within DPHY frame to search on.

FRAMESTARt specifies FRAMESTARt as the packet within DPHY frame to search on.

FRAMEEND specifies FRAMEEND as the packet within DPHY frame to search on.

LINESTARt specifies LINESTARt as the packet within DPHY frame to search on.

LINEEND specifies LINEEND as the packet within DPHY frame to search on.

GSP specifies GSP as the packet within DPHY frame to search on.

CSINULL specifies CSINULL as the packet within DPHY frame to search on.

CSIBLANk specifies CSIBLANk as the packet within DPHY frame to search on.

EMBEDDED specifies EMBEDDED as the packet within DPHY frame to search on.

USEREIGHTB specifies USEREIGHTB as the packet within DPHY frame to search on.

RGB565 specifies RGB565 as the packet within DPHY frame to search on.

RGB555 specifies RGB555 as the packet within DPHY frame to search on.

RGB444 specifies RGB444 as the packet within DPHY frame to search on.

RGB666 specifies RGB666 as the packet within DPHY frame to search on.

RAW6 specifies RAW6 as the packet within DPHY frame to search on.

RAW7 specifies RAW7 as the packet within DPHY frame to search on.

RAW8 specifies RAW8 as the packet within DPHY frame to search on.

RAW10 specifies RAW10 as the packet within DPHY frame to search on.

RAW12 specifies RAW12 as the packet within DPHY frame to search on.

RAW14 specifies RAW14 as the packet within DPHY frame to search on.

RAW16 specifies RAW16 as the packet within DPHY frame to search on.

RAW20 specifies RAW20 as the packet within DPHY frame to search on.

RGB888 specifies RGB888 as the packet within DPHY frame to search on.

YUV420L8 specifies YUV420L8 as the packet within DPHY frame to search on.

YUV8BIT specifies YUV8BIT as the packet within DPHY frame to search on.

YUV422B10 specifies YUV422B10 as the packet within DPHY frame to search on.

YUV420B10 specifies YUV420B10 as the packet within DPHY frame to search on.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:PACKets:LIST VSYNCSTART specifies VSYNCSTART of set command as the packet within DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PACKets:LIST? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PACKets:LIST VSYNCSTART, indicating that VSYNCSTART is the currently the packet within DPHY frame to search on

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:TYPe

This command sets or queries the packet type for DPHY bus. The default packet type condition is SHORt.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:TYPe
{SHORt|LONG}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:TYPe?
```

Arguments `SHORt` specifies SHORt as the field within a DPHY frame to search on. SHORt is the default packet type condition.

`LONGl` specifies LONG as the field within a DPHY frame to search on.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:PACKets:TYPe LONG` specifies LONG as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PACKets:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PACKets:TYPe LONG`, indicating that LONG is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:NUMBer

This command specifies the pixel number to be used for DPHY triggering if the trigger condition is on pixel number pattern. The search number is specified by x.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:NUMBer <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:NUMBer?
```

Arguments `<NR1>` specifies the pixel number. The default pixel number value is 0

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:PIXel:NUMBer 1234` specifies 1234 as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PIXel:NUMBer?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PIXel:NUMBer 1234`, indicating that 1234 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:SEARCHOPTion

This command sets or queries the search pattern type for DPHY bus. The default search pattern type value is PIXELVALue.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:SEARCHOPTion
{PIXELVALue|PIXELNUMBer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:SEARCHOPTion?
```

Arguments `PIXELVALue` specifies SHORt as the field within a DPHY frame to search on.

`PIXELNUMBer` specifies LONG as the field within a DPHY frame to search on.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:PIXel:SEARCHOPTion PIXELNUMBer` specifies PIXELNUMBer as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PIXel:SEARCHOPTion?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:PIXel:SEARCHOPTion PIXELNUMBer`, indicating that PIXELNUMBer is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:RED:VALue

This command specifies the red data string used for DPHY triggering if the trigger condition is on RGB packet.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:RED:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:RED:VALue?
```

Arguments `<QString>` is the identifier value. The default red value is XXXXXXXXXX.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:RED:VALue "1A1"` specifies 1A1 as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:RED:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:RED:VALue "1A1"`, indicating that 1A1 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:WORDCOUNt:VALue

This command specifies the word count data string used for DPHY triggering if the trigger condition is set on any pixel packet.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:WORDCOUNt:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:WORDCOUNt:VALue?
```

Arguments `<QString>` is the identifier value. The default word count value is XXXXXXXXXXXXXXXX.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:WORDCOUNt:VALue "0003"` specifies 1A1 as the field within a DPHY frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:WORDCOUNt:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:WORDCOUNt:VALue "0003"`, indicating that 0003 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CB

This command specifies the Cb data string used for DPHY triggering if the trigger condition is on YCbCr packet.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CB <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CB?
```

Arguments

<QString> is the identifier value. The default YCBCR packets CB value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:CB "1BA" specifies 1BA as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:DPHY:YCBCR:CB? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:CB "1BA", indicating that 1BA is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CR

This command specifies the Cr data string used for DPHY triggering if the trigger condition is on YCbCr packet.

Conditions

Requires option SR-DPHY.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CR <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CR?
```

Arguments

<QString> is the identifier value. The default YCBCR packets CR value is XXXXXXXXXX.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:CR "2EE" specifies 2EE as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:CR? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:CR "2EE", indicating that 2EE is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:Y

This command specifies the Y data string used for DPHY triggering if the trigger condition is on YCbCr packet.

Conditions

Requires option SR-DPHY.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:Y <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:Y?
```

Arguments <QString> is the identifier value. The default YCBCR packets Y value is XXXXXXXXXX.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:Y "1AA" specifies 1AA as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:Y? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YCBCR:Y "1AA", indicating that 1AA is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:U

This command specifies the U data string used for DPHY triggering if the trigger condition is on YUV packet.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:U <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:U?
```

Arguments <QString> is the identifier value. The default YUV packets U value is XXXXXXXXXX.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:U "123" specifies 123 as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:U? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:U "123", indicating that 123 is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:V

This command specifies the V data string used for DPHY triggering if the trigger condition is on YUV packet.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:V <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:V?
```

Arguments <QString> is the identifier value. The default YUV packets V value is XXXXXXXXXX.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:V "12A" specifies 12A as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:V? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:V "12A", indicating that 12A is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:Y

This command specifies the Y data string used for DPHY triggering if the trigger condition is on YUV packet.

Conditions Requires option SR-DPHY.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:Y <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:Y?
```

Arguments <QString> is the identifier value. The default YUV packets Y value is XXXXXXXXXX.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:Y "81A" specifies 12A as the field within a DPHY frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:Y? might return :SEARCH:SEARCH1:TRIGger:A:BUS:DPHY:YUV:Y "81A", indicating that 81A is the currently specified field within a DPHY frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

This command specifies a field or condition for an Ethernet bus to search on. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition
{SFD|MACADDRess|MACLENgth|IPHeader|TCPHeader|DATa|EOP|
IDLe|FCSError|QTAG}
```

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition?
```

Related Commands Most of the other `TRIGger:A:BUS:B<x>:ETHERnet` commands are impacted by the setting of this command.

Arguments

`SFD` – Start of frame delimiter.

`MACADDRess` – MAC addresses field.

`MACLENgth` – MAC length/type field.

`IPHeader` – IP header field.

`TCPHeader` – TCP header field.

`DATa` – TCP/IPv4 or MAC protocol client data field.

`EOP` – End of Packet field.

`IDLe` – Idle field.

`FCSError` – Frame Check Sequence Error (CRC) field.

`QTAG` – IEEE 802.1Q (VLAN) control information field.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:CONDITION MACADDRESS` specifies `MACADDRess` as the field within an Ethernet frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:CONDITION?` might return `DATA`, indicating that DATA is the currently specified field within an Ethernet frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue

This command sets or queries the binary data value to be used when searching on an Ethernet bus signal. The search condition must be set to DATA and the data qualifier to inside or outside range. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue?
```

Arguments <QString> is the binary data value to be used when searching on an Ethernet bus signal.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:ETHERnet:DATa:HIVALue “101011” sets the data HIVALue to 101011.

SEARCH:SEARCH1:TRIGger:A:BUS:ETHERnet:DATa:HIVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:HIVALUE "10101".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet

This command specifies the data offset value, in bytes, to use when searching on the Ethernet data field. The search condition needs to be set to DATa. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe

Arguments <NR1> is an integer whose minimum and default values are -1 (don't care) and maximum is 1,499.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:OFFSEt 36 sets the data offset to 36 bytes.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:OFFSET? might return -1, indicating that the data offset value is the default value, -1, meaning "don't care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:QUALifier

This command specifies the qualifier to be used when searching on an Ethernet bus signal. The search condition must be set to DATa. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:QUALifier
{LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual|
INrange|OUTrange}
```

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:QUALifier?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments LESSthan sets the qualifier to less than.

MOREthan sets the qualifier to greater than.

EQual sets the qualifier to equal.

UNEQual sets the qualifier to not equal.

LESSEQual sets the qualifier to less than or equal.

MOREEQual sets the qualifier to greater than or equal.

INrange sets the qualifier to in range.

OUTrange sets the qualifier to out of range.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:QUALIFIER LESSTHAN sets the qualifier to "less than".

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:QUALIFIER? might return SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:QUALIFIER EQUAL indicating that the qualifier is set to EQUAL.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe

This command specifies the length of the data string, in bytes, to use when searching on the Ethernet bus signal. The search condition needs to be set to DATa. The search number is specified by x.

Conditions

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments

The minimum and default values are 1, and the maximum value is 16.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:SIZE 4 specifies 4 as the number of contiguous TCP/IPv4/MAC client bytes to use when searching on the Ethernet data field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:SIZE? might return 6, indicating that 6 is the currently specified number of contiguous TCP/IPv4/MAC client bytes to use when searching on the Ethernet data field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue

This command specifies the binary value to use when searching on the Ethernet bus signal. The search condition needs to be set to DATa. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe

Arguments `<QString>` is a quoted string where the allowable characters are 0, 1, and X. The allowable number of characters depends on the setting for size (using SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe). The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:VALUE "00001000"` specifies 00001000 as the value to use when triggering on the Ethernet binary data field, assuming that `TRIGGER:A:BUS:ETHERNET:DATA:SIZe` is set to 1 byte.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:DATA:VALUE?` might return `"00001000"`, indicating 0000100 is the currently specified value used when searching on the Ethernet binary data field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:DESTinationaddr:VALue

This command specifies the binary destination address value to use when searching on an Ethernet bus signal. The search condition needs to be set to `IPHeader`. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:
DESTinationaddr:VALue <QString>
```

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:
DESTinationaddr:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments `<QString>` is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:DESTINATIONADDR:VALUE "11000000101010000000000000000001"` specifies 192.168.0.1 as the value to use when searching on the Ethernet IPv4 header address destination field.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:DESTINATIONADDR:VALUE?` might return `"11000000101010000000000000000001"`, indicating that 192.168.0.1 is the currently specified value used when searching on the Ethernet IPv4 header address destination field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:PROTOcol:VALue

This command specifies the binary protocol value to use when searching on the Ethernet bus signal. The search condition needs to be set to `IPHeader`. The search number is specified by x.

NOTE. *Commonly used protocol values are 1 (ICMP), 2 (IGMP), 6 (TCP) and 17 (UDP).*

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:PROTOcol:
VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:PROTOcol:
VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments <QString> is a quoted string of up to 8 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:PROTOCOL:VALUE "01010010" specifies 01010010 as the value to use when searching on the Ethernet IP header protocol field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:PROTOCOL:VALUE? might return "XXXXXXXX", indicating that the trigger value has been set to "don't care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:SOUrceaddr:VALue

This command specifies the binary source address value to use when searching on an Ethernet bus signal. The search condition needs to be set to IPHeader. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:SOUrceaddr:
VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:SOUrceaddr:
VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments QString is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:SOURCEADDR:VALUE "11000000101010000000000000000001" specifies to use the value of 192.168.0.1 when searching on the Ethernet IP header address source field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:IPHEADER:SOURCEADDR:VALUE might return "11000000101010000000000000000001", indicating that the search value has been set to 192.168.0.1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:DESTination:VALue

This command specifies the binary MAC address destination value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess. The search number is specified by x.

NOTE. *MAC Addresses are 48-bit values such as 08:00:11:1E:C9:AE hex.*

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:
DESTination:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:
DESTination:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments <QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:ADDRESS:DESTINATION:VALUE "XXXXXXXX00110101111111000000011110101010011001000" specifies to use the value of XX:35:FC:07:AA:C8 hex when searching on the Ethernet MAC address destination field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:ADDRESS:DESTINATION:VALUE? might return "XXXXXXXX00110101111111000000011110101010011001000", indicating a MAC address destination field value of XX:35:FC:07:AA:C8 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:SOUrce:VALue

This command specifies the binary MAC address source value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess. The search number is specified by x.

NOTE. *MAC Addresses are 48-bit values such as 08:00:11:1E:C9:AE hex.*

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:SOUrce:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:SOUrce:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments <QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:ADDRESS:SOURCE:VALUE "XXXXXXXX00110101111111000000011110101010110010000" specifies to use the value of XX:35:FC:07:AA:C8 hex when searching on the Ethernet MAC address source field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:ADDRESS:SOURCE:VALUE? might return "XXXXXXXX00110101111111000000011110101010110010000", indicating a MAC address source field value of XX:35:FC:07:AA:C8 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:HIVALue

This command specifies the binary MAC length high value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:HIVALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments <QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:LENGTH:HIVALUE "XXXXXXXX00001000" specifies to use the hexadecimal value XX08 when searching on the Ethernet MAC length.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:LENGTH:HIVALUE? might return "XXXXXXXX00001000", indicating an Ethernet MAC length value of XX08 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:VALue

This command specifies the MAC length value to use when searching on an Ethernet bus signal. The search condition needs to be set to MACADDRess. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments QString is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:LENGTH:HIVALUE "XXXXXXXX00001000" specifies to use the hexadecimal value XX08 when searching on the Ethernet MAC length.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:MAC:LENGTH:HIVALUE? might return "XXXXXXXX00001000", indicating an Ethernet MAC length value of XX08 hex.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:QTAG:VALue

This command specifies the binary Q-tag information to use when searching on an Ethernet bus signal. The search condition needs to be set to QTAG. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:QTAG:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:QTAG:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments <QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:QTAG:VALUE "XXXXXXXXXXXXXXXXXXXX010010001010" specifies to use the value of hexadecimal XXXXX48A when searching on the Ethernet Q-Tag field.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:QTAG:VALUE? might return "XXXXXXXXXXXXXXXXXXXX010010001010", indicating that hexadecimal XXXXX48A has been set as the Ethernet Q-Tag field search value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:ACKnum:VALue

This command specifies the binary ack number value to use when searching on an Ethernet bus signal. The default is all X's (don't care). The search condition needs to be set to TCPHeader. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:ACKnum:
VALue <QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:ACKnum:VALue?

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments QString is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:ACKNUM:VALUE "XXXXXXXXXXXXXXXXXXXX00001000" specifies hexadecimal XXXXXX08 as the value to use when searching on the Ethernet TCP header acknowledgement number.

TRIGGER:A:BUS:ETHERNET:TCPHEADER:ACKNUM:VALUE? might return "XXXXXXXXXXXXXXXXXXXX00001000", indicating that hexadecimal XXXXXX08 has been specified as the value to use when searching on the Ethernet TCP header acknowledgement number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:DESTinationport:VALue

This command specifies the binary destination port value to use when searching on the Ethernet TCP header destination port number. The search condition needs to be set to TCPHeader. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:
DESTinationport:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:
DESTinationport:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments <QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:DESTINATIONPORT:VALUE "XXXXXXXX00100010" specifies to use the value of hexadecimal XX22 when searching on the Ethernet TCP header destination port number.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:DESTINATIONPORT:VALUE? might return "XXXXXXXXXXXXXXXX", indicating that hexadecimal XX22 has been set as the value to use when searching on the Ethernet TCP header destination port number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SEQnum:VALue

This command specifies the binary sequence number value to use when searching on an Ethernet bus signal. The default is all X's (don't care). The search condition needs to be set to TCPHeader. The search number is specified by x.

Conditions

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SEQnum:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SEQnum:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments

QString is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:SEQNUM:VALUE "XXXXXXXXXXXXXXXXXXXX000100010001" specifies to use the value of hexadecimal XXXXX111 when searching on the Ethernet TCP header sequence number.

SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:SEQNUM:VALUE? might return "XXXXXXXXXXXXXXXXXXXXXX0010010100", indicating that hexadecimal XXXXX111 has been specified as the value to use when searching on the Ethernet TCP header sequence number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SOUrceport:VALue

This command specifies the binary source port value to use when searching on an Ethernet bus signal. The search condition needs to be set to `TCPHeader`. The search number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:
SOUrceport:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:
SOUrceport:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition

Arguments `<QString>` is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:SOURCEPORT:VALUE "XXXX000010100110"` specifies to use the value of hexadecimal X0A6 when searching on the Ethernet TCP header source port number.

`SEARCH:SEARCH1:TRIGGER:A:BUS:ETHERNET:TCPHEADER:SOURCEPORT:VALUE?` might return "XXXXX01001010110", indicating that hexadecimal X0A6 has been specified as the value to use when searching on the Ethernet TCP header source port number.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:HIVALue

This command sets the high binary address value to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:HIVALue?
```

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:HIVALue XXXXXXX specifies high binary address value as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:HIVALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:HIVALue XXXXXXX, indicating the high binary address value as XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:VALue

This command sets the binary address value to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:VALue?
```

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:VALue XXXXXXX specifies binary address value as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ADDress:VALue XXXXXXX, indicating the binary address value as XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:CONDition

This command sets or queries the search type for the specified search. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:CONDition {SYNC|
HANDSHAKEPacket| SPECIALPacket| ERRor| TOKENPacket|
DATAPacket| RESET| CONNect| CONTROLMESSage| SUSPEND|
DEVICEChirp| HOSTChirp| PORTReset| PORTConfiguration|
RESUME| EOP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:CONDition?
```

Arguments

`SYNC` specifies the search condition as Sync.

`HANDSHAKEPacket` specifies the search condition as Handshake Packet.

`SPECIALPacket` specifies the search condition as Special Packet.

`ERRor` specifies the search condition as Error.

`TOKENPacket` specifies the search condition as Token (Address) Packet.

`DATAPacket` specifies the search condition as Data Packet.

`RESET` specifies the search condition as Reset.

`CONNect` specifies the search condition as connect.

`CONTROLMESSage` specifies the search condition as Control Message.

`SUSPEND` specifies the search condition as Suspend.

`DEVICEChirp` specifies the search condition as Device chirp.

`HOSTChirp` specifies the search condition as Host Chirp.

`PORTReset` specifies the search condition as Port Reset.

`PORTConfiguration` specifies the search condition as Port Configuration.

`RESUME` specifies the search condition as Resume.

`RAP` specifies the search condition as Register Access Protocol.

`EOP` specifies the search condition as End of Packet.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:CONDition SYNC` specifies sync as the field within a eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:CONDition? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:CONDition STARt, indicating that STARt is the currently specified field within a eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:HIVALue

This command sets the high binary data value to be used when triggering on a eUSB bus signal. The trigger condition must be set to DATA. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:HIVALue?
```

Arguments <QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:HIVALue XXXXXXX specifies high binary data value as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:HIVALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:HIVALue XXXXXXX, indicating the high binary data value as XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:OFFSet

This command sets the byte offset to look for a data pattern at, in bytes, to be used when triggering on a eUSB bus signal. The trigger condition must be set to DATA. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:OFFSet?
```

Arguments

<NR1> specifies the byte offset. Default offset value is -1. The argument range is -1 to 1024

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:OFFSet -1 specifies the byte offset as -1 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:OFFSet? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:OFFSet XXXXXXX, indicating that the offset value is -1 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:QUAlifier

This command sets the qualifier to be used when triggering on a eUSB bus signal. The trigger condition must be set to IDANDDATA OR DATA. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:QUAlifier {EQual|
UNEQual| LESSthan| MOREthan| LESSEQual| MOREEQual| Inrange|
OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:QUAlifier?
```

Arguments

EQual specifies the qualifier as Equal.

UNEQual specifies the qualifier as Not Equal to.

LESSthan specifies the qualifier as Less Than.

MOREthan specifies the qualifier as More Than.

LESSEQual specifies the qualifier as Less Than or Equal to.

MOREEQual specifies the qualifier as More Than or Equal to.

Inrange specifies the qualifier as Inside Range.

OUTrange specifies the qualifier as Out of Range.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:QUAlifier EQual` specifies trigger value as Equal.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:QUAlifier?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:QUAlifier EQual`, indicating the trigger value as Equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:SIZe

This command sets the length of the data string, in bytes, to be used when triggering on a eUSB bus signal. The trigger condition must be set to DATA. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:SIZe?
```

Arguments `<NR1>` specifies the length of the data string. Default data size value is 1. The argument range is 1 to 16 bytes.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:SIZe 1` specifies the data size as 1 byte within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:SIZe?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:SIZe 1`, indicating that data size value is 1 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:TYPe

This command specifies the eUSB trigger type to be valid on any, data0, data1, data2, or mdata condition. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:TYPe {ANY| DATA0|
DATA1| DATA2| MDATA}
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:TYPe?

Arguments

ANY specifies either a DATA0 or DATA1 data packet type.

DATA0 specifies a DATA0 data packet type.

DATA1 specifies a DATA1 data packet type.

DATA2 specifies a DATA2 data packet type when on HIGH speed.

MDATA specifies a MDATA data packet type when on HIGH speed.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:TYPe DATA0 sets the data type as DATA0.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:TYPe? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:TYPe DATA0, indicating that the data type is DATA0.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:VALue

This command sets the binary data value to be used when triggering on a eUSB bus signal. The trigger condition must be set to DATA. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:VALue?

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:VALue XXXXXXX specifies the data value as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATa:VALue XXXXXXX, indicating the data value as XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATABITS:SIZe

This command sets the number of Sync bits, in bytes, to be used when triggering on a eUSB bus signal. The trigger condition must be set to SYNC. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATABITS:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATABITS:SIZe?
```

Arguments <NR1> specifies the number of the Sync bits. The default Sync bits is 12. The argument range is 12 to 32 bits.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATABITS:SIZe 12 specifies Sync bits as 12 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATABITS:SIZe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:DATABITS:SIZe 12, indicating that Sync bits value is 12 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDPoint:VALue

This command sets the endpoint binary value to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDPoint:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDPoint:VALue?
```

Arguments <QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; A-F, 0-9 and X for hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDPoint:VALue XXXX` specifies end point value as XXXX within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDPoint:VALue?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDPoint:VALue XXXX`, indicating that end point value as XXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ERRType

This command specifies the eUSB trigger type to be valid on a PID, Token CRC5, Data CRC16, or Bit stuffing condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ERRType
{BITSTUFFing|CRC5|CRC16|PID}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ERRType?
```

Arguments

`BITSTUFFing` specifies the error type as Bit Stuffing.

`CRC5` specifies the error type as Token CRC5 (Cyclic Redundancy Check 5).

`CRC16` specifies the error type as Data CRC16 (Cyclic Redundancy Check 16).

`PID` specifies the error type as PID Check Bits.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ERRType PID` specifies PID as the field within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ERRType?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ERRType PID`, indicating that PID is the currently specified field within a eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDSEARch

This command specifies eUSB Search EOP Bits to be On and Off. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDSEARch {OFF|ON}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDSEARch?
```

Arguments `OFF` specifies the EOP bits as OFF (for Low Speed and Full Speed).

`ON` specifies the EOP bits as ON (for High Speed).

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDSEARch OFF` specifies the EOP bits as OFF.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDSEARch?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:ENDSEARch OFF`, indicating the EOP bits as off.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe

This command sets the number of EOP bits, in bytes, to be used when triggering on a eUSB bus signal. The trigger condition must be set to EOP. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe?
```

Arguments `<NR1>` specifies the number of EOP bits. The default EOP bits is 8. The argument range is 8 to 100 bits.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe 8` specifies EOP bits as 8 within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe 8`, indicating that EOP bits value is 8 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:QUAlifier

This command sets the qualifier to be used when triggering on a eUSB bus signal. The trigger condition must be set to EOP Bits. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:QUAlifier {EQual| UNEQual| LESSthan| MOREthan| LESSEQual| MOREEQual| INrange| OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:QUAlifier?
```

Arguments

`EQual` specifies the qualifier as Equal.

`UNEQual` specifies the qualifier as Not Equal to.

`LESSthan` specifies the qualifier as Less Than.

`MOREthan` specifies the qualifier as More Than.

`LESSEQual` specifies the qualifier as Less Than or Equal to.

`MOREEQual` specifies the qualifier as More Than or Equal to.

`INrange` specifies the qualifier as Inside Range.

`OUTrange` specifies the qualifier as Out of Range.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:QUAlifier EQual` specifies the trigger value as Equal.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:QUAlifier?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOP:QUAlifier EQual`, indicating the trigger value is Equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe

This command sets the number of EOP bits to be used when triggering on a eUSB bus signal. The trigger condition must be set to EOP. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe?
```

Arguments <NR1> specifies the number of EOP bits. The default maximum EOP bits is 100. The argument range is 9 to 100 bits.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe 100 specifies maximum EOP bits as 100 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe 100, indicating that maximum EOP bits value is 100 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe

This command sets the number of EOP bits to be used when triggering on a eUSB bus signal. The trigger condition must be set to EOP. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe?
```

Arguments <NR1> specifies the number of EOP bits. The default minimum EOP bits is 8. The argument range is 8 to 99 bits.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe 8 specifies minimum EOP bits as 8 within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe 8, indicating that minimum EOP bits value is 8 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:HANDSHAKEType

This command specifies the eUSB trigger type to be valid on any, ack, nak, stall, or nyet condition. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:HANDSHAKEType
{ANY|ACK|NAK|STALL|NYET}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:HANDSHAKEType?
```

Arguments

ANY specifies the handshake type as Any (0010).

ACK specifies the handshake type as Acknowledgement (positive) (XX10).

NAK specifies the handshake type as Negative Acknowledgment (1010).

STALL specifies the error type as PID Check Bits.

NYET specifies the handshake type as Stall (endpoint is halted or control pipe request not supported) (1110).

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:HANDSHAKEType ANY` specifies any as the field within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:HANDSHAKEType?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:HANDSHAKEType ANY`, indicating that any is the currently specified field within a eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue

This command sets the binary Register Access Protocol address value to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue
<QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue?

Arguments <QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X for binary; for hexadecimal; and symbolic.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue XXXXXX specifies RAP address as XXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue XXXXXX, indicating that RAP address value XXXXXX is the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:COMMand

This command specifies the eUSB Register Access Protocol search option. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:COMMand
{ALL|WRITe|READ|CLEar|SET}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:COMMand?
```

Arguments ALL specifies the RAP command as All.

WRITe specifies the RAP command as Write.

READ specifies the RAP command as Read.

CLEar specifies the RAP command as Clear.

SET specifies the RAP command as Set.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:COMMand ALL specifies the command as the field within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:COMMand? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:COMMand ALL, indicating that command type is ALL.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:DATa:VALue

This command sets the binary Register Access Protocol data value to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:DATa:VALue?
```

Arguments <QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X for binary; for hexadecimal; and symbolic.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:DATa:VALue XXXXXXXX specifies RAP data value as XXXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:DATa:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:DATa:VALue XXXXXXXX, indicating that RAP data value XXXXXXXX is the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:OPTion

This command specifies the eUSB Register Access Protocol search option. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:OPTion
{COMMand|ADDRess|RAPDATa}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:OPTion?
```

Arguments COMMand specifies the RAP option as Command.

ADDRess specifies the RAP option as Address.

RAPDATa specifies the RAP option as RapData.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:OPTion COMMand` specifies the RAP option as the field within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:OPTion?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:RAP:OPTion COMMand`, indicating that RAP option is COMMand.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SOFFRAMENUMber

This command sets the binary frame number to be used when triggering on a eUSB bus signal. The trigger condition must be set to TOKEN. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SOFFRAMENUMber <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SOFFRAMENUMber?
```

Arguments `<QString>` specifies the binary frame number within the range 000 0000 0000 to 111 1111 1111 (000 hex to 7FF hex).

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SOFFRAMENUMber XXXXXXXXXXX` specifies frame number as XXXXXXXXXXX within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SOFFRAMENUMber?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SOFFRAMENUMber XXXXXXXXXXX`, indicating that XXXXXXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPECIALType

This command specifies the eUSB trigger type to be valid on any, err, split, ping, or reserved condition. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPECIALType {ANY|
PREamble| ERROr| SPLit| PING| RESERVed}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPECIALType?
```

Arguments

`ANY` specifies the PID value as Any (XX00).

`PREamble` specifies the PID value as PRE (1100).

`ERROr` specifies the PID value as ERR (1100).

`SPLit` specifies the PID value as Split (1000).

`PING` specifies the PID value as PING (0100).

`RESERVed` specifies the PID value as Reserved (0000).

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPECIALType ANY` specifies the special type as any.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPECIALType?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPECIALType ANY`, indicating the special type is any.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:ET:VALue

This command specifies the eUSB trigger type endpoint condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:ET:VALue {NOCARE|
CONTRol| ISOchronous| BULK| INTERRUPT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:ET:VALue?
```

Arguments

`NOCARE` specifies the Endpoint Type value as Don’t Care.

`CONTRol` specifies the Endpoint Type value as Control (00).

`ISOchronous` specifies the Endpoint Type value as Isochronous (01).

`BULK` specifies the Endpoint Type value as Bulk (10).

`INTERRUPT` specifies the Endpoint Type value as Interrupt (11).

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:ET:VALue CONTRol` sets the endpoint condition as Control.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:ET:VALue?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:ET:VALue CONTRol`, indicating the endpoint condition is Control.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue

This command sets the binary hub address value to be used when triggering on a eUSB bus signal. The trigger condition must be set to Special with packet type SPLIT. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue?
```

Arguments `<QString>` specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue XXXXXXX` specifies hub address value as XXXXXXX within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue XXXXXXX`, indicating that hub address value is XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue

This command sets the binary port address value to be used when triggering on a eUSB bus signal. The trigger condition must be set to Special with a packet type SPLIT. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue?
```

Arguments

<QString> specifies the value in the specified valid format. The valid characters are 0, 1, and X (for binary, for hexadecimal, and symbolic).

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue XXXXXXX specifies port address as XXXXXXX within an eUSB frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue XXXXXXX, indicating that port address value is XXXXXXX in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SC:VALue

This command specifies the eUSB trigger type start/complete condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SC:VALue {NOCARE|
SSPLIT| CSPLIT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SC:VALue?
```

Arguments

NOCARE sets the trigger on either the start or complete phase of the split transaction.

SSPLIT sets the trigger on the start phase of the split transaction.

CSPLIT sets the trigger on the complete phase of the split transaction.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SC:VALue NOCARE sets the start/complete conditon as nocare.

SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SC:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SC:VALue NOCARE, indicating that the start/complete condition is nocare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SE:VALue

This command specifies the eUSB trigger type start/complete condition. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SE:VALue {NOCARE|
FULLSPEED| LOWSPEED| ISOSTART| ISOMID| ISOEND| ISOALL}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SE:VALue?
```

Arguments

`NOCARE` sets the trigger on any combination of S and E bits.

`FULLSPEED` sets the trigger on S bit = 0 and E bit = 0.

`LOWSPEED` sets the trigger on S bit = 1 and E bit = 0.

`ISOSTART` sets the trigger on S bit = 1, E bit = 0.

`ISOMID` specifies Isochronous Data is Middle (00) Start/End bit value.

`ISOEND` specifies Isochronous Data is End (01) Start/End bit value.

`ISOALL` specifies Isochronous Data is All (11) Start/End bit value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SE:VALue FULLSPEED` sets the start/complete condition as fullspeed.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SE:VALue?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SPLit:SE:VALue FULLSPEED`, indicating that the start/complete condition is fullspeed.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNC:QUAlifier

This command sets the qualifier to be used when triggering on a eUSB bus signal. The trigger condition must be set to Sync Bits. The search number is specified by x.

Conditions Requires option SR-EUSB2

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNC:QUAlifier {EQual| UNEQual| LESSthan| MOREthan| LESSEQual| MOREEQual| INrange| OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNC:QUAlifier?
```

Arguments

`EQual` specifies the qualifier as Equal.

`UNEQual` specifies the qualifier as Not Equal to.

`LESSthan` specifies the qualifier as Less Than.

`MOREthan` specifies the qualifier as More Than.

`LESSEQual` specifies the qualifier as Less Than or Equal to.

`MOREEQual` specifies the qualifier as More Than or Equal to.

`INrange` specifies the qualifier as Inside Range.

`OUTrange` specifies the qualifier as Out of Range.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNC:QUAlifier EQual` specifies the trigger value as Equal.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNC:QUAlifier?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNC:QUAlifier EQual`, indicating the trigger value is Equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe

This command sets the number of sync bits, to be used when triggering on a eUSB bus signal. The trigger condition must be set to SYNC. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe?
```

Arguments

`<NR1>` specifies the number of sync bits. The default maximum Sync bits is 32. The argument range is 13 to 32 bits.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe 32` specifies minimum Sync bits as 32 within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe 32`, indicating that minimum Sync bits value is 32 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe

This command sets the number of sync bits, in bits, to be used when triggering on a eUSB bus signal. The trigger condition must be set to SYNC. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe?
```

Arguments

`<NR1>` specifies the number of sync bits. The default minimum Sync bits is 12. The argument range is 12 to 31 bits.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe 12` specifies minimum Sync bits as 12 within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe 12`, indicating that minimum Sync bits value is 12 in the currently specified field within an eUSB frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCSEARch

This command specifies eUSB Search Sync Bits to be On and Off. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCSEARch {OFF|ON}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCSEARch?
```

Arguments

`OFF` specifies the Sync bits as OFF (for Low Speed and Full Speed).

`ON` specifies the Sync bits as ON (for High Speed).

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCSEARch OFF` specifies the SYNC bits as off.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCSEARch?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:SYNCSEARch OFF`, indicating the SYNC bits is off.

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:TOKENType

This command specifies the eUSB trigger type to be valid on any, sof, out, in, or setup condition. The search number is specified by x.

Conditions

Requires option SR-EUSB2

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:TOKENType
{ANY|SOF|OUT|IN|SETUP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:TOKENType?
```

Arguments

`ANY` specifies the token type as Any (XX01).

`SOF` specifies the token type as Start Of Frame (0101).

`OUT` specifies the token type as OUT (0001).

`IN` specifies the token type as IN (1001).

`SETUP` specifies the token type as SETUP (1101).

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:TOKENType SETUP` specifies the token type Setup as the field within an eUSB frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:TOKENType?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:EUSB:TOKENType SETUP`, indicating that the token type is Setup.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CONDition

This command sets or queries the FlexRay bus search condition. The search number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:
FLEXRAY:CONDition {SOF|FRAMEType|
IDentifier|CYCLEcount|HEADer|DATA|IDANDDATA|EOF|ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CONDition?
```

Arguments Arguments are the available trigger conditions.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CONDition SOF` sets the trigger condition to start of frame.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CONDition?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:CONDITION SOF`, indicating the FlexRay trigger condition is start of frame.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue

This command sets or queries the heigh value when searching on a FlexRay bus cycle count field. The search number is specified by x. The search condition must be set to CYCLEcount.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue?
```

Arguments `<QString>` specifies the cycle count value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue "1010"` sets the cycle count value to XX1010.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:CYCLECOUNT:HIVALUE "XXXXXX"`, indicating the cycle count value is don't care, and it will trigger on any cycle count.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier

This command sets or queries the qualifier to be used when searching on a FlexRay bus search cycle count field. The search number is specified by x. The search condition must be set to CYCLEcount.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier
{EQUal|LESSEQual|MOREEQual|UNEQual|LESSthan|MOREthan|
INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier?
```

Arguments

Arguments are the available cycle count qualifiers.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier LESSthan` sets the cycle count qualifier to less than.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:CYCLECOUNT:QUALIFIER EQUAL`, indicating the cycle count qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue

This command sets or queries the low value when searching on a FlexRay bus cycle count field. The search number is specified by x. The search condition must be set to CYCLEcount.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue?
```

Arguments <QString> specifies the cycle count value.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue "1010"` sets the cycle count value to XX1010.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:CYCLECOUNT:VALUE "XXXXXX"`, indicating the cycle count value is don't care, and it will trigger on any cycle count.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue

This command sets or queries the high value when searching on a FlexRay bus data field. The search number is specified by x. The search condition must be set to IDANDDATA OR DATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue?
```

Arguments <QString> specifies the data value.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue "1010"` sets the data value to XXXXXX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:HIVALUE "XXXXXXXXXX", indicating the data value is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet

This command sets or queries the offset of the data string, in bytes, when searching on a FlexRay bus data field. The search number is specified by x. The search condition must be set to IDANDDATA OR DATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet <NR3>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet?
```

Arguments <NR3> specifies the data offset in bytes. A setting of X/Don't care is accomplished by setting the offset to –1.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet 2 sets the data offset to 2 bytes.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:OFFSET -1, indicating the data offset is don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier

This command sets or queries the qualifier to be used when searching on a FlexRay bus signal. The search number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier
{EQUal|LESSEQual|MOREEQual|UNEQual|LESSthan|MOREthan|
INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier?
```

Arguments Arguments are the available data qualifiers.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier LESSTHAN` sets the data qualifier to less than.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:QUALIFIER EQUAL`, indicating the data qualifier is equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to be used when searching on a FlexRay bus data field. The search number is specified by x. The search condition must be set to IDANDDATA OR DATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:SIZe?
```

Arguments `<NR1>` specifies the data size in bytes. A data size of –1 is don't care.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:SIZe 2` sets the data size to 2.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:SIZe?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:SIZE 1`, indicating the data size is 1 byte.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:VALue

This command sets or queries the low value to be used when searching on a FlexRay bus data field. The search number is specified by x. The search condition must be set to IDANDDATA or DATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:VALue?
```

Arguments <QString> specifies the data value.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:VALue "1010" sets the data value to XXXXXX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:DATa:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:DATA:VALUE "XXXXXXXXXX", indicating the data value is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:EOFTYPE

This command sets or queries the end of frame type when searching on a FlexRay bus signal. The search number is specified by x. The search condition must be set to EOF.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:EOFTYPE
{ANY|STATIC|DYNAMIC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:EOFTYPE?
```

Arguments ALL specifies either end of file type.

STATIC specifies the static end of file type.

DYNAMIC specifies the static end of file type.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:EOFTYPE STATIC sets the end of file type to static.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:EOFTYPE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:EOFTYPE ANY, indicating either end of file type will cause a trigger.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE

This command sets or queries the error type when searching on a FlexRay bus signal. The search number is specified by x. The search condition must be set to ERROR.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE
{CRCHeader|CRCTrailer|NULLFRStatic|NULLFRDynamic|
SYNCFrame|STARTupnosync}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE?
```

Arguments Arguments are the available error types.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:ERRTYPE SYNCFRAME sets the error type to SYNCFRAME.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:ERRTYPE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:ERRTYPE CRCHEADER, indicating the error type is CRCHEADER.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue

This command sets or queries the high value when searching on a FlexRay bus frame id field. The search condition must be set to IDentifier. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue?
```

Arguments

<QString> is the frame id high value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue "XXXXXXX1010" sets the HIVALUE to XXXXXXX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:FRAMEID:HIVALUE "XXXXXXXXXXX" indicating the HIVALUE is XXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:QUALifier

This command sets the qualifier to be used when searching on a FlexRay bus signal. The search condition must be set to Identifier. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:QUALifier
{EQual|UNEQual|LESSthan|MOREthan|LESSEQual|MOREEQual|
INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:QUALifier?
```

Arguments

Arguments are the available data qualifiers.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:QUALifier UNEQUAL sets the qualifier to unequal.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:QUALifier? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:FRAMEID:QUALIFIER EQUAL indicating the qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:VALue

This command sets the low value when searching on a FlexRay bus id field. The search condition must be set to IDentifier. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:VALue?
```

Arguments

<QString> is the frame id value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:VALue "XXXXXXX1010" sets the value to XXXXXXX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXray:FRAMEID:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:FRAMEID:VALUE "XXXXXXXXXXX" indicating the value is XXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEType

This command sets or queries the FlexRay bus search frame type. The search number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEType
{NORMal|PAYload|NULL|SYNC|STARTup}
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEType?
```

Arguments

Arguments are the available frame types.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:FRAMEType startup sets the frame type to startup.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:FRAMEType? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:FRAMETYPE NORMAL, indicating the frame type is normal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC

This command sets or queries the CRC portion of the binary header string whensearching on a FlexRay bus header. The search number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC?
```

Arguments <QString> specifies the CRC.

Examples SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC "1010" sets the header CRC to XXXXXX101.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:CRC "XXXXXXXXXX", indicating the header CRC is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount

This command sets or queries the cycle count portion of the binary header string when searching on a FlexRay bus header. The search number is specified by x. The search condition must be set to HEADer.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount?
```

Arguments <QString> specifies the header cycle count.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount "1010" sets the header cycle count to XX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:CYCLECOUNT "XXXXXX" indicating the cycle count has not been set.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID

This command sets or queries the frame id portion of the binary header string when searching on a FlexRay bus header. The search number is specified by x. The search condition must be set to HEADer or IDANDDATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID?
```

Arguments <QString> specifies the frame ID.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID "1010" sets the header frame ID to XXXXXXX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID? might return SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:FRAMEID "XXXXXXXXXXX", indicating the frame ID is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits

This command sets or queries the Indicator bits portion of the binary header string when searching on a FlexRay bus header. The search number is specified by x. The search condition must be set to HEADer.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits?
```

Arguments <QString> specifies the header Indicator Bits.

Examples `SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits "1010"` sets the header Indicator Bits to X1010.

`SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:INDBITS "XXXXX"`, indicating the Indicator Bits have not been set.

SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth

This command sets or queries the payload length portion of the binary header string when searching on a FlexRay bus search trigger header. The search number is specified by x. The search condition must be set to HEADer.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth?
```

Arguments `<QString>` specifies the header Payload Length.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth "1010"` sets the Payload Length to 1010.

`SEARCH:SEARCH1:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:FLEXRAY:HEADER:PAYLENGTH "XXXXXXX"`, indicating the Payload Length has not been set.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe

This command sets or queries the I2C address mode for the specified bus search to determine where to place a mark. The search number is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe
{ADDR10|ADDR7}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe?
```

Arguments

ADDR10 specifies the address mode as ADDR10.

ADDR7 specifies the address mode as ADDR7.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:MODE ADDR10 sets the address mode for I2C bus trigger search 1 to ADDR10.

SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:MODE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:MODE ADDR7, indicating that the address mode for I2C bus trigger search 2 is set to ADDR7.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:VALue

This command sets or queries the binary address string used for the I2C search the specified search condition is Address or AddressData. The search number is specified by x.

Conditions

Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe

Arguments

<QString> specifies the address value. This is either a 7-bit or 10-bit value depending on the address mode. The valid characters are 0-9, A-F, and X for addresses in hexadecimal format; and 0, 1, and X otherwise.

Examples

SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:ADDRESS:VALUE "01XXXXX" sets the address value to "01XXXXX" when the mode is ADDR7 and the format is binary.

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:ADDRESS:VALUE "XX", indicating

that the address value is "XX" when the address mode is set to ADDR7 and the address format is hexadecimal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition

This command sets or queries the search condition for an I2C bus. The search number is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition
{ADDRess|ADDRANDDATA|DATa|ACKMISS|REPEATstart|STARt|STOP}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition?
```

Arguments

`ADDress` specifies the trigger condition as Address.

`ADDRANDDATA` specifies the trigger condition as Address and Data.

`DATa` specifies the trigger condition as Data.

`ACKMISS` specifies the trigger condition as Missing of Acknowledgement.

`REPEATstart` specifies the trigger condition as Repeat of Start.

`STARt` specifies the trigger condition as Start.

`STOP` specifies the trigger condition as Stop.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:CONDITION ADDRESS` sets the trigger condition for I2C bus trigger search 1 to Address.

`SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:CONDITION?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:CONDITION ADDRANDDATA`, indicating that the trigger condition for I2C bus trigger search 1 is set to Address and Data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:DIRection

This command sets or queries the direction of the data for the I2C bus search to determine where to place a mark. The search number is specified by x. Read or write is indicated by the R/W bit in the I2C protocol.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:DIRection
{NOCARE|READ|WRITE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:DIRection?
```

Arguments `NOCARE` specifies the direction of data as Don't Care.

`READ` specifies the direction of data as Read.

`WRITE` specifies the direction of data as Write.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:DIRECTION READ` sets the data direction for I2C bus trigger search 1 to READ.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:DIRECTION?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:DIRECTION DONTCARE`, indicating that the data direction of the I2C bus trigger search 2 is DONTCare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:SIZe

This command sets or queries the length of the data string in bytes used for an I2C bus search to determine where to place a mark. The search number is specified by x. The search condition must be DATA or ADDRANDDATA.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:SIZe?
```

Arguments `<NR1>` specifies the data size in bytes.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:SIZE 1` sets the length of the data string for I2C bus trigger search 1 to 1 byte.

SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:SIZE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:SIZE 3, indicating that the length of the data string for I2C bus trigger search 1 is 3 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue

This command sets or queries the binary data string used for I2C bus search to determine where to place a mark. The search number is specified by x. The search condition must be DATA or ADDRANDDATA.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue?
```

Arguments <QString> specifies the data value. The valid characters are 0, 1, or X for binary format; and A-F, 0-9, and X for hexadecimal format.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:I2C:DATA:VALUE "1001" sets the data value for I2C bus trigger search 1 to "1001".

SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:VALUE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:I2C:DATA:VALUE "XX", indicating that the data value for I2C bus trigger search is "XX" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:MODe

This command sets or queries the address mode for the specified I3C bus trigger search to determine where to place a search mark.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:MODe
{ADDR10|ADDR7}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:MODe?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:VALue

Arguments `Search<x>` specifies the search number.

`ADDR10` specifies the address mode as ADDR10.

`ADDR7` specifies the address mode as ADDR7.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:ADDRESS:MODE ADDR10` sets the address mode for I3C bus trigger Search 1 to ADDR10.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:ADDRESS:MODE?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:ADDRESS:MODE ADDR7`, indicating that the address mode for I3C bus trigger Search 2 is set to ADDR7.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:VALue

This command sets or queries the I3C address value for which to search, to determine where to place a search mark.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:MODe

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the address value for the specified Search <x>. This is either a 7-bit or 10-bit value depending on the address mode. The valid characters are 0-9, A-F, and X for Hexadecimal address format; and 0, 1, and X (don't care) for Binary address format.

Examples `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:ADDRESS:VALUE "01XXXXX"` sets the address value to "01XXXXX" when the mode is ADDR7 and the format is binary.

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:ADDRESS:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:ADDRESS:VALUE "XX"`, indicating that the address value is "XX" when the address mode is set to ADDR7 and the address format is hexadecimal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCR:VALue

This command sets or queries the binary field string used for the I3C search if the trigger condition is BCR.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCR:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCR:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:BCR:VALue "1001"` sets the BCR value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BCR:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BCR:VALue "1001"`, indicating that the BCR value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCRType:VALue

This command sets or queries the binary field string used for the I3C bus search if the trigger condition is BCRTYPE.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCRType:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCRType:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the data value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:BCRType:VALue "1001"` sets the DCR Type value for I3C bus Search 1 to "1001".

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BCRType:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BCRType:VALue "1001", indicating that the BCR Type value for I3C bus Search 2 is "1001" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BPACKets

This command sets or queries the I3C Broadcast Packets search to Broadcast or Response.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BPACKets
{RESPonse|BROadcast}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BPACKets?
```

Arguments

Search<x> specifies the search number.

RESPonse specifies the broadcast address mode as Response.

BROadcast specifies the broadcast address mode as Broadcast.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:BPACKets RESPonse sets the broadcast packets type for I3C bus Search 1 to Response.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BPACKets? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BPACKets RESPonse, indicating that the broadcast packets type for I3C bus Search 2 is set to Response.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BRGTID:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is SETBRGTGT packet ID.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BRGTID:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BRGTID:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:BRGTID:VALue "1001"` sets the Brgtgt search value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BRGTID:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:BRGTID:VALue "9A"`, indicating that the Brgtgt search value for I3C bus trigger Search 2 is 9A in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CCODe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is SETBRGTGT packet ID.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CCODe:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CCODe:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and A-F, 0-9, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:CCODe:VALue "1001"` sets the command code value for I3C bus Search 1 to Binary 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:CCODe:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:CCODe:VALue "1001"`, indicating that the command code value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CONDition

This command sets or queries the trigger condition for the specified I3C bus trigger search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CONDition
{STARt|REPEATStart|ADDRess|DATa|
SDRDirect|SDRBroadcast|DDR|ERRors|HOTJoin|
DMSGEnd|BMSGEnd|HDRRestart|HDRExit}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CONDition?
```

Arguments

`Search<x>` specifies the search number.

`STARt` specifies trigger condition as Start.

`REPEATStart` specifies trigger condition as Repeated Start.

`ADDRess` specifies trigger condition as Address.

`DATa` specifies trigger condition as Data.

`SDRDirect` specifies trigger condition as I3C SDR Direct.

`SDRBroadcast` specifies trigger condition as I3C SDR Broadcast.

`DDR` specifies trigger condition as DDR.

`ERRors` specifies trigger condition as Errors.

`HOTJoin` specifies trigger condition as Hot-Join.

`DMSGEnd` specifies trigger condition as Direct Message End.

`BMSGEnd` specifies trigger condition as Stop.

`HDRRestrat` specifies trigger condition as HDR Restart.

`HDRExit` specifies trigger condition as HDR Exit.

Examples

`SEARCH:SEARCH3:TRIGGER:A:BUS:I3C:CONDITION START` sets the trigger condition for I3C bus trigger Search 3to Start.

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:CONDITION?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:CONDITION ADDRESS`, indicating that the trigger condition for I3C bus trigger search 1 is set to Address.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:DIRection

This command sets or queries the direction of the data for the I3C bus trigger search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:DIRection
{NOCARE|READ|WRITE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:DIRection?
```

Arguments `Search<x>` specifies the search number.

`NOCARE` specifies the data direction for which to search is Don't Care.

`READ` specifies the data direction for which to search is Read.

`WRITE` specifies the data direction for which to search is Write.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DATA:DIRECTION READ` sets the data direction for I3C bus Search 1 to READ.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DATA:DIRECTION?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DATA:DIRECTION NOCARE`, indicating that the data direction of the I3C bus Search 2 is Don't Care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:SIZe

This command sets or queries the length of the data string in bytes used for the specified I3C bus trigger search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:SIZe?
```

Arguments `Search<x>` specifies the search number.

`<NR1>` specifies the data size in bytes.

Examples `SEARCH:SEARCH4:TRIGGER:A:BUS:I3C:DATA:SIZE 1` sets the length of the data string for I3C bus Search 4 to 1 byte.

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DATA:SIZE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DATA:SIZE 3`, indicating that the length of the data string for I3C bus Search 1 is 3 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:VALue

This command sets or queries the data value of the data token for the specified I3C bus trigger search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:VALue?
```

Arguments

`Search<x>` specifies the search number.

`<Qstring>` specifies the data value. The valid characters are 0, 1, or X for Binary format; and A-F, 0-9, and X for Hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DATA:VALUE "1001"` sets the data value for I3C bus trigger Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DATA:VALUE?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DATA:VALUE "XX"`, indicating that the data value for I3C bus Search 2 is "XX" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCR:VALue

This command sets or queries the binary field string used for the I3C bus search if the trigger condition is DCR.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCR:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCR:VALue?
```

Arguments

`Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DCR:VALue "1001"` sets the DCR value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DCR:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DCR:VALue "1001", indicating that the DCR value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCRType:VALue

This command sets or queries the binary field string used for the I3C bus search if the trigger condition is DCRTYPE.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCRType:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCRType:VALue?
```

Arguments

Search<x> specifies the search number.

<Qstring> specifies the data value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DCRType:VALue "1001" sets the DCR Type value for I3C bus Search 1 to "1001".

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DCRType:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DCRType:VALue "1001", indicating that the DCR Type value for I3C bus Search 2 is "1001" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DWORd:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is DATAWORDS.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DWORd:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DWORd:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:DWORd:VALue "1001"` sets the data word value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DWORd:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:DWORd:VALue "1001"`, indicating that the **event byte** value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ERRORTYPe

This command sets or queries the error type condition for the specified I3C bus search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ERRORTYPe
{NACK|TBIT|PARity| BADDr|DADDr|PREamble|CRC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ERRORTYPe?
```

Arguments `Search<x>` specifies the search number.

`NACK` specifies the error type on which to search as Missing Ack.

`TBIT` specifies the error type on which to search as T-Bit.

`PARity` specifies the error type on which to search as Parity.

`BADDr` specifies the error type on which to search as Broadcast Address.

`DADDr` specifies the error type on which to search as Dynamic Address.

`PREamble` specifies the error type on which to search as Preamble.

`CRC` specifies the error type on which to search as CRC.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:ERRORTYPe NACK` sets the error condition for I3C bus trigger Search 1 to Missing ack.

`SEARCH:SEARCH3:TRIGGER:A:BUS:I3C:ERRORTYPe?` might return `SEARCH:SEARCH3:TRIGGER:A:BUS:I3C:ERRORTYPe PARITY`, indicating that the error condition for I3C bus trigger Search 3 is set to Parity.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:EVENTBYTe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is EVENTBYTE.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:EVENTBYTe:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:EVENTBYTe:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:EVENTBYTe:VALue "1001"` sets the event byte value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:EVENTBYTe:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:EVENTBYTe:VALue "1001"`, indicating that the event byte value for I3C bus trigger Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:FREQBYTe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is FREQUENCYBYTE.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:FREQBYTe:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:FREQBYTe:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:FREQBYTe:VALue "1001"` sets the frequency byte value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:FREQBYTe:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:FREQBYTe:VALue "1001"`, indicating that the frequency byte value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSLSb:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is GETSTATUSLSB.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSLSb:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSLSb:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:GSLSb:VALue "1001"` sets the GetStatus MSb value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:GSLSb:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:GSLSb:VALue "1001"`, indicating that the GetStatus MSb value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSMSb:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is GETSTATUSMSB.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSMSb:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSMSb:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:GSMSb:VALue "1001"` sets theGetStatus MSb value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:GSMSb:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:GSMSb:VALue "1001"`, indicating that the GetStatus MSb value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ID:VALue

This command sets or queries the binary field string used for the I3C search if the trigger condition is ENTDAA packet ID.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ID:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ID:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:ID:VALue "1001"` sets the ID value for I3C bus trigger Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:ID:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:ID:VALue "1001"`, indicating that the ID value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:INACCBYTe:VALue

This command sets or queries the binary field string used for the I3C search if the trigger condition is INACCURACYBYTE.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:INACCBYTe:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:INACCBYTe:VALue?
```

Arguments

`Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:INACCBYTe:VALue "1001"` sets the inaccuracy byte value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:INACCBYTe:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:INACCBYTe:VALue "1001"`, indicating that the inaccuracy byte value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXREAD:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is MAXREAD.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXREAD:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXREAD:VALue?
```

Arguments

`Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:MAXREAD:VALue "1001"` sets the Maximum Read value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH4:TRIGGER:A:BUS:I3C:MAXREAD:VALue?` might return `SEARCH:SEARCH4:TRIGGER:A:BUS:I3C:MAXREAD:VALue "1001"`, indicating that the Maximum Read value for I3C bus Search 4 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXRTURN:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is MAXREADTURN.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXRTURN:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXRTURN:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH5:TRIGGER:A:BUS:I3C:MAXRTURN:VALue "1001"` sets the Maximum Read Turnaround value for I3C bus Search 5 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:MAXRTURN:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:MAXRTURN:VALue "1001"`, indicating that the Maximum Read Turnaround value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXWRITe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is MAXWRITE.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXWRITe:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXWRITe:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:MAXWRITe:VALue "1001"` sets the Maximum Write value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:MAXWRITe:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:MAXWRITe:VALue "1001"`, indicating that the Maximum Write value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:PACKets

This command sets or queries the I3C Direct Packets to Direct or Response.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:PACKets {RESPonse|DIRect}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:PACKets?
```

Arguments `Search<x>` specifies the search number.

`RESPonse` specifies the packet address mode as Response.

`DIRect` specifies the packet address mode as Direct.

Examples `SEARCH:SEARCH3:TRIGGER:A:BUS:I3C:PACKets DIRect` sets the packets for I3C bus Search 3 to Direct.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:PACKets?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:PACKets RESPonse`, indicating that the packets for I3C bus Search 2 is set to Response.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SADDress:VALue

This command sets or queries the binary field string used for the I3C bus searchif the trigger condition is SLAVEADDRESS.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SADDress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SADDress:VALue?
```

Arguments

`Search<x>` specifies the search number.

`<Qstring>` specifies the address value for which to search. The valid characters are 0, 1, or X for Binary format; and A-F, 0-9, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:SADDress:VALue "1001" sets the slave address value for I3C bus trigger search 1 to "1001".

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:SADDress:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:SADDress:VALue "1001"`, indicating that the slave address value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:BROADCASTPacket

This command sets or queries the SDR broadcast packets for the specified I3C bus trigger search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:
BROADCASTPacket {ENSLave|DISLave|ENTasx|RSTDya|
ENTRDya|SETMwrl|SETMrdl|LSLave|ENTRTSTMode|EXTime>}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:BROADCASTPacket?
```

Arguments

`Search<x>` specifies the search number.

`ENSLave` specifies the I3C broadcast packet type as Enable Slave.

`DISLave` specifies the I3C broadcast packet type as Disable Slave.

`ENTasx` specifies the I3C broadcast packet type as Enter Activity Stat.

`RSTDya` specifies the I3C broadcast packet type as Reset Dynamic Address.

`ENTRDya` specifies the I3C broadcast packet type as Reset Dynamic Address.

`SETMwrl` specifies the I3C broadcast packet type as Set Max Write Length.

`SETMrdl` specifies the I3C broadcast packet type as Set Max Read Length.

`LSLave` specifies the I3C broadcast packet type as Define List of Slaves.

`ENTRTSTMode` specifies the I3C broadcast packet type as Enter Test Mode Broadcast.

`EXTime` specifies the I3C broadcast packet type as Set Exchange Time.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:SDR:BROADCASTPacket ENSLave` sets the trigger condition for I3C bus trigger search 1 to Enable Slave.

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:SDR:BROADCASTPacket?` might return `SEARCH:SEARCH5:TRIGGER:A:BUS:I3C:SDR:BROADCASTPacket ENTRDYA`, indicating that the trigger condition for I3C bus trigger Search 5 is set to Reset Dynamic Address.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:DIRECTPacket

This command sets or queries the SDR direct packet types for the specified I3C bus trigger search to determine where to place a mark.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:
DIRECTPacket {ENSLave|DISLave|ENTasx|RSTDya|
SETMwrl|SETMrdl|SEText|SETDya|SETNdya|GETMWrl|GETMRdl|
GETPrid|GETBusch|GETDevch|GETSlave|ACCM|SETBrt|
MDATASpeed|HDRCapability}
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:DIRECTPacket?
```

Arguments

`Search<x>` specifies the search number.

`ENSLave` specifies the trigger condition as Enable Slave.

`DISLave` specifies the trigger condition as Disable Slave.

`ENTasx` specifies the trigger condition as Enter Activity State.

`RSTDya` specifies the trigger condition as Reset Dynamic Address.

`SETMwrl` specifies the trigger condition as Set Max Write Length.

`SETMrdl` specifies the trigger condition as Set Max Read Length.

`SEText` specifies the trigger condition as Set Exchange Time.

`SETDya` specifies the trigger condition as Set Dynamic Address.

`SETNdya` specifies the trigger condition as Set New Dynamic Address.

`GETMWrl` specifies the trigger condition as Get Max Write Length.

`GETMRdl` specifies the trigger condition as Get Max Read Length.

`GETPrid` specifies the trigger condition as Get Provisional ID.

`GETBusch` specifies the trigger condition as Get Provisional ID.

GETDevch specifies the trigger condition as Get Dev Characteristics.

GETSlave specifies the trigger condition as Get Slave Current Status.

ACCM specifies the trigger condition as Get Accept Mastership.

SETBrt specifies the trigger condition as Set Bridge Direct Target.

MDATASpeed specifies the trigger condition as Get Max Data Speed.

HDRCapability specifies the trigger condition as Get HDR Capability.

Examples

SEARCH:SEARCH4:TRIGGER:A:BUS:I3C:SDR:DIRECTPacket ENSLave sets the trigger condition for I3C bus trigger Search 4 to Enable Slave.

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:SDR:DIRECTPacket? might return SEARCH:SEARCH8:TRIGGER:A:BUS:I3C:SDR:DIRECTPacket ENTASX, indicating that the trigger condition for I3C bus Search 8 is set to Enter Activity State.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATEBYTe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is STATEBYTE.

Requires purchase and installation of option SRI3C.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATEBYTe:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATEBYTe:VALue?
```

Arguments

Search<x> specifies the search number.

<Qstring> specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:STATEBYTe:VALue "1001" sets the state byte value for I3C bus Search 1 to 1001.

SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:STATEBYTe:VALue? might return SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:STATEBYTe:VALue "1001", indicating that the state byte value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATic:ADDRess

This command sets or queries the binary field string used for the I3C search if the trigger condition is STATICADDRESS.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATic:ADDRess <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATic:ADDRess?
```

Arguments

`Search<x>` specifies the search number.

`<Qstring>` specifies the address value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:STATic:ADDRess "1001"` sets the address value for I3C bus trigger Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:STATic:ADDRess?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:STATic:ADDRess "1001"`, indicating that the address value for I3C bus trigger Search 2 is "1001" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SUPPORTBYTe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is SUPPORTEDBYTE.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SUPPORTBYTe:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SUPPORTBYTe:VALue?
```

Arguments

`Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:SUPPORTBYTe:VALue "1001"` sets the supported byte value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:SUPPORTBYTe:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:SUPPORTBYTe:VALue "1001"`, indicating that the supported byte value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:TESTMODe:VALue

This command sets or queries the binary field string used for the I3C trigger if the trigger condition is TESTMODE.

Requires purchase and installation of option SRI3C.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:TESTMODe:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:TESTMODe:VALue?
```

Arguments `Search<x>` specifies the search number.

`<Qstring>` specifies the value for which to search. The valid characters are 0, 1, or X for Binary format; and 0-9, A-F, and X for Hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:I3C:TESTMODe:VALue "1001"` sets the test mode value for I3C bus Search 1 to 1001.

`SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:TESTMODe:VALue?` might return `SEARCH:SEARCH2:TRIGGER:A:BUS:I3C:TESTMODe:VALue "1001"`, indicating that the test mode value for I3C bus Search 2 is 1001 in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition

This command sets or queries the condition for a LIN bus search. The search number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition
{DATA|IDANDDATA|ERRor|IDentifier|SLEEP|SYNCfield|WAKEup}
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition?
```

Arguments

Arguments are the available trigger conditions.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:LIN:CONDition DATA` sets the trigger condition to data.

`SEARCH:SEARCH1:TRIGger:A:BUS:LIN:CONDition?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:CONDITION SYNC`, indicating the trigger condition is sync.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue

This command sets or queries the high data value string used in a LIN bus search. The search number is specified by x. The search condition must be DATA or IDANDDATA and the data qualifier must be INRANGE or OUTRANGE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue?
```

Arguments

`<QString>` is a quoted string of 1s, 0s, or Xs representing the binary data string to be used in a LIN search if the search condition is `IDentifier` or `IDANDDATA` (identifier and data).

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:HIVALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:HIVALUE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"` indicating the high value is "don't care".

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:QUALifier

This command sets or queries the data qualifier used in a LIN bus search. The search number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:QUALifier
{EQual|LESSEQual|MOREEQual|UNEQual|LESSthan|MOREthan|
INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:QUALifier?
```

Arguments Arguments are the available data qualifiers.

Examples `SEARCH:SEARCH2:TRIGger:A:BUS:LIN:DATa:QUALifier LESS` sets the data qualifier to less than.

`SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATa:QUALifier?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:QUALIFIER EQUAL`, indicating the data qualifier is equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZe

This command sets or queries the length of the stat string in bytes used for a LIN bus search. The search number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZe?
```

Arguments `<NR1>` specifies the data size.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATa:SIZe 1.0` sets the data size to 1.

`SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATa:SIZe?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:SIZE 1`, indicating the data size is 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue

This command sets or queries the data string used for a LIN bus search. The search number is specified by x. The search condition must be DATA or IDANDDATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue?
```

Arguments <QString> specifies the data value.

Examples SEARCH:SEARCH2:TRIGger:A:BUS:LIN:DATa:VALue 1010 sets the data value to XXXXXX1010.

SEARCH:SEARCH1:TRIGger:A:BUS:LIN:DATa:VALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:DATA:VALUE "XXXXXXXXXX", indicating that the data value is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE

This command sets or queries the error type for a LIN bus search. The search number is specified by x. The search condition must be set to ERROR.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE
{CHecksum|PARity|SYNC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE?
```

Arguments CHecksum specifies the error type is checksum.

PARity specifies the error type is parity.

SYNC specifies the error type is sync.

Examples `SEARCH:SEARCH2:TRIGger:A:BUS:LIN:ERRTYPE Parity` sets the error type to parity.

`SEARCH:SEARCH2:TRIGger:A:BUS:LIN:ERRTYPE?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:LIN:ERRTYPE SYNC`, indicating that the error type is sync.

SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue

This command sets or queries the string used for a LIN bus identifier value. The search number is specified by x. The search condition must be IDENTIFIER or IDANDDATA.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue?
```

Arguments `<QString>` specifies the identifier value.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:LIN:IDentifier:VALue 1010` sets the identifier value to XX1010.

`SEARCH:SEARCH1:TRIGger:A:BUS:LIN:IDentifier:VALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:LIN:IDENTIFIER:VALUE "XXXXXX"`, indicates the identifier value is don't care.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition

This command specifies a field or condition for an Manchester bus to search on. The search number is specified by x.

Conditions Requires the SR-MANCH Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition
{SYNCBITS| HEADer| DATa| packetOffData| TRAiler| ERRors}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition?
```

Related Commands

BUS:B<x>:MANChester:DISplaymode

Arguments

`SYNCBITS` specifies Sync Bits symbols as the field within an Manchester frame to search on.

`HEADer` specifies header symbols as the field within an Manchester frame to search on.

`DATa` specifies data symbols as the field within an Manchester frame to search on.

`packetOffData` specifies the binary string used for Manchester triggering if the trigger condition is DATA.

`TRAiler` specifies trailer symbols as the field within an Manchester frame to search on.

`ERRors` specifies error symbols as the field within an Manchester frame to search on.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MANChester:CONDition SYNCBITS` specifies Sync Bit symbols as the field within an Manchester frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MANChester:CONDition?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:MANChester:CONDition SYNCBITS`, indicating that Sync Bits is the currently specified field within an Manchester frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:SIZe

This command specifies the number of bits per word for the specified Manchester bus. The search number is specified by x.

Conditions

Requires the SR-MANCH Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:SIZe?
```

Related Commands BUS:B<x>:MANChester:DISplaymode, SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition

Arguments `Search<x>` is the number of the search.

`<NR1>` specifies the data string in a valid format. The minimum and default values are 1. The maximum value is 5.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:SIZe 2` sets the data string for Manchester bus trigger search 1 to 2 bits.

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:SIZe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:SIZe 2`, indicating that the length of the data string for Manchester bus trigger search 1 is 2 bits.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:VALue

This command specifies the binary string used for Manchester triggering if the trigger condition is data when searching on an Manchester signal. The search condition needs to be set to data. The search number is specified by <x>.

Conditions Requires the SR-MANCH Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:VALue?
```

Related Commands BUS:B<x>:MANChester:DISplaymode, SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition

Arguments `<QString>` is the data value.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue "00001000"` specifies binary as the value to be used while searching.

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue "00001000"`, indicating that binary is the value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:ERRor:TYPe

This command sets or queries the error to be used when searching on a Manchester signal. The search condition must be set to error. Parity error identifies odd and even parity. The search number is specified by <x>.

Conditions Requires the SR-MANCH Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:ERRor:TYPe
{MANChester|PARity}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:ERRor:TYPe?
```

Related Commands BUS:B<x>:MANChester:DISplaymode, SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition

Arguments `MANChester` sets the error type to Manchester.

`PARity` sets the error type to parity.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:ERRor:TYPe MANChester` sets the error type to Manchester.

`SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:ERRor:TYPe?` might return `:SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:ERRor:TYPe PARity`, indicating the error type is parity.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:HEADER:VALue

This command specifies the binary string used for Manchester triggering if the trigger condition is header when searching on an Manchester signal. The search condition needs to be set to header. The search number is specified by <x>.

Conditions Requires the SR-MANCH Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:HEADER:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:HEADER:VALue?
```

Related Commands BUS:B<x>:MANChester:DISplaymode, SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition

Arguments <QString> is the header value.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue "00001000" specifies binary as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:DATa:VALue "00001000", indicating the binary header value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:packetOffData:VALue

This command specifies the binary string used for Manchester triggering if the trigger condition is DATA. The search number is specified by <x>.

Conditions Requires the SR-MANCH Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:packetOffData:
VALue <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:packetOffData:
VALue?
```

Related Commands BUS:B<x>:MANChester:DISplaymode, SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition

Arguments <NR1> specifies the data string in a valid format.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue 1 specifies binary as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue 1, indicating the binary value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:SYNC:VALue

This command specifies the binary string used for Manchester triggering if the trigger condition is Sync when searching on an Manchester signal. The search condition needs to be set to Sync. The search number is specified by <x>.

Conditions Requires the SR-MANCH Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:SYNC:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:SYNC:VALue?
```

Related Commands BUS:B<x>:MANChester:DISplaymode, SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition

Arguments <QSTRING> specifies the data string in a valid format.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue "00001000" specifies binary as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue? might return SEARCH:SEARCH1:TRIGger:A:BUS:MANChesterDATAOFF:VALue "00001000", indicating the binary value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:TRAILER:VALue

This command specifies the binary string used for Manchester triggering if the trigger condition is Trailer when searching on an Manchester signal. The search condition needs to be set to Trailer. The search number is specified by <x>.

Conditions Requires the SR-MANCH Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:TRAILER:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:TRAILER:VALue?
```

Related Commands BUS:B<x>:MANChester:DISplaymode, SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition

Arguments <QString> specifies the data string in a valid format.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:TRAILER:VALue "00001000" specifies binary as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:TRAILER:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:MANChester:TRAILER:VALue "00001000", indicating the binary value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ADDress:VALue

This command specifies the binary address string used for MDIO triggering if the trigger condition is address. The search number is specified by x.

Conditions Requires option SR-MDIO.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ADDress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ADDress:VALue?
```

Arguments <QString> is the address value. Default address value is XXXXXXXXXXXXXXXX.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ADDress:VALue "XXXXXXXXXXXXXXXX" specifies address value as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ADDress:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ADDress:VALue "XXXXXXXXXXXXXXXX", indicating that data is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:CONDition

This command sets or queries the trigger condition for MDIO bus. The search number is specified by x.

Conditions Requires option SR-MDIO.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:CONDition {STARTPACKet|
PHYSICALADDRess| REGISTERADDRess| DEVICETYPe| DATA| ADDRess|
ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:CONDition?
```

Arguments

`STARTPACKet` specifies STARTPACKet as the field within a MDIO frame to search on. This is the default trigger on condition.

`PHYSICALADDRess` specifies PHYSICALADDRess as the field within a MDIO frame to search on.

`REGISTERADDRess` specifies REGISTERADDRess as the field within a MDIO frame to search on.

`DEVICETYPe` specifies DEVICETYPe as the field within a MDIO frame to search on.

`DATA` specifies DATA as the field within a MDIO frame to search on.

`ADDRess` specifies ADDRess as the field within a MDIO frame to search on.

`ERRor` specifies ERRor as the field within a MDIO frame to search on.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:CONDition ERRor` specifies ERRor as the field within a MDIO frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:CONDition?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:CONDition ERRor`, indicating that ERRor is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DATa:VALue

This command specifies the binary data string used for MDIO triggering if the trigger condition is DATA. The search number is specified by x.

Conditions Requires option SR-MDIO.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DATa:VALue?
```

Arguments <QString> is the data value. Default value is XXXXXXXXXXXXXXXX.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DATa:VALue "XXXXXXXXXXXXXXXX" specifies data as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DATa:VALue "XXXXXXXXXXXXXXXX", indicating that data is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DEVicetype:VALue

This command specifies the binary device type string used for MDIO triggering if the trigger condition is Device type. The search number is specified by x.

Conditions Requires option SR-MDIO.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DEVicetype:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DEVicetype:VALue?
```

Arguments <QString> is the device type. Default value is XXXXX.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DEVicetype:VALue "XXXXX" specifies device type as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DEVicetype:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:DEVicetype:VALue "XXXXX", indicating that device type is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ERRType

This command sets or queries the error type to be used when triggering on MDIO data. This applies when the MDIO bus trigger condition is set to ERRor. The search number is specified by x.

Conditions Requires option SR-MDIO.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ERRType {OPCODEERRor|
DEVERRor| ANY}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ERRType?
```

Arguments `OPCODEERRor` specifies OPCODEERRor as the field within a MDIO frame to search on. This is the default trigger on condition.

`DEVERRor` specifies DEVERRor as the field within a MDIO frame to search on.

`ANY` specifies ANY as the field within a MDIO frame to search on.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ERRType ANY` specifies error type as the field within a MDIO frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ERRType?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:ERRType ANY`, indicating that error type is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:OPCode:VALue

This command specifies the binary op-code string used for MDIO triggering if the trigger condition is op-code. The search number is specified by x.

Conditions Requires option SR-MDIO.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:OPCode:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:OPCode:VALue?
```

Arguments `<QString>` is the op-code value. Default op-code value is XX.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:OPCode:VALue "XX"` specifies op-code value as the field within a MDIO frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:OPCode:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:OPCode:VALue "XX"`, indicating that op-code value is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:PHYSicaladdress:VALue

This command specifies the binary physical address string used for MDIO triggering if the trigger condition is physical address. The search number is specified by x.

Conditions Requires option SR-MDIO.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:PHYSicaladdress:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:PHYSicaladdress:VALue?
```

Arguments <QString> is the physical address value. Default physical address value is XXXXX.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:PHYSicaladdress:VALue "XXXXX" specifies physical address as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:PHYSicaladdress:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:PHYSicaladdress:VALue "XXXXX", indicating that physical address is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:REGisteraddress:VALue

This command specifies the binary register address string used for MDIO triggering if the trigger condition is register address. The search number is specified by x.

Conditions Requires option SR-MDIO.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:REGisteraddress:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:REGisteraddress:VALue?
```

Arguments <QString> is the register address value. Default register address value is XXXXX.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:REGisteraddress:VALue "XXXXX" specifies register address as the field within a MDIO frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:REGisteraddress:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:MDIO:REGisteraddress:VALue "XXXXX", indicating that register address is the currently specified field within a MDIO frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:HIVALue

This command sets or queries the high value when searching on command word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to COMMAND, and the address qualifier must be INrange or OUTrange.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:
HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:
HIVALue?
```

Arguments <QString> is the address value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:HIVALUE "X1000" sets the value to X1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:HIVALUE? might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:QUALifier

This command sets or queries the qualifier to be used when searching on command word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to COMMAND.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:
ADDRess:QUALifier {EQual|UNEQual|LESSthan
|MOREthan|LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:
QUALifier?
```

Arguments Arguments are the available address qualifiers.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:QUALIFIER LESSTHAN` sets the address qualifier to less than.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:QUALIFIER?` might return EQUAL, indicating that the address qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:VALue

This command sets or queries the low value when searching on command word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to COMMAND.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:
VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:
VALue?
```

Arguments `<QString>` is the address value.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:VALUE "X1000"` sets the value to X1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:ADDRESS:VALUE? might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:COUNt

This command sets or queries the value of the command word "word count" field for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:COUNt
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:COUNt?
```

Arguments <QString> is the word count value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:COUNT "X1000" sets the value to X1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:COUNT? might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:PARity
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:PARity?
```

Arguments

ONE filters command words to only match those where the parity bit has a value of 1.

ZERO filters command words to only match those where the parity bit has a value of 0.

NOCARE disables filtering of command words on the parity bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:PARITY ONE specifies filtering command words for those where the parity bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:PARITY? might return NOCARE, indicating that command words are not being filtered based on the parity bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:SUBADdress

This command sets or queries the value of the command word subaddress field for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:SUBADdress
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:SUBADdress?
```

Arguments

<QString> is the word count value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:SUBADDRESS "X1000" sets the value to X1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:SUBADDRESS? might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit

This command sets or queries the value of the command word Transmit / Receive bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to COMMAND.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit
{RX|TX|X}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit?
```

Arguments RX filters command words to only match those that are receive packets.

TX filters command words to only match those that are transmit packets.

X disables filtering of command words on the R/T bit.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:TRBIT TX specifies filtering command words for only transmit messages.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:COMMAND:TRBIT? might return X, indicating that command words are not being filtered based on the R/T bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition

This command sets or queries the field or condition for a MIL-STD-1553 bus to search on. The search number is specified by x.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition
{SYNC|COMMAND|STATus|DATA|ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition?
```

Arguments SYNC specifies a search for the sync pulse of any word.

COMMAND specifies a search for a matching command word.

STATus specifies a search for a matching status word.

DATA specifies a search for a matching data word.

ERRor specifies a search for a specified error condition.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:CONDITION DATA specifies finding matching data word(s).

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:CONDITION? might return SYNC, indicating that the bus is being searched for sync pulses found in any word.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to DATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:PARity
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:PARity?
```

Arguments

ONE filters data words to only match those where the parity bit has a value of 1.

ZERo filters data words to only match those where the parity bit has a value of 0.

NOCARE disables filtering of data words on the parity bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:PARITY ONE specifies filtering data words for those where the parity bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:PARITY? might return NOCARE, indicating that data words are not being filtered based on the parity bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue

This command sets or queries the value when searching on data words for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to DATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue?
```

Arguments <QString> is the data value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:VALUE "XXXXXXXXXXX1000" sets the value to XXXXXXXXXXX1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:DATA:VALUE? might return "XXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe

This command sets or queries the type of error condition for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to ERRor.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe
{PARity|SYNC|DATA}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe?
```

Arguments PARity specifies searching for an incorrectly calculated parity bit in any word.

SYNC specifies searching for any sync pulse that does not transition in the middle of the pulse as required.

DATA specifies searching for any non-contiguous data words.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:ERRTYPE DATA specifies searching for non-contiguous data words.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:ERRTYPE? might return PARITY, indicating that the bus is being searched for parity errors in any word.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:HIVALue

This command sets or queries the high value when searching on status word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to STATus and the address qualifier must be INrange or OUTrange.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:
HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:
HIVALue?
```

Arguments <QString> is the address value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:HIVALUE "X1000" sets the value to X1000.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:HIVALUE? might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:QUALifier

This command sets or queries the qualifier used when searching on status word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:
MIL1553B:STATus:ADDRess:QUALifier
{EQual|UNEQual|LESSthan|MOREthan|LESSEQual|MOREEQual|
INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:
QUALifier?
```

Arguments

Arguments are the available address qualifiers.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:QUALIFIER LESSTHAN` sets the address qualifier to less than.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:QUALIFIER?` might return EQUAL, indicating that the address qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:VALue

This command sets or queries the low value when searching on status word addresses for a MIL-STD-1553 bus. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:
VALue?
```

Arguments

`<QString>` is the address value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:VALUE "X1000"` sets the value to X1000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:ADDRESS:VALUE?` might return "XXXXX", indicating that the value is XXXXX.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BCR

This command sets or queries the value of the broadcast command received bit (BCR bit, bit 15) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BCR
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BCR?
```

Arguments `ONE` filters status words to only match those where the BCR bit has a value of 1.

`ZERo` filters status words to only match those where the BCR bit has a value of 0.

`NOCARE` disables filtering of status words on the BCR bit.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:BCR ONE` specifies filtering status words for those where the BCR bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:BCR?` might return NOCARE, indicating that status words are not being filtered based on the BCR bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BUSY

This command sets or queries the value of the busy bit (BUSY bit, bit 16) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BUSY
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BUSY?
```

Arguments

ONE filters status words to only match those where the BUSY bit has a value of 1.

ZERo filters status words to only match those where the BUSY bit has a value of 0.

NOCARE disables filtering of status words on the BUSY bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:BUSY ONE specifies filtering status words for those where the BUSY bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:BUSY? might return NOCARE, indicating that status words are not being filtered based on the BUSY bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:DBCA

This command sets or queries the value of the dynamic bus control acceptance bit (DBCA bit, bit 18) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:DBCA
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:DBCA?
```

Arguments

ONE filters status words to only match those where the DBCA bit has a value of 1.

ZERo filters status words to only match those where the DBCA bit has a value of 0.

NOCARE disables filtering of status words on the DBCA bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:DBCA ONE specifies filtering status words for those where the DBCA bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:DBCA? might return NOCARE, indicating that status words are not being filtered based on the DBCA bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:INSTR

This command sets or queries the value of the instrumentation bit (INSTR bit, bit 10) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:INSTR
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:INSTR?
```

Arguments `ONE` filters status words to only match those where the INSTR bit has a value of 1.

`ZERo` filters status words to only match those where the INSTR bit has a value of 0.

`NOCARE` disables filtering of status words on the INSTR bit.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:INSTR ONE` specifies filtering status words for those where the INSTR bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:INSTR?` might return NOCARE, indicating that status words are not being filtered based on the INSTR bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:ME

This command sets or queries the value of the message error bit (ME bit, bit 9) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:ME
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:ME?
```

Arguments

ONE filters status words to only match those where the ME bit has a value of 1.

ZERo filters status words to only match those where the ME bit has a value of 0.

NOCARE disables filtering of status words on the ME bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:ME ONE specifies filtering status words for those where the ME bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:ME? might return NOCARE, indicating that status words are not being filtered based on the ME bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SRQ

This command sets or queries the value of the status word service request bit (SRQ bit, bit 11) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SRQ
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SRQ?
```

Arguments

ONE filters status words to only match those where the SRQ bit has a value of 1.

ZERo filters status words to only match those where the SRQ bit has a value of 0.

NOCARE disables filtering of status words on the SRQ bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:SRQ ONE specifies filtering status words to those where the SRQ bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:SRQ? might return NOCARE, indicating that status words are not being filtered based on the SRQ bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SUBSF

This command sets or queries the value of the subsystem flag bit (SUBSF bit, bit 17) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x>. The search condition must be set to STATus.

Conditions Requires SR-AERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SUBSF
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SUBSF?
```

Arguments

`ONE` filters status words to only match those where the SUBSF bit has a value of 1.

`ZERo` filters status words to only match those where the SUBSF bit has a value of 0.

`NOCARE` disables filtering of status words on the SUBSF bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:SUBSF ONE` specifies filtering status words for those where the SUBSF bit has a value of 1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:SUBSF?` might return NOCARE, indicating that status words are not being filtered based on the SUBSF bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:TF

This command sets or queries the value of the terminal flag bit (TF bit, bit 19) in a status word for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:TF
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:TF?
```

Arguments

ONE filters status words to only match those where the TF bit has a value of 1.

ZERo filters status words to only match those where the TF bit has a value of 0.

NOCARE disables filtering of status words on the TF bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:TF ONE specifies filtering status words for those where the TF bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:BIT:TF? might return NOCARE, indicating that status words are not being filtered based on the TF bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:PARity

This command sets or queries the value of the status word parity bit for a MIL-STD-1553 bus to search on. The search number is specified by x. The search condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:PARity
{ONE|ZERo|NOCARE}
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:PARity?
```

Arguments

ONE filters status words to only match those where the parity bit has a value of 1.

ZERo filters status words to only match those where the parity bit has a value of 0.

NOCARE disables filtering of status words on the parity bit.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:PARITY ONE specifies filtering status words for those where the parity bit has a value of 1.

SEARCH:SEARCH1:TRIGGER:A:BUS:MIL1553B:STATUS:PARITY? might return NOCARE, indicating that status words are not being filtered based on the parity bit value.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:NRZ:CONDition

This command specifies a field or condition for an NRZ bus to search on. The search number is specified by x.

Conditions Requires the SRNRZ Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:NRZ:CONDition {DATa}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:NRZ:CONDition?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:SIZe

Arguments `Search<x>` is the number of the search.

`DATa` specifies the search condition as Data Characters.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:CONDition DATA` specifies data as the field within an NRZ frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:CONDITION?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:CONDITION DATA`, indicating that DATA is the currently specified field within an NRZ frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:SIZe

This command specifies the length of the data string, in bytes, to use when searching on the NRZ signal. The search condition needs to be set to DATa. The search number is specified by x.

Conditions Requires the SRNRZ Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:SIZe?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:B:NRZ:CONDition

Arguments `Search<x>` is the number of the search.

`<NR1>` specifies the data string in a valid format. The minimum and default values are 1. The maximum value is 5.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:DATA:SIZE 1` sets the data string for NRZ bus trigger search 1 to 1 byte.

`SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:DATA:SIZE?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:DATA:SIZE 3`, indicating that the length of the data string for NRZ bus trigger search 1 is 3 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:VALue

This command sets or queries the binary data value to be used when searching on a NRZ signal. The search condition must be set to DATA.

Conditions Requires the SRNRZ Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:VALue?
```

Arguments `Search<x>` is the number of the search.

`<QString>` specifies the data value in a valid format.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:NRZ:DATa:VALue "1111"` sets the data value to 1111.

`SEARCH:SEARCH1:TRIGger:A:BUS:NRZ:DATa:VALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:NRZ:DATA:VALUE "1010"`, indicating the data value is 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:PARallel:DATa:VALue

This command sets or queries the binary data string used for a parallel bus search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:PARallel:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:PARallel:DATa:VALue?
```

Arguments

<QString> specifies the data value in a valid format. Valid characters are 0-9,

Examples

SEARCH:SEARCH2:TRIGGER:A:BUS:PARALLEL:DATA:VALUE "XXXXXXXX" sets the string data value to "XXXXXXXX" in binary format.

SEARCH:SEARCH1:TRIGGER:A:BUS:PARALLEL:DATA:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:PARALLEL:DATA:VALUE "XXXXXXXXXXXXXXX10010", indicating that the data value string is set to "XXXXXXXXXXXXXXX10010" in binary format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:BLOCKDATa:VALue

This command sets or queries the binary block data string used for PSI5 triggering if the trigger condition is block id.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:BLOCKDATa:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:BLOCKDATa:VALue?
```

Arguments

<QString> specifies the binary block data string.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:BLOCKDATa:VALue "1100001111" sets the block data value as the field within a PSI5 frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:BLOCKDATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:BLOCKDATa:VALue ”1100001111”, indicating that the binary block data value is the currently "1100001111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:CONDition

This command sets or queries the trigger condition for an PSI5 bus. The search number is specified by x.

Conditions Requires option SR-PSI5

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:CONDition
{STARt|STATus|DATa|BLOCkid|SENSORSTATus|ERRors|STARTBIT|FUNCTIONCODe|EC
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:CONDition?
```

Arguments

`STARt` specifies the search condition as Start. This is the default trigger.

`STATus` specifies the search condition as Status.

`DATa` specifies the search condition as Data.

`BLOCkid` specifies the search condition as Block ID.

`SENSORSTATus` specifies the search condition as Sensor Status.

`ERRors` specifies the search condition as Errors.

`STARTBIT` specifies the search condition as Start Bit.

`FUNCTIONCODe` specifies the search condition as Function Code.

`ECUDATa` specifies the search condition as ECU Data.

`SENSORADDRess` specifies the search condition as Sensor Address.

`REGISTERADDRess` specifies the search condition as Register Address.

`SENSORSTATus` specifies the search condition as Sensor Status.

`CRCERRor` specifies the search condition as CRC Error.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:CONDition STARt` specifies start as the field within a PSI5 frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:CONDition?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:CONDition STARt`, indicating that start is the currently specified field within a PSI5 frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:ECU:SENSor:VALue

This command sets or queries the binary Data string used for PSI5 triggering if the trigger condition is Data in ECU to Sensor communication mode. The search number is specified by x.

Conditions Requires option SR-PSI5

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:ECU:SENSor:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:ECU:SENSor:VALue?
```

Arguments <QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:B:PSIFIVe:DATa:ECU:SENSor:VALue "01111001" sets the data value as "01111001".

SEARCH:SEARCH1:TRIGger:A:BUS:B:PSIFIVe:DATa:ECU:SENSor:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:B:PSIFIVe:DATa:ECU:SENSor:VALue "0111100111", indicating that the data value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:REGion:A:VALue

This command sets or queries the binary data string used for PSI5 triggering if the trigger condition is data region A. The search number is specified by x.

Conditions Requires option SR-PSI5

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:REGion:A:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:REGion:A:VALue?
```

Arguments <QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:A:VALue "0111100111" sets the data value as "0111100111".

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:A:VALue? might return

:SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:A:VALue "0111100111", indicating that the data value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:REGion:B:VALue

This command sets or queries the binary data string used for PSI5 triggering if the trigger condition is data region B. The search number is specified by x.

Conditions Requires option SR-PSI5

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:REGion:B:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:REGion:B:
VALue?
```

Arguments <QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:B:VALue "0111100111" sets the data value as "0111100111".

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:B:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATa:REGion:A:VALue "0111100111", indicating that the data value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATABITs

This command sets or queries the binary register address string used for PSI5 triggering if the trigger condition is Data in ECU to Sensor communication mode. The search number is specified by x.

Conditions Requires option SR-PSI5

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATABITs
{FOURBIT|EIGHTBIT|TWENTYBIT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATABITs?
```

Arguments

`FOURBIT` specifies the data bit value as 4 bit.

`EIGHTBIT` specifies the data bit value as 8 bit.

`TWENTYBIT` specifies the data bit value as 20 bit.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATABITs FOURBIT` specifies sets data bit as 4 bit.

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATBITs?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:DATBITs EIGHTBIT`, indicating that the data bit is 8 bit.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:ERRORTYPe

This command sets or queries the error type for an PSI5 bus. The search number is specified by x.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:ERRORTYPe
{ANY|PARity|CRC|RESPONSECODe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:ERRORTYPe?
```

Arguments

`ANY` specifies the error type value as any. This is the default value.

`PARity` specifies the error type value as parity.

`CRC` specifies the error type value as CRC.

`RESPONSECODe` specifies the error type value as response code.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:ERRORTYPe PARity` specifies parity as the field within a PSI5 frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:ERRORTYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:ERRORTYPe PARity`, indicating that parity is the current specified field within a PSI5 frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:FUNCTIONCODETYPe

This command sets or queries the binary data string used for PSI5 triggering if the trigger condition is a Function Code. The search number is specified by x.

Conditions Requires option SR-PSI5

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:FUNCTIONCODETYPe
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:FUNCTIONCODETYPe?
```

Arguments <QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:FUNCTIONCODETYPe "011"` specifies function code as "011".

`SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:FUNCTIONCODETYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:FUNCTIONCODETYPe "011"`, indicating that Function Code Type is currently "011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:REGister:ADDRess:VALue

This command sets or queries the binary register address string used for PSI5 triggering if the trigger condition is Register Address. The search number is specified by x.

Conditions Requires option SR-PSI5

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:REGister:ADDRess:
VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:REGister:ADDRess:
VALue?
```

Arguments <QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:B:PSIFIVe:REGister:ADDRess:VALue "0111100111" sets the register address value as "0111100111".

SEARCH:SEARCH1:TRIGger:A:BUS:B:PSIFIVe:REGister:ADDRess:VALue? might return :SEARCH:SEARCH1:TRIGger:A:BUS:B:PSIFIVe:REGister:ADDRess:VALue "011", indicating that the register address value is "011".

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:REGISTERADDRess

This command sets or queries the register address for PSI5 bus. The search number is specified by x.

Conditions Requires option SR-PSI5

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:REGISTERADDRess
{TWOBIT|SIXBIT|EIGHTBIT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:REGISTERADDRess?
```

Arguments TWOBIT specifies the register address value as 2 bit. This is the default value.

SIXBIT specifies the register address value as 6 bit.

EIGHTBIT specifies the register address value as 8 bit.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:REGISTERADDRess SIXBIT specifies the register address as 6 bit as the field within a PSI5 frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:REGISTERADDRess? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:REGISTERADDRess SIXBIT, indicating that register address as 6 bit is the currently specified field within a PSI5 frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:SENSor:ADDRess:VALue

This command sets or queries the binary sensor address string used for PSI5 triggering if the trigger condition is Sensor Address. The search number is specified by x.

Conditions Requires option SR-PSI5

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:SENSor:ADDRess:
VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:SENSor:ADDRess:
VALue?
```

Arguments <QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSor:ADDRess:VALue "101" specifies sensor address as "101".

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSor:ADDRess:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSor:ADDRess:VALue "101", indicating that the sensor address value is currently "101".

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:SENSORSTATus:TYPe

This command sets or queries the sensor status for PSI5 bus.

Conditions Requires option SR-PSI5

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:SENSORSTATus:TYPe
{READy|BUSY|DEFect|SERVICEMODe|UNLocked}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:SENSORSTATus:TYPe?
```

Arguments

READy specifies the sensor status as ready. This is the default value.

BUSY specifies the sensor status as busy.

DEFect specifies the sensor status as defect.

SERVICEMODe specifies the sensor status as service mode.

UNLocked specifies the sensor status as unlocked.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSORSTATus:TYPe READy specifies ready as the field within a PSI5 frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSORSTATus:TYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:SENSORSTATus:TYPe READy, indicating that ready is the specified field within a PSI5 frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:STATus:VALue

This command sets or queries the binary status string used for PSI5 triggering if the trigger condition is status.

Conditions

Requires option SR-PSI5

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:STATus:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:STATus:VALue?
```

Arguments

<QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:STATus:VALue "10" specifies the status value as "10".

SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:STATus:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:PSIFIVe:STATus:VALue "10", indicating that the status value is 10.

SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition

This command sets or queries the condition for an RS232C bus search to determine where to place a mark. The search number is specified by x.

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition
{DATa|EOp|PARItyerror|STARt}
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition?
```

Arguments

`DATa` specifies the search condition as Data.

`EOp` specifies the search condition as End of Packet.

`PARItyerror` specifies the search condition as Parity Error.

`STARt` specifies the search condition as Start.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:CONDITION DATA` sets the search condition for RS232C bus trigger search 1 to Data.

`SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:CONDITION?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:CONDITION EOP`, indicating that the search condition for RS232C bus trigger search 2 is set to End of Packet.

SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:SIZe

This command sets or queries the length of the data string in bytes to be used for an RS232 bus search to determine where to place a mark when the search condition is Data. The search number is specified by x.

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:SIZe <NR3>
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:SIZe?
```

Arguments <NR3> is the number of bits per word in the data string, from 1 to 8.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:SIZE 1 sets the number of bits per word in RS232C bus trigger search 1 data string to 1.

SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:DATA:SIZE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:RS232C:DATA:SIZE 8, indicating that the bits per word for RS232C bus trigger search 2 is set to 8.

SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:VALue

This command sets or queries the data string used for the specified RS232C bus trigger search to determine where to place a mark. The search condition must be Data. The search number is specified by x.

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:VALue?
```

Arguments <QString> specifies the value of the data string. The valid characters are 0, 1, and X for values in binary format; and A-F, 0-9, and X for values in hexadecimal format.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:VALUE "01" sets the value of the data string for RS232C bus trigger search 1 to "01" when the format is hexadecimal.

SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:RS232C:DATA:VALUE "XXXXXXX1", indicating that the data string value for RS232C bus trigger search 1 is set to "XXXXXXX1" when the format is binary.

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition

This command sets or queries the type of data search for an 8B10B bus. The search number is specified by x.

Conditions Requires the SR8B10b Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition
{SYMbols|ERRors}
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:DISParity

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:ERRor

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:MODe

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TYPe

Arguments `Search<x>` is the number of the search.

`SYMbols` specifies the search condition as data or control.

`ERRors` specifies the search condition as Symbol error or disparity error.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:CONDition SYMBOL` specifies Symbol as the field within an 8B10b frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:CONDITION?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:CONDITION SYMBOL`, indicating that SYMBOL is the currently specified field within an 8B10b frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:DISParity

This command sets or queries the disparity to be used when searching on a 8B10b signal. The search condition must be set to Symbols > 8-bit.

Conditions Requires the SR8B10b Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:DISParity
{POSITIVe|NEGAtive|EITHer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:DISParity?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition

Arguments

`Search<x>` is the Search identifier number.

`POSITIVe` specifies searching for positive disparity.

`NEGAtive` specifies searching for negative disparity.

`EITHer` specifies searching for neutral or either positive/negative disparity.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:DISParity POSITIVE` sets the disparity to positive.

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:DISParity?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:DISParity POSITIVE`, indicating the disparity is positive.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:ERRor

This command sets or queries the error type to be used when searching on a 8B10b signal. The search condition must be set to Errors.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:ERRor
{SYMBol|DISParity}
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:ERRor?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition

Arguments

`Search<x>` is the Search identifier number.

`SYMBol` specifies searching for Symbol Errors.

`DISParity` specifies searching for Disparity Errors.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:ERRor DISParity` sets the error type to Disparity.

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:ERRor?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:ERRor DISPARITY`, indicating the Error type to be disparity.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue

This command sets or queries the control symbol value to be used when searching on 8b10b signal. The default is all X's (don't care). The search condition needs to be set to Symbols. The search number is specified by x.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition

Arguments

`Search<x>` is the Search identifier number.

`<Qstring>` specifies the 10 bit value.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue "D28.5"` specifies the value to be used while searching.

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:CSYMVALue "D28.5"`, indicating the value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue

This command specifies the 8 bit value to be used when searching on an 8b10b signal. The default is all X's (don't care). The search condition needs to be set to Symbols. The search number is specified by x.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition

Arguments `Search<x>` is the Search identifier number.

`<Qstring>` specifies the 8 bit value.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue "00001000"` specifies hexadecimal 08 as the value to be used while searching.

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:EBITVALue "00001000"`, indicating the binary value and 08 as the hexadecimal value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:MODe

This command sets or queries the mode of data search to be used when searching on 8b10b signal. The search condition must be set to Symbols.

Conditions Requires the SR8B10b Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:MODe
{EBIT|TBIT|SYMB}
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:MODe?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition

Arguments `Search<x>` is the number of the search.

`EBIT` specifies searching for Eight Bit data.

`TBIT` specifies searching for Ten bit data (6 bits and 4 bits combined). Only valid values are considered.

SYMB specifies searching for symbol data with disparity. Takes only K and D with possible values..

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:MODe TBIT sets the symbol mode to tbit.

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:MODe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:MODe TBIT, indicating the symbol mode to be tbit.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg

This command specifies the 10 bit RD negative value to be used when searching on an 8b10b signal. The default is all X's (don't care). The search condition needs to be set to Symbols. The search number is specified by x.

Conditions

Requires the SR8B10b Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition

Arguments

Search<x> is the Search identifier number.

<Qstring> specifies the 10 bit RD negative value.

Examples

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg "0011101010" specifies hexadecimal 0EA as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:TBITRDNeg "0011101010", indicating the binary value and 0EA as the hexadecimal value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos

This command specifies the 10 bit RD positive value to be used when searching on an 8b10b signal. The default is all X's (don't care). The search condition needs to be set to Symbols. The search number is specified by x.

Conditions Requires the SR8B10b Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition

Arguments `Search<x>` is the Search identifier number.

`<Qstring>` specifies the 10 bit RD positive value.

Examples `SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos "0011101010"` specifies hexadecimal 0EA as the value to be used while searching.

`SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:TBITRDPos "0011101010"`, indicating the binary value and 0EA as the hexadecimal value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue

This command specifies the 10 bit value to be used when searching on an 8b10b signal. The default is all X's (don't care). The search condition needs to be set to Symbols. The search number is specified by x.

Conditions Requires the SR8B10b Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue
<QString>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue?

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition

Arguments Search<x> is the Search identifier number.

<Qstring> specifies the 10 bit value.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue "0011101010" specifies hexadecimal 0EA as the value to be used while searching.

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:TBITVALue "0011101010", indicating the binary value and 0EA as the hexadecimal value to be used while searching.

SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TYPe

This command sets or queries the type of symbol when searching on 8b10b signal. The search condition must be set to Symbols.

Conditions Requires the SR8B10b Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TYPe
{DATa|CONTrol}
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TYPe?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition

Arguments Search<x> is the number of the search.

DATa specifies searching for Data Symbols.

CONTrol specifies searching for Control Symbols.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TYPe DATA sets the symbol type to data.

SEARCH:SEARCH1:TRIGger:A:BUS:S8B10B:SYMbol:TYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:S8B10B:SYMbol:TYPe DATA, indicating the symbol type to be data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ADDRess:TYPe

This commands specifies the Address type in SDLC frame which can be Broadcast, No-Station and Address value can be specified.

Conditions Requires option SR-SDLC.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ADDRess:TYPe
{BROadcast|NOSTATion|STATIONADDr}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ADDRess:TYPe?
```

Arguments Search<x> is the number of the search.

BROadcast specifies the Broadcast as the field within a SDLC frame to search on. BROadcast is the default value.

NOSTATion specifies the No-Station as the field within a SDLC frame to search on.

STATIONADDr specifies the Address as the field within a SDLC frame to search on.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ADDRess:TYPe BROadcast specifies the Address Type as the field within a SDLC frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ADDRess:TYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ADDRess:TYPe BROadcast, indicating that the Address Type is specified as Broadcast.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:CONDition

This command sets or queries the trigger condition for a SDLC bus.

Conditions Requires option SR-SDLC.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:CONDition {STARt| DATa|
ABORt| ADDRess| UNNumbered| INFormation| SUPervisory|
ERRors| END}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:CONDition?
```

Arguments

`Search<x>` is the number of the search.

`STARt` specifies start as the field within a SDLC frame to search on. The default Trigger on Condtion is STARt.

`DATa` specifies data as the field within a SDLC frame to search on.

`ABORt` specifies abort as the field within a SDLC frame to search on.

`ADDRess` specifies address as the field within a SDLC frame to search on.

`UNNumbered` specifies unnumbered as the field within a SDLC frame to search on.

`INFormation` specifies information as the field within a SDLC frame to search on.

`SUPervisory` specifies supervisory as the field within a SDLC frame to search on.

`ERRors` specifies errors as the field within a SDLC frame to search on.

`END` specifies end as the field within a SDLC frame to search on.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:CONDition STARt` specifies start as the field within a SDLC frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:CONDition?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:CONDition STARt`, indicating that start is the currently specified field within a SDLC frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:SIZe

This command specifies the length of the data string in bytes to be used for SDLC triggering if the trigger condition is DATA. The search number is specified by x.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:SIZe <NR3>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:SIZe?
```

Arguments <NR3> is the length of the data string in bytes. The default Data size is 1, otherwise ranges between 1 to 5.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:SIZe 4 sets the data size value as 4.

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:SIZe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:SIZe 4, indicating that the data size value is 4.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:VALue

This command specifies the binary data string used for SDLC triggering if the trigger condition is DATA. The search number is specified by x.

Conditions Requires option SR-SDLC.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:VALue?
```

Arguments <QString> specifies the value of the data string. The default value is XXXXXXXX .

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:VALue "0111100111" sets the Data value as "0111100111".

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:DATa:VALue "0111100111", indicating that the data value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ERROR:TYPe

This command sets or queries the Error Type in SDLC bus.

Conditions Requires option SR-SDLC.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ERROR:TYPe
{FCS|NUMERICORDer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ERROR:TYPe?
```

Arguments

`Search<x>` is the number of the search.

`FCS` sets FCS as the Error Type in SDLC bus. The default error type condition is FCS

`NUMERICORDer` sets NUMERICORDer as the Error Type in SDLC bus.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ERROR:TYPe FCS` specifies parity as the field within a FCS frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ERROR:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:ERROR:TYPe FCS`, indicating that FCS is the currently specified field within a FCS frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:FRAMe:TYPe

This command sets or queries the Frame Type in SDLC bus.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:FRAMe:TYPe
{COMMand|RESPonse|BOTH}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:FRAMe:TYPe?
```

Arguments

`Search<x>` is the number of the search.

`COMMand` sets Command as the Frame Type in SDLC bus. The default value of frame type for SDLC Unnumbered frame is Command.

`RESPonse` sets Response as the Frame Type in SDLC bus.

`BOTH` sets both Command and Response as the Frame Type in SDLC bus.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:FRAMe:TYPe RESPonse` specifies the frame type as Response.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:FRAMe:TYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:FRAMe:TYPe RESPonse`,

indicating that frame type is Response for Unnumbered frames and accordingly packet list will change.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:STADDress:VALue

This command specifies the binary Station Address string used for SDLC triggering if the trigger condition is Address. The search number is specified by x.

Conditions Requires option SR-SDLC.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:STADDress:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:STADDress:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ADDRess:TYPe

Arguments <QString> specifies the value of the data string. The default value is XXXXXXXX.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:STADDress:VALue "0111100111" sets the Station Address value as "0111100111".

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:STADDress:VALue? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:STADDress:VALue "0111100111", indicating that the station address value is "0111100111".

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:SUPervisory:FRAMETYPe

This command sets or queries the Supervisory Frame Type in SDLC bus.

Conditions Requires option SR-SDLC.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:SUPervisory:FRAMETYPe
{RR|RNR|REJ}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:SUPervisory:FRAMETYPe?
```

Arguments

`Search<x>` is the number of the search.

`RR` sets the frame type as RR.

`RNR` sets the frame type as RNR.

`REJ` sets the frame type as REJ.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:SUPervisory:FRAMETYPe RNR` specifies Receive Not Ready supervisory packet type.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:SUPervisory:FRAMETYPe?` might return `:SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:SUPervisory:FRAMETYPe RNR`, indicating that RNR is the currently specified SDLC Supervisory Packet to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:UNNumbered:FRAMETYPe

This command sets or queries the Unnumbered Frame Type in SDLC bus.

Conditions

Requires option SR-SDLC.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:UNNumbered:FRAMETYPe
{UP| UI| SNRM| SNRME| REQDISConnect| UACK| REQSETINIT|
DISCMODe| FRAMEREJ| TEST| XID| CONFigure| BEACon}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:UNNumbered:FRAMETYPe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:FRAMe:TYPe

Arguments

`Search<x>` is the number of the search.

`UP` sets the Unnumbered Frame Type as UP.

`UI` sets the Unnumbered Frame Type as UI.

`SNRM` sets the Unnumbered Frame Type as SNRM.

`SNRME` sets the Unnumbered Frame Type as SNRME.

`REQDISConnect` sets the Unnumbered Frame Type as REQDISConnect.

`UACK` sets the Unnumbered Frame Type as UACK.

`REQSETINIT` sets the Unnumbered Frame Type as REQSETINIT.

DISCMODe sets the Unnumbered Frame Type as DISCMODe.

FRAMEREJ sets the Unnumbered Frame Type as FRAMEREJ.

TEST sets the Unnumbered Frame Type as TEST.

XID sets the Unnumbered Frame Type as XID.

CONFigure sets the Unnumbered Frame Type as CONFigure.

BEACon sets the Unnumbered Frame Type as BEACon.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:UNNumbered:FRAMETYPe SNRME specifies the frame type as SNRME.

SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:UNNumbered:FRAMETYPe? might return :SEARCH:SEARCH1:TRIGGER:A:BUS:SDLC:UNNumbered:FRAMETYPe SNRME, indicating that Unnumbered Frame type is SNRME when related command Frame Type is set as COMMand.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

This command sets or queries the search condition for a SENT bus.

Conditions

Requires option SRAUTOSEN.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition
{START|FAST|SLOW|PAUSE|ERRor}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition?
```

Arguments

Search<x> is the number of the search.

START specifies searching for start of packet.

FAST specifies searching for fast channel data.

SLOW specifies searching for slow channel data.

PAUSE specifies searching for pause pulses.

ERRor specifies searching on errors.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:CONDITION SLOW specifies searching for slow channel packet.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:CONDITION? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:CONDITION PAUSE, indicating the search is set to find pause pulses.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType

This command sets or queries the error type to be used when searching on SENT data.

Conditions Requires option SRAUTOSEN.

SENT bus trigger condition is set to ERRor.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType
{FRAMELENgth|CRC}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments Search<x> is the number of the search.

FRAMELENgth specifies searching for SENT frame length errors.

CRC specifies searching for CRC errors.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:SENT:ERRTYPE FRAMELENGTH sets the search to find SENT frame length errors.

SEARCH:SEARCH1:TRIGger:A:BUS:SENT:ERRTYPE? might return SEARCH:SEARCH1:TRIGger:A:BUS:SENT:ERRTYPE CRC to indicate the search is set to find CRC errors.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC

This command sets or queries the CRC error type to be used when searching on SENT data.

Conditions Requires option SRAUTOSEN.

SENT bus trigger condition is set to ERRor and ERRType is set to CRC.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC {FAST|SLOW}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType

Arguments `Search<x>` is the number of the search.

`FAST` specifies searching for CRC errors only in the fast channel.

`SLOW` specifies searching for CRC errors only in the slow channel.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE:CRC SLOW` sets the search to find SENT slow channel CRC errors.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE:CRC?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:ERRTYPE:CRC FAST`, indicating that the SENT fast channel is being searched for CRC errors.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue

This command sets or queries the high binary fast channel 1 value to use when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The SENT bus search condition must be set to FAST.

Group Trigger

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments `Search<x>` is the Search identifier number.

`<Qstring>` sets the Fast Channel 1 binary data high value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:HIVALue "XXXXXXXXXXXX" sets the Fast Channel 1 high value on which to search to XXXXXXXXXXXX, or "don't care."

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:HIVALue? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:HIVALUE "0101XXXX1111", indicating the Fast Channel 1 high value on which to search is set to the binary value 0101XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 1.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:
QUALifier {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue

Arguments

Search<x> is the number of the search.

EQUal specifies the qualifier as Equal.

LESSEQual specifies the qualifier as Less Than or Equal to.

LESSThan specifies the qualifier as Less Than.

MOREEQual specifies the qualifier as More Than or Equal to.

MOREThan specifies the qualifier as More Than.

UNEQual specifies the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER UNEQUAL sets the fast channel 1 data qualifier to not equal for search 1.

SEARCH:SEARCH3:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:QUALIFIER MOREEQUAL to indicate that the fast channel 1 data qualifier is set to greater than or equal for search 3.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue

This command sets or queries the binary fast channel 1 value to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to FAST.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments

Search<x> is the number of the search.

<Qstring> is the Fast Channel 1 binary value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE "XXXXXXXXXXXX" sets the Fast Channel 1 value to XXXXXXXXXXXX, or "don't care."

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN1A:VALUE "0000XXXX1111" to indicate the binary value 0000XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue

This command sets or queries the high binary fast channel 2 value to use when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The SENT bus search condition must be set to FAST.

Group Trigger

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments `Search<x>` is the Search identifier number.

`<Qstring>` sets the Fast Channel 2 high binary data value.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:HIVALue "100000000000"` sets the Fast Channel 2 high value on which to search to 100000000000.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:HIVALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:HIVALUE "0101XXXX1111"`, indicating the Fast Channel 2 high value on which to search is set to the binary value 0101XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for device channel 2.

Conditions Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:
QUALifier {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue

Arguments

`Search<x>` is the number of the search.

`EQUal` specifies the qualifier as Equal.

`LESSEQual` specifies the qualifier as Less Than or Equal to.

`LESSThan` specifies the qualifier as Less Than.

`MOREEQual` specifies the qualifier as More Than or Equal to.

`MOREThan` specifies the qualifier as More Than.

`UNEQual` specifies the qualifier as Unequal.

`INrange` sets the qualifier to inside a range.

`OUTrange` sets the qualifier to outside a range.

Examples

```
SEARCH:SEARCH5:TRIGGER:A:BUS:SENT:FAST:CHAN2B:QUALIFIER
UNEQUAL
```
sets the fast channel 2 data qualifier to not equal for search 5.

```
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:QUALIFIER?
```
might return
```
SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:QUALIFIER
MOREEQUAL
```
to indicate that the fast channel 2 data qualifier is set to greater than or equal for search 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue

This command sets or queries the binary fast channel 2 value to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to FAST.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments

`Search<x>` is the number of the search.

`<Qstring>` is the Fast Channel 2 binary value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:VALUE "111111111111"` sets the Fast Channel 2 value to 111111111111.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:CHAN2B:VALUE "000000000000"` to indicate the binary value 000000000000.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:HIVALue

This command sets or queries the high binary fast message counter value to use when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The SENT bus search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group

Trigger

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:HIVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

BUS:B<x>:SENT:NUMCHANnel

BUS:B<x>:SENT:NIBBLECount

Arguments

`Search<x>` is the Search identifier number.

`<Qstring>` sets the Fast Channel 1 counter binary value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:HIVALUE "XXXXXXXX"` sets the Fast Channel secure counter high value on which to search to "don't care."

SEARCH:SEARCH4:TRIGGER:A:BUS:SENT:FAST:COUNTER:HIVALUE? might return SEARCH:SEARCH4:TRIGGER:A:BUS:SENT:FAST:COUNTER:HIVALUE "11110000" to indicate a binary value of 11110000 on which to search.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:QUALifier

This command sets or queries the qualifier to be used when searching on SENT fast packet bus data for the secure format counter.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:
QUALifier {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:QUALifier?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue

Arguments

Search<x> is the number of the search.

EQUal specifies the qualifier as Equal.

LESSEQual specifies the qualifier as Less Than or Equal to.

LESSThan specifies the qualifier as Less Than.

MOREEQual specifies the qualifier as More Than or Equal to.

MOREThan specifies the qualifier as More Than.

UNEQual specifies the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

SEARCH:SEARCH5:TRIGGER:A:BUS:SENT:FAST:COUNTer:QUALifier LESSThan sets the fast channel 2 data qualifier on search 5 to less than.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTer:QUALifier? might return

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTer:QUALIFIER EQUAL to indicate that the fast channel 2 data qualifier on search 1 is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue

This command sets or queries the binary fast message counter value to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

BUS:B<x>:SENT:NIBBLECount

BUS:B<x>:SENT:NUMCHANnel

Arguments

Search<x> is the number of the search.

<Qstring> is the Fast Channel 1 counter value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:VALUE "XXXXXXXX" sets the Fast Channel 1 secure counter value to "don't care."

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:COUNTER:VALUE "00001111" to search for a binary counter value of 00001111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue

This command sets or queries the binary fast message inverted nibble value to be used when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

BUS:B<x>:SENT:NIBBLECount

BUS:B<x>:SENT:NUMCHANnel

Arguments `Search<x>` is the number of the search.

`<Qstring>` is the fast message inverted nibble binary value.

Examples `TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "XXXX"` sets the Fast Channel 1 secure inverted nibble value to "don't care."

`TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE?` might return `:TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "1111"` to search for a binary inverted nibble value of 1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue

This command sets or queries the binary status value to be used when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The trigger condition must be set to FAST.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue
<Qstring>
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue?

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments Search<x> is the number of the search.

<Qstring> is the binary status binary value.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:STATus:VALue "XXXX" sets the binary status value to XXXX.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:STATUS:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:FAST:STATUS:VALUE "1010" to indicate searching on a binary status value of 1010.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier

This command sets or queries the qualifier to be used when searching on SENT pause pulses.

Conditions Requires option SRAUTOSEN

The SENT bus trigger condition must be set to PAUSE.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:
QUALifier {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MOREEQual|INrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue

Arguments Search<x> is the number of the search.

EQUal specifies the qualifier as Equal.

INrange sets the qualifier to be within a range.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQual sets the qualifier as More Than or Equal to.

MOREThan sets the qualifier as More Than.

OUTrange sets the qualifier to be outside a range.

UNEQual specifies the qualifier as Unequal.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:QUALIFIER LESSTHAN sets the pause pulse qualifier to less than.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:QUALIFIER? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:QUALIFIER INRANGE to indicate that the pause pulse qualifier is set to search inside the range of ticks low and ticks high.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue

This command sets or queries the maximum number of pause clock ticks to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to PAUSE.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue ?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments

Search<x> is the number of the search.

<NR1> is the maximum number of pause clock ticks to be used when searching.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:HIVALUE 15 sets the maximum pause tick count for which to search to 15.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:HIVALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:HIVALUE 20 to indicate the maximum number of pause ticks to search for is 20.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue

This command sets or queries the minimum number of pause clock ticks to be used when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The trigger condition must be set to PAUSE.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments `Search<x>` is the number of the search.

`<NR1>` is the minimum number of pause clock ticks to be used when searching.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:VALUE 12` sets the minimum pause tick count on which to search to 12.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:PAUSE:TICKS:VALUE 15` to indicate the minimum number of pause ticks to search for is 15.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue

This command sets or queries the high binary Slow channel data value to use when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The SENT bus search condition must be set to SLOW.

Group Trigger

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments Search<x> is the Search identifier number.

<Qstring> sets the binary Slow channel data value.

Examples SEARCH:SEARCH2:TRIGGER:A:BUS:SENT:SLOW:DATA:HIVALUE "XXXXXXXX" sets the Slow data high value on which to search to "don't care" for Search 2.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:HIVALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:HIVALUE "11110001", to indicate searching on slow data values that match 11110001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier

This command sets or queries the qualifier to be used when searching on SENT slow packet bus data.

Conditions Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to SLOW.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:
QUALifier {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MOREEQualINrange|OUTrange}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue

Arguments Search<x> is the number of the search.

EQUal specifies the qualifier as Equal.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQual sets the qualifier as More Than or Equal to.

MOREThan sets the qualifier as More Than.

UNEQual specifies the qualifier as Unequal.

INrange sets the search qualifier to inside a range.

OUTrange sets the search qualifier to outside a range.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:QUALIFIER LESSEQUAL sets the slow channel data qualifier to less than or equal.

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:QUALIFIER? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:QUALIFIER EQUAL to indicate that the slow channel data qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue

This command sets or queries the binary slow channel data value to be used when searching on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The trigger condition must be set to SLOW.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments

Search<x> is the number of the search.

<Qstring> is the slow channel data binary value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:VALUE "XXXXXXXX" sets the slow data search value to "don't care."

SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:DATA:VALUE? might return "XXXX1111", indicating the search is marking slow data values that match XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue

This command sets or queries the binary slow identifier value to be used when searching on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The trigger condition must be set to SLOW.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition

Arguments `Search<x>` is the number of the search.

`<Qstring>` is the slow channel identifier binary value.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:IDENTIFIER:VALUE "XXXX"` sets the search identifier value to "don't care."

`SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:IDENTIFIER:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SENT:SLOW:IDENTIFIER:VALUE "0001"` indicating to search for the slow channel identifier value of 0001.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce

This command sets or queries the bus source for the bus search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce
{B0|B1|B2|B3|B4|B5|B6|B7|B8|B9|B10| B11|B12|B13|B14|B15|B16}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce?
```

Arguments `B0–B16` specifies the bus source as a bus number from B01 to B16.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE B1` sets the bus source for bus trigger search 1 to B1.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE B15`, indicating that the bus source for bus trigger search 1 is B15.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition

This command sets or queries the field or condition for which to search on a SpaceWire bus.

Conditions

Requires option SR-SPACEWIRE

This command affects most of the SEARCH:SEARCH<x>:TRIGger:A:BUS:B<x>:SPACEWIRe commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition
{SYNC|DATa|ERRors|CONTROLCODe|CONTROLCHAR}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition?
```

Arguments

`Search<x>` is the search number.

`SYNC` sets the search condition to Synchronization.

`DATa` sets the search condition to Data Characters.

`ERRors` sets the search condition to SpaceWire Bus Decode Errors.

`CONTROLCODe` sets the search condition to control code.

`CONTROLCHAR` sets the search condition to control character.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:CONDition SYNC` sets Synchronization as the field within an SpaceWire frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:CONDITION?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:CONDITION DATA`, indicating that the data field is the currently specified field within an SpaceWire frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe

This command sets or queries the control character type to use when searching on a SpaceWire bus signal.

Conditions Requires option SR-SPACEWIRE

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe
{FCT|ESC|ENDOFPACKet}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition

Arguments

`Search<x>` is the search number.

`FCT` sets the control character type to Flow Control Token.

`ESC` sets the control character type to Escape.

`ENDOFPACKet` sets the control character type to End of Packet.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe FCT` specifies Flow control token as the field on which to search within a SpaceWire frame.

`SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIReCONTROLCHARTYPe ESC`.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe

This command sets or queries the control code type to use when searching on a SpaceWire bus signal.

Conditions Requires option SR-SPACEWIRE

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe
{NULL|TIMECODe}
```

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe?

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition

Arguments Search<x> is the search number.

NULL sets the control code type to Null code.

TIMECODe sets the control code type to Time code (only varying part of Time code [6 bits] is considered for search).

Examples SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe NULL specifies Null as the field on which to search within an SpaceWire frame.

SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe? might return SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIReCONTROLCODETYPe TIMECODe.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to use when searching on a SpaceWire bus signal. The search condition needs to be set to Data (see *Related Commands*).

Conditions Requires option SR-SPACEWIRE

Group Search and Mark

Syntax SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe<NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe?

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition

Arguments Search<x> is the search number.

NR1 is an integer value that specifies the number of contiguous data bytes to use when searching on the SpaceWire data field. The valid range is 1 to 10.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe 4 specifies 4 as the number of contiguous data bytes to use when searching on the SpaceWire data field.

SEARCH:SEARCH5:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe? might return SEARCH:SEARCH5:TRIGger:A:BUS:SPACEWIRe:DATa:SIZE 10, indicating that Search 5 is set to use 10 contiguous data bytes searching on the SpaceWire data field.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:VALue

This command sets or queries the binary data value to be used when searching on an SpaceWire bus signal. The search condition must be set to Data (see *Related Commands*).

Conditions Requires option SR-SPACEWIRE

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:VALue<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition

Arguments Search<x> is the search number.

QString specifies the binary data value to search for on a SpaceWire bus signal.

Examples SEARCH:SEARCH1:TRIGger:A:BUS:SPACEWIRe:DATa:VALue “101011” sets the data VALue to 101011.

SEARCH:SEARCH5:TRIGger:A:BUS:SPACEWIRe:DATa:VALue? might return SEARCH:SEARCH5:TRIGger:A:BUS:SPACEWIRe:DATa:VALue "10101", indicating that Search 5 is set to look for binary data value 10101.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:ERRORTYPe

This command sets or queries the error type for which to search on the SpaceWire bus signal.

Conditions Requires option SR-SPACEWIRE

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:ERRORTYPe
{EEP|ESC|PARity}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:ERRORTYPe?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition

Arguments

`Search<x>` is the search number.

`EEP` sets the error search type to Error end of packet.

`ESC` sets the error search type to Escape error.

`PARity` sets the error search type to Parity error.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:ERRORTYPe EEP` specifies Error end Of Packet as the field on which to search within a SpaceWire frame.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:ERRORTYPe?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SPACEWIRe:ERRORTYPe ESC`, indicating that Escape Error is the currently specified field on which to search within a SpaceWire frame.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:TIMECode:VALue

This command sets or queries the binary data string used for SpaceWire search if the search condition is Time Code.

Conditions

Requires option SR-SPACEWIRE

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:TIMECode:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:TIMECode:VALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition

Arguments

`Search<x>` is the search number.

`QString` sets the binary data value to be used when searching on an SpaceWire bus signal.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:B:SPACEWIRe:TIMECode:VALue “101011”` sets the timecode value to binary 101011.

`SEARCH:SEARCH1:TRIGger:A:BUS:B:SPACEWIRe:TIMECode:VALue?` might return
`SEARCH:SEARCH1:TRIGger:A:BUS:B:SPACEWIRe:TIMECode:VALue "101011"`.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition

This command sets or queries the search condition for an SPI bus search to determine where to place a mark. The search number is specified by x.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition
{DATA|SS|STARTofframe}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition?
```

Arguments

`DATA` specifies the trigger condition as Data.

`SS` specifies the trigger condition as Slave Selection.

`STARTofframe` specifies the trigger condition as start of frame.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:CONDITION SS` sets the trigger condition for SPI bus trigger search 1 to SS.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:CONDITION?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:CONDITION DATA`, indicating that the trigger condition for SPI bus trigger search 1 is set to Data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZe

This command sets or queries the length of the data string in bytes used for the specified SPI bus trigger search to determine where to place a mark. The search condition must be DATA. The search number is specified by x.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZe?
```

Arguments <NR1> specifies the number of contiguous data bytes.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:SIZE 1 sets the length of the data string for SPI bus trigger search 1 to 1 byte.

SEARCH:SEARCH2:TRIGGER:A:BUS:SPI:DATA:SIZE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:SPI:DATA:SIZE 4, indicating that the length of the data string for SPI bus trigger search 2 is 4 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue

The command sets or queries the binary data string used for an SPI bus search to determine where to place a mark. The search number is specified by x. The search condition must be DATA.

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue?
```

Arguments <QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; and A-F, 0-9, and X for hexadecimal format.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:VALUE "00001111" sets the data value for SPI bus trigger search 1 to "00001111" in binary format.

SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:DATA:VALUE "11", indicating that the data value for SPI bus trigger search 1 is set to "11" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCETYpe

This command sets or queries trigger Source for SPI bus. The search number is specified by x.

Conditions Requires option SR-SPI.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCETYpe {MISo|MOSi}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCETYpe?
```

Arguments `MISo` specifies the trigger source as MISo. The default search source type is MISo.

`MOSi` specifies the trigger source as MOSi.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:SOURCETYpe MOSi` sets the trigger source for SPI bus trigger search 1 to MOSi.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:SOURCETYpe?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SPI:SOURCETYpe MOSi`, indicating that the trigger source for SPI bus trigger search 1 is MOSi.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition

This command sets or queries the search condition for an SPMI bus.

Conditions Requires option SRPM

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:
SPMI:CONDition {SSC|RESet|SLEep|
SHUTdown|WAKeup|MASTERREAd|MASTERWRIte|REGREAd|REGWRIte|
DEVICEDESCMASTERREAd|DEVICEDESCSLAVEREAd|EXTREGREAd|
EXTREGWRIte|LONGEXTREGREAd|LONGEXTREGWRIte|REGOWRIte|
AUTHenticate|TRANSferbusownership|PARItyerror}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition?
```

Arguments `Search<x>` is the search identifier number.

`SSC` specifies searching for the Sequence Start Condition.

RESet specifies searching for the Reset command sequence.

SLEep specifies searching for the Sleep command sequnce.

SHUTdown specifies searching for the Shutdown command sequence.

WAKeup specifies searching for the Wakeup command sequence.

MASTERREAd specifies searching for the Master Read command sequence.

MASTERWRITe specifies searching for the Master Write command sequence.

REGREAd specifies searching for the Register Read command sequence.

REGWRITe specifies searching for the Register Write command sequence.

DEVICEDESCMASTERREAd specifies searching for the Device Descriptor Block Master Read command sequence

DEVICEDESCSLAVEREAd specifies searching for the Device Descriptor Block Slave Read command sequence.

EXTREGREAd specifies triggering on the Extended Register Read command sequence.

EXTREGWRITe specifies searching for the Extended Register Write command sequence.

LONGEXTREGREAd specifies searching for the Extended Register Read Long command sequence.

LONGEXTREGWRITe specifies searching for the Extended Register Write Long command sequence.

REG0WRITe specifies searching for the Register 0 Write command sequence.

AUTHenticate specifies searching for the Authentication command sequence.

TRANSferbusownership specifies searching for the Transfer Bus Ownership (TBO) command sequence.

PARItyerror specifies searching for the parity errors.

Examples

SEARCH:SEARCH4:TRIGger:A:BUS:SPMI:CONDITION SLEEP specifies searching for a Sleep command sequence in search Search4.

SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:CONDITION? might return :SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:CONDITION MASTERREAD, indicating that search Search2 is searching for Master Read command sequences.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to be used when searching on an SPMI bus signal.

Conditions Requires option SRPM.

The Search condition must be set to EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, or LONGEXTREGWRIte.

Note that this a floating point value. If the condition is set to REG0WRIte, the query return value will be a fractional value (0.875) because the data size is less than a byte.

Group Trigger

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:SIZe <NR2>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:SIZe?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:VALue

Arguments `Search<x>` is the Search identifier number.

`<NR2>` is the size of the data string in bytes.

Examples `SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:SIZE 1` sets the data size to 1 byte.

`SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:SIZE?` might return `SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:SIZE 4` to indicate the search data pattern has four bytes of data.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:VALue

This command specifies the binary data string used for SPMI when the search condition is MASTERREAd, MASTERWRIte, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, LONGEXTREGWRIte, or REG0WRIte.

Conditions Requires option SRPM

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition

Arguments Search<x> is the search identifier number.

<Qstring> is the binary data string that identifies the data value for which to search.

Examples SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:VALUE "XXXXXXXX" sets the binary data value to "don't care" for Search 2.

SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:VALUE? might return SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:DATA:VALUE "XXXX1111" to indicate Search 2 is searching for binary data value XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:MASTERADDRess:VALue

This command sets or queries the binary data string that identifies the master address used in SPMI when the search condition is MASTERREAd, MASTERWRIte, or DEVICEDESCMASTERREAd.

Conditions Requires option SRPM

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:MASTERADDRess:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:MASTERADDRess:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition

Arguments Search<x> is the search identifier number.

<Qstring> is the binary data string that identifies the master address for which to search.

Examples SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:MASTERADDRESS:VALUE "XX" sets the binary master address on which to search to "don't care."

SEARCH:SEARCH3:TRIGger:A:BUS:SPMI:MASTERADDRESS:VALUE? might return SEARCH:SEARCH3:TRIGger:A:BUS:SPMI:MASTERADDRESS:VALUE "11" to indicate that Search 3 is searching for binary master address 11.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPonse

This command sets or queries whether or not to search for No Response frames.

Conditions Requires option SRPM

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPonse <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPonse?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition

Arguments

`Search<x>` is the search identifier number.

`<Qstring>` is either "TRUE" or "FALSE" boolean value for searching on SPMI No Response frames.

Examples

`SEARCH:SEARCH6:TRIGger:A:BUS:SPMI:NORESPONSE TRUE` sets Search 6 to look for No Response frames.

`SEARCH:SEARCH4:TRIGger:A:BUS:SPMI:NORESPONSE?` might return `SEARCH:SEARCH4:TRIGger:A:BUS:SPMI:NORESPONSE FALSE` to indicate that the Search 4 is not looking for No Response frames.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:REGISTERADDRess:VALue

This command sets or queries the binary data string that identifies the register address used in SPMI triggering if the trigger condition is MASTERREAd, MASTERWRIte, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, or LONGEXTREGWRIte.

Conditions Requires option SRPM

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:REGISTERADDRess:VALue
<Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:REGISTERADDRess:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition

Arguments `Search<x>` is the search identifier number.

`<Qstring>` is the binary data string that identifies the register address for which to search.

Examples `SEARCH:SEARCH2:TRIGger:A:BUS:SPMI:REGISTERADDRESS:VALUE "XXXXXXXX"` sets the binary register address to "don't care."

`SEARCH:SEARCH5:TRIGger:A:BUS:SPMI:REGISTERADDRESS:VALUE?` might return `SEARCH:SEARCH5:TRIGger:A:BUS:SPMI:REGISTERADDRESS:VALUE "XXXX1111"` to indicate Search 5 is searching for binary register address is XXXX1111.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:SLAVEADDRess:VALue

This command sets or queries the binary data string that identifies the slave address used in SPMI when the search condition is RESet, SLEep, SHUTdown, WAKeup, AUTHenticate, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, LONGEXTREGWRIte, DEVICEDESCSLAVEREAd, or REG0WRIte.

Conditions Requires option SRPM

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:SLAVEADDRess:VALue <Qstring>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:SLAVEADDRess:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition

Arguments `Search<x>` is the search identifier number.

`<Qstring>` is the binary data string that identifies the slave address for which to search.

Examples `SEARCH:SEARCH3:TRIGger:A:BUS:SPMI:SLAVEADDRESS:VALUE "0011"` sets the binary slave address to 0011 on Search 3.

SEARCH:SEARCH1:TRIGger:A:BUS:SPMI:SLAVEADDRESS:VALUE? might return SEARCH:SEARCH1:TRIGger:A:BUS:SPMI:SLAVEADDRESS:VALUE "1010" to indicate the binary slave address is 1010 for Search 1.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:RESPonse

This command sets or queries the command response of the respective command type for SVID bus.

Conditions Requires option SR-SVID

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:RESPonse
{GETReg| GETRegvendor| GETRegtestcfg| SETFast| SETSlow|
SETDecay| SETPs| SETRegaddr| SETRegdata| REGAddrvendor|
REGDatavendor| REGAddrtestconfg| REGDatatestconfg| TESTmode|
SETWp| GETRegpktrecent| GETRegpktalert| GETRegpktbad|
GETRegvrevent}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:RESPonse?
```

Arguments

Search<x> is the search identifier number.

GETReg specifies the search condition as GETReg. This is the default command response condition.

GETRegvendor specifies the search condition as GETRegvendor.

GETRegtestcfg specifies the search condition as GETRegtestcfg.

SETFast specifies the search condition as SETFast.

SETSlow specifies the search condition as SETSlow.

SETDecay specifies the search condition as SETDecay.

SETPs specifies the search condition as SETPs.

SETRegaddr specifies the search condition as SETRegaddr.

SETRegdata specifies the search condition as SETRegdata.

REGAddrvendor specifies the search condition as REGAddrvendor.

REGDatavendor specifies the search condition as REGDatavendor.

REGAddrtestconfg specifies the search condition as REGAddrtestconfg.

REGDatatestconfg specifies the search condition as REGDatatestconfg.

TESTmode specifies the search condition as TESTmode.

SETWp specifies the search condition as SETWp.

GETRegpktrecent specifies the search condition as GETRegpktrecent.

GETRegpktalert specifies the search condition as GETRegpktalert.

GETRegpktbad specifies the search condition as GETRegpktbad.

GETRegvrevent specifies the search condition as GETRegvrevent.

Examples

SEARCH:SEARCH3:TRIGger:A:BUS:SVID:COMMand:RESPonse SETSlow specifies setslow of set command as the field within a SVID frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:SVID:COMMand:RESPonse? might return SEARCH:SEARCH1:TRIGger:A:BUS:SVID:COMMand:RESPonse SETSlow, indicating that SETSlow is the currently specified field within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:TYPe

This command sets or queries the command type for SVID bus.

Conditions

Requires option SR-SVID

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:TYPe
{GET|SET|OPTional}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:TYPe?
```

Arguments

Search<x> is the search identifier number.

GET specifies the search condition as GET. This is the default command type condition.

SET specifies the search condition as SET.

OPTional specifies the search condition as OPTional.

Examples

SEARCH:SEARCH3:TRIGger:A:BUS:SVID:COMMand:TYPe SET specifies SET as the field within a SVID frame to search on.

SEARCH:SEARCH1:TRIGger:A:BUS:SVID:COMMand:TYPe? might return SEARCH:SEARCH1:TRIGger:A:BUS:SVID:COMMand:TYPe SET, indicating that SET is the currently specified field within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:CONDition

This command sets or queries the trigger condition for SVID bus.

Conditions Requires option SR-SVID

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:CONDition
{STARTCONDition| SLAVEADDRess| COMMand| PAYLoad|
ERRors}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:CONDition?
```

Arguments

`Search<x>` is the search identifier number.

`STARTCONDition` specifies the search condition as STARTCONDition. This is the default trigger on condition.

`SLAVEADDRess` specifies the search condition as SLAVEADDRess.

`COMMand` specifies the search condition as COMMand.

`PAYLoad` specifies the search condition as PAYLoad.

`ERRors` specifies the search condition as ERRors.

Examples

`SEARCH:SEARCH3:TRIGger:A:BUS:SVID:SVID:CONDition ERRors` specifies ERRors as the field within a SVID frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:SVID:CONDition?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:SVID:CONDition ERRors`, indicating that ERRors is the currently specified field within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:ERRor:TYPe

This command sets or queries the error type for SVID bus.

Conditions Requires option SR-SVID

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:ERRor:TYPe {ANY| NACK|
REJect| MIXed| PARity}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:ERRor:TYPe?
```

Arguments

`Search<x>` is the search identifier number.

`ANY` specifies the search condition as ANY. This is the default error type condition.

`NACK` specifies the search condition as NACK.

`REJect` specifies the search condition as REJect.

`MIXed` specifies the search condition as MIXed.

`PARity` specifies the search condition as PARity.

Examples

`SEARCH:SEARCH3:TRIGger:A:BUS:SVID:SVID:ERRor:TYPe PARity` specifies PARity as the field within a SVID frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:SVID:ERRor:TYPe?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:SVID:ERRor:TYPe PARity`, indicating that PARity is the currently specified field within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:TYPe

This command sets or queries the payload type for SVID bus.

Conditions

Requires option SR-SVID

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:TYPe
{MASTer|SLAVe|EITHer}
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:TYPe?
```

Arguments

`Search<x>` is the search identifier number.

`MASTer` specifies the search condition as MASTer. This is the default payload type condition.

`SLAVe` specifies the search condition as SLAVe.

`EITHer` specifies the search condition as EITHer.

Examples

`SEARCH:SEARCH3:TRIGger:A:BUS:SVID:SVID:PAYLoad:TYPe SLAVe` specifies SLAVe payload as the field within a SVID frame to search on.

`SEARCH:SEARCH1:TRIGger:A:BUS:SVID:PAYLoad:TYPe?` might return `SEARCH:SEARCH1:TRIGger:A:BUS:SVID:PAYLoad:TYPe PARity`,

indicating that SLAVe is the currently specified field within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLOAD:VALue

This command sets or queries the payload value to be used when search is on SVID bus.

Conditions Requires option SR-SVID

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLOAD:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLOAD:VALue?
```

Arguments `Search<x>` is the search identifier number.

`<QString>` specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary; for hexadecimal; and symbolic).

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:PAYLOAD:VALue "XXXX"` specifies the payload value as XXXX within a SVID frame to search on.

`SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:PAYLOAD:VALue?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:PAYLOAD:VALue "XXXX"`, indicating the payload value is XXXX within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:SLAVE:ADDRESS

This command sets or queries the slave address to be used when search is on SVID bus.

Conditions Requires option SR-SVID

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:SLAVE:ADDRESS <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:SLAVE:ADDRESS?
```

Arguments

Search<x> is the search identifier number.

<QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary; for hexadecimal; and symbolic). the default address is XXXX.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:SLAVE:ADDRESS "XXXX" specifies the pslave address as XXXX within a SVID frame to search on.

SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:SLAVE:ADDRESS? might return SEARCH:SEARCH1:TRIGGER:A:BUS:SVID:SLAVE:ADDRESS "XXXX", indicating the slave address is XXXX within a SVID frame to search on.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:HIVALue

This command sets or queries the high binary address value used when searching on a USB bus signal to determine where to place a mark. The search number is specified by x. The search condition must be set to TOKEN.

The VALue and HIVALue set a range that the INrange and OUTrange qualifiers use to decide when to trigger. For example, if the QUALIFER is set to INrange, and the address is within the range set by VALue and HIVALue, then a trigger can be generated.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:HIVALue?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:VALue

Arguments

<QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary; for hexadecimal; and symbolic).

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:HIVALUE

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:HIVALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:HIVALUE "XX", indicating that the address value for normal token for USB bus trigger search 1 is set to "XX," in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:VALue

This command sets or queries the binary address value used for a USB bus search to determine where to place a mark. The search number is specified by x. The search condition must be set to TOKEN.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:VALue?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:HIVALue

Arguments <QString> specifies the data value in the specified valid format. Valid characters are 0, 1, and X for binary; A-F, 0-9 and X for hexadecimal; and for symbolic.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:VALUE "1110000" sets the address value for the normal token for USB bus trigger search 1 to "1110000" in binary format.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ADDRESS:VALUE "11", indicating that the address value for normal token for USB bus trigger search 1 is set to "11" hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition

This command sets or queries the search condition for a USB bus search to determine where to place a mark. The search number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition
{DATAPacket|EOP|ERRor|HANDSHAKEPacket|RESET|RESUME|
SPECIALPacket|SUSPEND|SYNC|TOKENPacket}
```

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition?
```

Arguments

DATAPacket specifies the search condition as Data Packet.

EOP specifies the search condition as End of Packet.

ERROR specifies the search condition as Error.

HANDSHAKEPacket specifies the search condition as Handshake Packet.

RESET specifies the search condition as Reset.

RESUME specifies the search condition as Resume.

SPECIALPacket specifies the search condition as Special Packet.

SUSPEND specifies the search condition as Suspend.

SYNC specifies the search condition as Sync.

TOKENPacket specifies the search condition as Token (Address) Packet.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CONDITION EOP sets the search condition for USB bus trigger search 1 to End of Packet.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CONDITION? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:CONDITION SYNC, indicating that the search condition for USB bus trigger search 1 is set to Sync.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:HIVALue

This command sets or queries the high binary data value used with In Range and Out of Range qualifiers for a USB bus search to determine where to place a mark. The search number is specified by x. The search condition must be set to DATA.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:HIVALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:HIVALue?
```

Arguments

<QString> specifies the data value in the specified valid format (binary, hexadecimal, or symbolic). The valid characters for binary are 0, 1, and X;

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:HIVALUE "XXXXX111"` sets the data value for USB bus trigger search 1 to "XXXXX111" in binary format.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:HIVALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:HIVALUE "11"`, indicating that the data value for USB bus trigger search 1 is set to "11" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:OFFSet

This command sets the byte offset to look for a data pattern at, in bytes, to be used when searching on a USB bus signal. The search number is specified by x. The search condition must be set to DATA.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:OFFSet <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:OFFSet?
```

Arguments `<NR1>` specifies the data offset in bytes.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:OFFSET 5` sets the data offset for USB bus trigger search 1 to 5 bytes.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:OFFSET?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:OFFSET DONTCARE`, indicating that the data offset for USB bus trigger search 1 is set to DONTCare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:QUALifier

This command sets or queries the qualifier to be used when searching on a USB bus signal. The search condition must be set to IDANDDATA OR DATA. The search number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:
USB:DATa:QUALifier {EQual|UNEQual|
LESSthan|MOREthan|LESSEQual|MOREEQual|INrange|OUTrange}
```

Arguments

`EQUal` specifies the qualifier as Equal.

`INrange` specifies the qualifier as Inside Range.

`LESSEQual` specifies the qualifier as Less Than or Equal to.

`MOREEQual` specifies the qualifier as More Than or Equal to.

`OUTrange` specifies the qualifier as Out of Range.

`UNEQual` specifies the qualifier as Unequal.

`LESSThan` specifies the qualifier as Less Than.

`MOREThan` specifies the qualifier as More Than.

Examples

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:DATa:QUALifier INRANGE` sets the qualifier to INRANGE.

`SEARCH:SEARCH1:TRIGger:A:BUS:USB:DATa:QUALifier?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:QUALIFIER EQUAL` indicating the qualifier is set to equal.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:SIZe

This command sets or queries the length of the data string, in bytes, used for a USB bus search to determine where to place a mark. The search number is specified by x. The search condition must be set to DATA.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:SIZe <NR1>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:SIZe?
```

Arguments

`<NR1>` specifies the data size in bytes.

Examples

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:SIZE 1` sets the size of the data string for USB bus trigger search 1 to 1 byte.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:SIZE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:SIZE 3, indicating that the size of the data string for USB bus trigger search 1 is set to 3 bytes.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:TYPe

This command sets or queries the USB bus search type. The search number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:TYPe
{ANY|DATA0|DATA1|DATA2|MDATA}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:TYPe?
```

Arguments

ANY specifies the data packet type as Any.

DATA0 specifies the data packet type as DATA0.

DATA1 specifies the data packet type as DATA1.

DATA2 specifies the data packet type as DATA2.

MDATA specifies the data packet type as MDATA.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:TYPE ANY sets the data packet type for USB bus trigger search 1 to ANY.

SEARCH:SEARCH2:TRIGGER:A:BUS:USB:DATA:TYPE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:USB:DATA:TYPE MDADA, indicating that the packet type for USB bus trigger search 2 is set to MDATA.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:VALue

This command sets or queries the binary data value used for a USB bus search to determine where to place a mark. The search number is specified by x. The search condition must be set to DATA.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:VALue?
```

Arguments <QString> specifies the data value. The valid characters are 0, 1, and X for binary format; and A-F, 0-9, and X for hexadecimal format.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:VALUE "AB" sets the data value for data token for USB bus trigger search 1 to "AB" in hexadecimal format.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:DATA:VALUE "XXXXXXXX", indicating that the data value for data token for USB bus trigger search 1 is set to "XXXXXXXX" in binary format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALue

This command sets or queries the endpoint binary value used for a USB bus search to determine where to place a mark. The search number is specified by x. The search condition must be set to TOKEN.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALue?
```

Arguments <QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format; A-F, 0-9 and X for hexadecimal format.

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ENDPOINT:VALUE "1111" sets the endpoint value for normal token for USB bus trigger search 1 to "1111" in binary format.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ENDPOINT:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ENDPOINT:VALUE "1", indicating

that the endpoint value for normal token for USB bus trigger search 1 is set to "1" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE

This command sets or queries the error type for a USB bus search to determine where to place a mark. The search number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE
{BITSTUFFing|CRC5|CRC16|PID}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE?
```

Arguments `BITSTUFFing` specifies the error type as Bit Stuffing.

`CRC5` specifies the error type as Token CRC5 (Cyclic Redundancy Check 5).

`CRC16` specifies the error type as Data CRC16 (Cyclic Redundancy Check 16).

`PID` specifies the error type as PID Check Bits.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ERRTYPE CRC5` sets the error type for USB bus trigger search 1 to Token CRC5.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ERRTYPE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:ERRTYPE PID`, indicating that the error type for USB bus trigger search 1 is set to PID Check Bits.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HANDSHAKEType

This command sets or queries the handshake type for the specified USB bus trigger search to determine where to place a mark. The search number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HANDSHAKEType
{ACK|ANY|NAK|NYET|STALL}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HANDSHAKEType?
```

Arguments

ACK specifies the handshake type as Acknowledgement (positive) (XX10).

ANY specifies the handshake type as Any (0010).

NAK specifies the handshake type as Negative Acknowledgment (1010).

NYET specifies the handshake type as No response Yet (0110).

STALL specifies the handshake type as Stall (endpoint is halted or control pipe request not supported) (1110).

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:HANDSHAKETYPE NAK sets the handshake type for USB bus trigger search 1 to NAK.

SEARCH:SEARCH2:TRIGGER:A:BUS:USB:HANDSHAKETYPE? might return :SEARCH:SEARCH2:TRIGGER:A:BUS:USB:HANDSHAKETYPE NYET, indicating that the handshake type for USB bus trigger search 2 is set to NYET.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SOFFRAMENUMber

This command sets or queries the frame number string to use for the Start of Frame for the specified USB bus trigger search to determine where to place a mark. The search number is specified by x. The search condition must be set to TOKEN.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SOFFRAMENUMber <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SOFFRAMENUMber?
```

Arguments

<QString> specifies the frame number string for the Start of Frame in a valid format (binary, hexadecimal, or symbolic).

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SOFFRAMENUMBER "1001" sets the frame number string for the SOF for USB bus trigger search 1 to "1001" in symbolic format.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SOFFRAMENUMBER? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SOFFRAMENUMBER "xxxxxxx1001", indicating that the frame number string for the SOF for USB bus trigger search 1 is set to "XXXXXXX1001" in binary format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPECIALType

This command sets or queries the USB search type for the specified USB bus trigger search to determine where to place a mark. The search number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPECIALType
{ANY|ERROr|PING|PREamble|RESERVed|SPLit}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPECIALType?
```

Arguments

ANY specifies the PID value as Any (XX00).

ERROr specifies the PID value as ERR (1100).

PING specifies the PID value as PING (0100).

PREamble specifies the PID value as PRE (1100).

RESERVed specifies the PID value as Reserved (0000).

SPLit specifies the PID value as Split (1000).

Examples SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPECIALTYPE ERROR sets the PID value for USB bus trigger search 1 to ERROR.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPECIALTYPE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPECIALTYPE SPLIT, indicating that the PID value for USB bus trigger search 1 is set to SPLIT.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:ET:VALue

This command sets or queries the Endpoint Type value for the specified USB bus trigger search on split token field to determine where to place a mark. The search number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:ET:VALue
{BULK|CONTROL|NOCARE|INTERRUPT|ISOchronous}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:ET:VALue?
```

Arguments `BULK` specifies the Endpoint Type value as Bulk (10).

`CONTROL` specifies the Endpoint Type value as Control (00).

`NOCARE` specifies the Endpoint Type value as Don't Care.

`INTERRUPT` specifies the Endpoint Type value as Interrupt (11).

`ISOchronous` specifies the Endpoint Type value as Isochronous (01).

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:ET:VALUE BULK` sets the Endpoint Type value for USB bus trigger search 1 to Bulk.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:ET:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:ET:VALUE DONTCARE`, indicating that the Endpoint Type value for USB bus trigger search 1 is set to DONTCare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:HUB:VALue

This command sets or queries the binary hub address value to be used when searching on a USB bus signal. The search number is specified by x. The search condition must be set to Special with packet type SPLIT.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:HUB:VALue <QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:HUB:VALue?
```

Arguments `<QString>` specifies the hub address. The valid characters are .

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:HUB:VALUE "1001"` sets the hub address to "XXX1001".

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:HUB:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:HUB:VALUE "XX"` (for binary format), indicating that the hub address for the USB bus trigger search on split token field is "XX".

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:PORT:VALue

This command sets or queries the binary port address used when searching on a USB bus signal. The search number is specified by x. The search condition must be set to Special with a packet type SPLIT.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:PORT:VALue
<QString>
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:PORT:VALue?
```

Arguments `<QString>` specifies the port address in the valid format. The valid characters are

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:PORT:VALUE "1111"` sets the port address for USB bus trigger search 1 to "XXX1111" for binary format.

`SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:PORT:VALUE?` might return `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:PORT:VALUE "XX"`, indicating that the port address for USB hub trigger search 1 is set to "XX" in hexadecimal format.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SC:VALue

This command sets or queries the Start/Complete value for the specified USB bus trigger on split token field search to determine where to place a mark. The search number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SC:VALue
{CSPLIT|NOCARE|SSPLIT}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SC:VALue?
```

Arguments `CSPLIT` specifies Complete (CSPLIT)(1) Start/Complete value.

`NOCARE` specifies the Start/Complete value as Don't Care (X).

`SSPLIT` specifies Start (SSPLIT) (0) Start/Complete value.

Examples `SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:SC:VALUE DONTCARE` sets the Start/Complete value for USB bus trigger search 1 to DONTCare.

`SEARCH:SEARCH2:TRIGGER:A:BUS:USB:SPLIT:SC:VALUE?` might return `:SEARCH:SEARCH2:TRIGGER:A:BUS:USB:SPLIT:SC:VALUE SSPLIT`, indicating that the Start/Complete value for USB bus trigger search 2 is set to SSPLIT.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SE:VALue

This command sets or queries the Start/End value for the specified USB bus trigger on split token field search to determine where to place a mark. The search number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SE:VALue
{NOCARE|FULLSPEED|ISOALL|ISOEND|ISOMID|ISOSTART|LOWSPEED}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SE:VALue?
```

Arguments `NOCARE` specifies Don't Care (X) Start/End bit value.

`FULLSPEED` specifies Control/Bulk/Interrupt Full Speed device (0X) Start/End bit value.

`ISOALL` specifies Isochronous Data is All (11) Start/End bit value.

`ISOEND` specifies Isochronous Data is End (01) Start/End bit value.

`ISOMID` specifies Isochronous Data is Middle (00) Start/End bit value.

ISOSTART specifies Isochronous Data is Start (10) Start/End bit value.

LOWSPEED specifies Control/Bulk/Interrupt Low Speed device (1X) Start/End bit value.

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:SE:VALUE ISOSTART sets the Start/End value for USB bus trigger on split token field search 1 to ISOSTART.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:SE:VALUE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:SPLIT:SE:VALUE DONTCARE, indicating that the Start/End value for USB bus trigger on split token field search 1 is set to DONTCare.

SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TOKENType

This command sets or queries the token type used to search a USB bus signal. The search number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TOKENType
{ANY|IN|OUT|SETUP|SOF}
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TOKENType?
```

Arguments

ANY specifies the token type as Any (XX01).

IN specifies the token type as IN (1001).

OUT specifies the token type as OUT (0001).

SETUP specifies the token type as SETUP (1101).

SOF specifies the token type as Start Of Frame (0101).

Examples

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:TOKENTYPE IN sets the token type for USB bus trigger search 1 to IN.

SEARCH:SEARCH1:TRIGGER:A:BUS:USB:TOKENTYPE? might return SEARCH:SEARCH1:TRIGGER:A:BUS:USB:TOKENTYPE SOF, indicating that the token type for USB bus trigger search 1 is set to Start Of Frame.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

This command sets or queries the DDRRead search burst detection method.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod
{DQDQS|ChipSelect|LogicState}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard

Arguments `DQDQS` sets the burst detection method to DQDQS.

`ChipSelect` sets the burst detection method to ChipSelect.

`LogicState` sets the burst detection method to LogicState.

Examples `SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTDETectmethod DQDQS` sets the burst detection method as DQ/DQS Phase alignment.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTDETectmethod?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTDETectmethod DQDQS` indicating that the DDR read burst detection method is DQDQS.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLatency

This command sets or queries the DDR read logic state burst latency value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLatency <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLatency?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`NR3` sets the logic state burst latency value.

Examples `SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTLatency 500E-3` sets the DDRREAD burst latency value of search 1 to .005.

`SEARCH:SEARCH3:TRIGger:A:DDRREAD:BURSTLatency?` might return `SEARCH:SEARCH3:TRIGger:A:DDRREAD:BURSTLatency 500E-3`, indicating that the DDRREAD logic burst latency value is set to 500E-3 for search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLENGTH

This command sets or queries the DDR read logic state burst length value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLENGTH <NR1>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLENGTH?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`NR1` sets the logic state burst length value.

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREAD:BURSTLENGTH 10` sets the DDRREAD logic state burst length value of search 3 to 10 bursts.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTLENGTH?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:BURSTLENGTH 8`, indicating that the DDRREAD logic state burst length value is set to 8 bursts for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSActive

This command sets or queries the DDR Read chip select active state.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSActive {LOW|HIGH}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSActive?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments

`SEARCH<x>` specifies the search number.

`LOW` sets the DDR read chip select active state to low.

`HIGH` sets the DDR read chip select active state to high.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:CSActive LOW` sets the DDRREAD chip select active to LOW for search 2.

`SEARCH:SEARCH4:TRIGger:A:DDRREAD:CSActive?` might return `SEARCH:SEARCH4:TRIGger:A:DDRREAD:CSActive HIGH`, indicating the DDRREAD chip select active value for search 4 is set to HIGH.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSLevel

This command sets or queries the DDR Read chip select level value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSLevel <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSLevel?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode

Arguments `SEARCH<x>` specifies the search number.

`NR3` sets the DDR read chip select level value.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRREAD:CSLevel 2.8E-3` sets the DDRREAD chip select level value to 2.8E-3

`SEARCH:SEARCH5:TRIGger:A:DDRREAD:CSLevel?` might return `SEARCH:SEARCH5:TRIGger:A:DDRREAD:CSLevel 2.8E-3`, indicating DDRREAD search chip select level value for search 5 is set to 2.8E-3

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode

This command sets or queries the DDR Read chip select mode.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode {Auto|Manual}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

Auto sets the DDR read chip select mode to auto.

Manual sets the DDR read chip select mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:CSMode MANUAL sets the DDRREAD search chip select mode to Manual for search 2.

SEARCH:SEARCH3:TRIGger:A:DDRREAD:CSMode? might return SEARCH:SEARCH3:TRIGger:A:DDRREAD:CSMode Manual, indicating that the DDRREAD search 3 chip select mode is set to manual.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSSource

This command sets or queries the DDR Read search chip select source.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSSource
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSSource?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

CH<x> specifies channel <x> as the DDR read chip select source for the specified search <x>.

CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR read strobe source for the specified search <x>.

Math<x> specifies math waveform <x> as the DDR read chip select source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR read chip select source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read strobe source for the specified search <x>.

Examples

SEARCH:SEARCH5:TRIGger:A:DDRREAD:CSSource CH2 sets the search 5 chip select source waveform to CH2 for DDR Read search.

SEARCH:SEARCH3:TRIGger:A:DDRREAD:CSSource? might return SEARCH:SEARCH3:TRIGger:A:DDRREAD:CSSource CH2, indicating that Channel 2 waveform is the source for the DDRRead chip select source of search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATARate

This command sets or queries the DDR read search data rate for DDR3 and LPDRR3 standards of the specified search.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATARate
{333|800|1066|1200|1333|1466|1600|1866|2133}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATARate?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard

Arguments

333–2133 sets the data rate to the specified value.

The DDR3 standard supports the following data rates: 800|1066|1333|1600|1866|2133.

The LPDDR3 standard supports the following data rates: 333|800|1066|1200|1333|1466|1600|1866|2133.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRREAD:DATARate 1333` sets the data rate to 1333 for DDR read Search 2.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:DATARate?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:DATARate 800` indicating that the data rate for DDR read Search 1 is 800.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATASource

This command sets or queries the DDR read data source when the search type is DDR READ.

Conditions Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATASource
{CH<x>|CH<x>_D<x<|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATASource?
```

Arguments `CH<x>` specifies channel <x> as the DDR read data source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR read data source for the specified search <x>.

`Math<x>` specifies math waveform <x> as the DDR read data source for the specified search <x>.

`REF<x>` specifies reference waveform <x> as the DDR read data source for the specified search <x>.

`REF<x>_D<x>` specifies digital waveform <x> of reference waveform <x> as the DDR read data source for the specified search <x>.

Examples `SEARCH:SEARCH1:TRIGGER:A:DDRREAD:DATASource CH2` sets the source waveform for the A trigger to Channel 2.

`SEARCH:SEARCH1:TRIGGER:A:DDRREAD:DATASource?` might return `SEARCH:SEARCH1:TRIGGER:A:DDRREAD:DATASource REF1`, indicating that the Reference waveform 1 DDR read trigger data is the source for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis

This command sets or queries the DDR read hysteresis reference level value, when the search type is DDR READ.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments NR3 sets the DDR read search hysteresis percent value in the range of 0% to 50%.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRREAD:HYSteresis 15` sets the DDR read search hysteresis value to 15%.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:HYSteresis?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:HYSteresis 10`, indicating that the DDR read search hysteresis value is set to 10%.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol

This command sets or queries the DDR read logic source 1 symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 1 symbol to H (High).

L sets the logic source 1 symbol to L (Low).

X sets the logic source 1 symbol to X (Don't care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol X sets the DDRREAD logic source 1 symbol of search 3 to X (don't care).

SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol L, indicating that the DDRREAD logic source 1 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol

This command sets or queries the DDR read logic source 2 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 2 symbol to H (High).

L sets the logic source 2 symbol to L (Low).

X sets the logic source 2 symbol to X (Don't care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol X sets the DDRREAD logic source 2 symbol of search 3 to X (don't care)

SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol L, indicating that the DDRREAD logic source 2 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol

This command sets or queries the DDR read logic source 3 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 3 symbol to H (High).

L sets the logic source 3 symbol to L (Low).

X sets the logic source 3 symbol to X (Don't care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol X sets the DDRREAD logic source 3 symbol of search 3 to X (don't care)

SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol L, indicating that the DDRREAD logic source 3 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol

This command sets or queries the DDR read logic source 4 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 4 symbol to H (High).

L sets the logic source 4 symbol to L (Low).

X sets the logic source 4 symbol to X (Don’t care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol X sets the DDRREAD logic source 4 symbol of search 3 to X (don’t care)

SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol L, indicating that the DDRREAD logic source 4 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin

This command sets or queries the DDR read margin reference level value, when the search type is DDR READ.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments NR3 sets the DDR read search margin percent value in the range of 0% to 100%.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRREAD:MARgin 12` sets the DDR read search margin value to 12%.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:MARgin?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:MARgin 10`, indicating that the DDR read search margin value is set to 10%.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MAXCAS

This command sets or queries the DDR read chip maximum Column Access Strobe (CAS) value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MAXCAS <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MAXCAS?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

NR3 sets the DDR read chip select CAS maximum value.

Examples

EARCH:SEARCH1:TRIGger:A:DDRREAD:MAXCas 3.8E-3 sets the DDR read chip select CAS maximum value to 3.8 milliseconds for search 1.

SEARCH:SEARCH7:TRIGger:A:DDRREAD:MAXCas? might return SEARCH:SEARCH7:TRIGger:A:DDRREAD:MAXCas 3.8E-3, indicating that the DDR read chip select CAS maximum value is set to 3.8 milliseconds for search 7.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MINCas

This command sets or queries the DDR read chip minimum Column Access Strobe (CAS) value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MINCas <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MINCas?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

NR3 sets the DDR read chip select CAS minimum value.

Examples

EARCH:SEARCH1:TRIGger:A:DDRREAD:MINCas 3.8E-3 sets the DDR read chip select CAS minimum value to 3.8 milliseconds for search 1.

SEARCH:SEARCH7:TRIGger:A:DDRREAD:MINCas? might return SEARCH:SEARCH7:TRIGger:A:DDRREAD:MINCas 3.8E-3, indicating that the DDR read chip select CAS minimum value is set to 3.8 milliseconds for search 7.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:POSTAMBLE:LENGth

This command sets or queries the DDR read postamble length when the search type is DDR READ.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:POSTAMBLE:LENGth
{500E-3|500E-1.5|500E-1}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:POSTAMBLE:LENGth?
```

Arguments `500E-1.5` sets the postamble length to 1.5 tCK (number of clock cycles).

`500E-1` sets the postamble length to 1 tCK (number of clock cycles).

`500E-3` sets the postamble length to 0.5 tCK (number of clock cycles).

Examples `SEARCH:SEARCH2:TRIGger:A:DDRREAD:POSTAMBLE:LENGth 500E-3` sets the DDR read postamble length for Search 2 to 0.5 tCK.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:POSTAMBLE:LENGth?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:POSTAMBLE:LENGth 500E-1`, indicating that the read postamble length for Search 1 is set to 1 tCK.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE

This command sets or queries the DDR read preamble type when the search type is DDR READ.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE
{STATIC|DYNAMIC}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE?
```

Arguments

`STATIC` sets the DDR Read preamble type to Static.

`DYNAMIC` sets the DDR Read preamble type to Dynamic.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:PREAMBLE:TYPE STATIC` sets the DDR read preamble type to Static for Search 2.

`SEARCH:SEARCH4:TRIGger:A:DDRREAD:PREAMBLE:TYPE?` might return `SEARCH:SEARCH4:TRIGger:A:DDRREAD:PREAMBLE:TYPE DYNAMIC`, indicating that the DDR read preamble type for Search 4 is set to dynamic mode.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH

This command sets or queries the DDR read data high reference level value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments

`NR3` sets the DDR read search data high reference value.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH 3.8E-3` sets the DDR read search data high reference value.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH 50E-3`, indicating that the DDR read search data high reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW

This command sets or queries the DDR read data low reference level value, when the search type is DDR READ.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments `NR3` sets the DDR read search data low reference value.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:DATA:LOW 3.8E-3` sets the DDR read search data low reference value.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:LOW?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:LOW 50E-3`, indicating that the DDR read search data low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID

This command sets or queries the DDR read data mid reference level value, when the search type is DDR READ.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments

NR3 sets the DDR read search data mid reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:DATA:MID 3.8E-3 sets the DDR read search data mid reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:MID? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:DATA:MID 50E-3, indicating that the DDR read search data mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH

This command sets or queries the DDR read reference level strobe high value, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH
<NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments

NR3 sets the DDR read search strobe high value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH 3.8E-3 sets the DDR read search strobe high value.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH 50E-3, indicating that the DDR read search strobe high value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW

This command sets or queries the DDR read reference level strobe low value, when the search type is DDR READ.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments NR3 sets the DDR read search strobe low reference value.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW 3.8E-3` sets the DDR read search strobe low reference value.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:MID 50E-3`, indicating that the DDR read search strobe low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:MID

This command sets or queries the DDR read reference level strobe mid value, when the search type is DDR READ.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:MID <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:MID?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

Arguments

NR3 sets the DDR read search strobe mid reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREAD:REFLevel:STROBE:MID 3.8E-3 sets the DDR read search strobe mid reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:MID? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLevel:STROBE:MID 50E-3, indicating that the DDR read search strobe mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode

This command sets or queries the DDR read reference level mode to auto or manual, when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode
{AUTO|MANUAL}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID

Arguments

AUTO sets the DDR read reference level mode to auto.

MANUAL sets the DDR read reference level mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREAD:REFLEVELMode MANUAL` sets the DDR read search reference level mode to Manual for Search 3.

`SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLEVELMode?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREAD:REFLEVELMode MANUAL`, indicating that the DDR read search reference level mode set to automatic for Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard

This command sets or queries the DDR read search standard as DDR3 or LPDDR3.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard {DDR3|LPDDR3}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard?
```

Arguments

`DDR3` sets the DDR read search standard as DDR3.

`LPDDR3` sets the DDR read search standard as LPDDR3.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREAD:STANdard DDR3` sets the DDR standard to DDR3 for DDR read Search 2.

`SEARCH:SEARCH5:TRIGger:A:DDRREAD:STANdard?` might return `SEARCH:SEARCH5:TRIGger:A:DDRREAD:STANdard LPDDR3`, indicating that the DDR standard is set to LPDDR3 for DDR read Search 5.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource

This command sets or queries the DDR read strobe source when the search type is DDR READ.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource
{CH<x>|CH<x>_D<x<|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource?
```

Arguments `CH<x>` specifies channel <x> as the DDR read strobe trigger source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR read strobe trigger source for the specified search <x>.

`Math<x>` specifies math waveform <x> as the DDR read strobe trigger source for the specified search <x>.

`REF<x>` specifies reference waveform <x> as the DDR read strobe trigger source for the specified search <x>.

`REF<x>_D<x>` specifies digital waveform <x> of reference waveform <x> as the DDR read strobe trigger source for the specified search <x>.

Examples `SEARCH:SEARCH3:TRIGGER:A:DDRREAD:STROBESource MAT25` sets the Search 3 source waveform to Math 2 for the DDR read strobe search.

`SEARCH:SEARCH1:TRIGGER:A:DDRREAD:STROBESource?` might return `SEARCH:SEARCH1:TRIGGER:A:DDRREAD:STROBESource CH3`, indicating that the Channel 3 waveform is the source for the DDR read strobe of Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:TOLERance

This command sets or queries the DDR read logic state tolerance value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:TOLERance <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:TOLERance?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod

Arguments SEARCH<x> specifies the search number.

NR3 sets the logic state tolerance value.

Examples SEARCH:SEARCH3:TRIGger:A:DDRREAD:TOLERance 400E-3 sets the DDRREAD logic state tolerance value of search 3 to .004.

SEARCH:SEARCH1:TRIGger:A:DDRREAD:TOLERance? might return SEARCH:SEARCH1:TRIGger:A:DDRREAD:TOLERance 500E-3, indicating that the DDRREAD logic state tolerance value is set to 500E-3 for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

This command sets or queries the DDR Read/Write search burst detection method.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod
{DQDQS|ChipSelect|LogicState}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard

Arguments DQDQS sets the read/write burst detection method to DQDQS.

ChipSelect sets the read/write burst detection method to ChipSelect.

LogicState sets the read/write burst detection method to LogicState.

Examples SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTDETectmethod DQDQS sets the burst detection method as Logic state + burst latency

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTDETectmethod? might return
SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

LogicState, indicating that the DDR ReadWrite burst detection method is Logic state + burst latency.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLatency

This command sets or queries the DDR read/write logic state burst latency value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLatency <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLatency?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments SEARCH<x> specifies the search number.

NR3 sets the logic state burst latency value.

Examples SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTLatency 500E-3 sets the DDRREADWRITE burst latency value of search 1 to .005.

SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:BURSTLatency? might return SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:BURSTLatency 500E-3, indicating that the DDRREADWRITE logic burst latency value is set to 500E-3 for search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLENGTH

This command sets or queries the DDR read/write logic state burst length value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLENGTH <NR1>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLENGTH?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`NR1` sets the logic state burst length value.

Examples `SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:BURSTLENGTH 10` sets the DDRREADWRITE logic state burst length value of search 3 to 10 bursts.

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTLENGTH?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:BURSTLENGTH 8`, indicating that the DDRREADWRITE logic state burst length value is set to 8 bursts for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSActive

This command sets or queries the DDR Read/Write chip select active state.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSActive {LOW|HIGH}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSActive?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

LOW sets the DDR read/write chip select active state to low.

HIGH sets the DDR read/write chip select active state to high.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:CSActive LOW sets the DDRREADWRITE chip select active to LOW.

SEARCH:SEARCH4:TRIGger:A:DDRREADWRITE:CSActive? might return SEARCH:SEARCH4:TRIGger:A:DDRREADWRITE:CSActive HIGH, indicating the DDRREADWRITE search chip select active value for search 4 is set to HIGH.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSLevel

This command sets or queries the DDR Read/Write chip select level value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSLevel <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSLevel?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode

Arguments

SEARCH<x> specifies the search number.

NR3 sets the DDR read/write chip select level value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:CSLevel 2.8E-3 sets the DDR read/write chip select level value to 2.8E-3.

SEARCH:SEARCH4:TRIGger:A:DDRREADWRITE:CSLevel? might return SEARCH:SEARCH4:TRIGger:A:DDRREADWRITE:CSLevel 2.8E-3, indicating the DDR read/write search chip select level value for search 4 is set to 2.8E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode

This command sets or queries the DDR read/write chip select mode.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode {Auto|Manual}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`Auto` sets the DDR read/write chip select mode to auto.

`Manual` sets the DDR read/write chip select mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples `SEARCH:SEARCH4:TRIGger:A:DDRREADWRITE:CSMode AUTO` sets the DDR read/write search chip select mode to Auto for search 4.

`SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:CSMode?` might return `SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:CSMode Manual`, indicating that the DDR read/write chip select mode is set to manual for search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSSource

This command sets or queries the DDR Read/Write search chip select source.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select".

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSSource
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSSource?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

CH<x> specifies channel <x> as the DDR read/write chip select source for the specified search <x>.

CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR read/write strobe source for the specified search <x>.

Math<x> specifies math waveform <x> as the DDR read/write chip select source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR read/write chip select source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read/write strobe source for the specified search <x>.

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:CSSource CH2 sets the search 3 chip select source waveform to CH2 for the specified DDR read/write search.

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:CSSource? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:CSSource CH2, indicating that the Channel 2 waveform is the source for the DDR read/write chip select source of search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate

This command sets or queries the DDR read/write data rate for DDR3 and LPDRR3 standards of the specified search.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read/Write standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate
{333|800|1066|1200|1333|1466|1600|1866|2133}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard

Arguments `333–2133` sets the data rate to the specified value.

The DDR3 standard supports the following data rates: 800|1066|1333|1600|1866|2133.

The LPDDR3 standard supports the following data rates: 333|800|1066|1200|1333|1466|1600|1866|2133.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:DATARate 1333` sets the data rate to 1333 for DDR read Search 2.

`SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:DATARate?` might return `SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:DATARate 1466` indicating that the data rate for DDR read Search 5 is 1466.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource

This command sets or queries the DDR read/write data source when the search type is DDR READWRITE.

Conditions Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource
{CH<x>|CH<x>_D<x<|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource?
```

Arguments `CH<x>` specifies channel <x> as the DDR read/write data source for the specified search <x>.

CH<x>_D<x> specifies digital waveform <x> of channel <x> as the DDR read/write data source for the specified search <x>.

Math<x> specifies math waveform <x> as the DDR read/write data source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR read/write data source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR read/write data source for the specified search <x>.

Examples

SEARCH:SEARCH3:TRIGGER:A:DDRREADWRITE:DATASource CH2 sets the source waveform of Search 3 to Channel 2 for the DDR read/write search.

SEARCH:SEARCH1:TRIGGER:A:DDRREADWRITE:DATASource? might return SEARCH:SEARCH1:TRIGGER:A:DDRREADWRITE:DATASource CH2, indicating that the Channel 2 waveform is the source for the DDR read/write data of Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis

This command sets or queries the DDR read/write hysteresis reference level value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments

NR3 sets the DDR read/write search hysteresis percent value in the range of 0% to 50%.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:HYSteresis 10 sets the DDR read/write search hysteresis value to 10%.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:HYSteresis? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:HYSteresis 5, indicating that the DDR read/write search hysteresis value is set to 5%.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol

This command sets or queries the DDR read/write logic source 1 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 1 symbol to H (High).

L sets the logic source 1 symbol to L (Low).

X sets the logic source 1 symbol to X (Don't care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol X sets the DDRREADWRITE logic source 1 symbol of search 3 to X (don't care)

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol L, indicating that the DDRREADWRITE logic source 1 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol

This command sets or queries the DDR read/write logic source 2 symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`H` sets the logic source 2 symbol to H (High).

`L` sets the logic source 2 symbol to L (Low).

`X` sets the logic source 2 symbol to X (Don't care).

Examples `SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol X` sets the DDRREADWRITE logic source 2 symbol of search 3 to X (don't care)

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol L`, indicating that the DDRREADWRITE logic source 2 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol

This command sets or queries the DDR read/write logic source 3 symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`H` sets the logic source 3 symbol to H (High).

`L` sets the logic source 3 symbol to L (Low).

`X` sets the logic source 3 symbol to X (Don't care).

Examples `SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol X` sets the DDRREADWRITE logic source 3 symbol of search 3 to X (don't care)

`SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol L`, indicating that the DDRREADWRITE logic source 3 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol

This command sets or queries the DDR read/write logic source 4 symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

H sets the logic source 4 symbol to H (High).

L sets the logic source 4 symbol to L (Low).

X sets the logic source 4 symbol to X (Don't care).

Examples

SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol X sets the DDRREADWRITE logic source 4 symbol of search 3 to X (don't care)

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol L, indicating that the DDRREADWRITE logic source 4 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin

This command sets or queries the DDR read/write margin reference level value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments NR3 sets the DDR read/write search margin percent value in the range of 0% to 100%.

Examples SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:MARgin 15 sets the DDR read/write search margin value to 15%.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:MARgin? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:MARgin 10, indicating that the DDR read/write search margin value is set to 10%.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MAXCAS

This command sets or queries the DDR read/write chip maximum Column Access Strobe (CAS) value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MAXCAS <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MAXCAS?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments SEARCH<x> specifies the search number.

NR3 sets the DDR read chip select CAS maximum value.

Examples EARCH:SEARCH1:TRIGger:A:DDRREADWRITE:MAXCAS 3.8E-3 sets the DDR read chip select CAS maximum value to 3.8 milliseconds for search 1.

SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:MAXCAS? might return SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:MAXCAS 2.1E-3, indicating that the DDR read chip select CAS maximum value is set to 2.1 milliseconds for search 5.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MINCas

This command sets or queries the DDR read/write chip minimum Column Access Strobe (CAS) value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MINCas <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MINCas?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`NR3` sets the DDR read chip select CAS minimum value.

Examples `EARCH:SEARCH1:TRIGger:A:DDRREADWRITE:MINCas 3.8E-3` sets the DDR read chip select CAS minimum value to 3.8 milliseconds for search 1.

`SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:MINCas?` might return `SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:MINCas 2.1E-3`, indicating that the DDR read chip select CAS minimum value is set to 2.1 milliseconds for search 5.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth

This command sets or queries the DDR read/write postamble length when the search type is DDR READWRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth
{500E-3|1.5|1}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth?
```

Arguments

`500E-1.5` sets the postamble length to 1.5 tCK (number of clock cycles).

`500E-1` sets the postamble length to 1 tCK (number of clock cycles).

`500E-3` sets the postamble length to 0.5 tCK (number of clock cycles).

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth 500E-1.5` sets the DDR read/write postamble length for Search 3 to 1.5 tCK.

`SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth?` might return `SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth 500E-1`, indicating that the read/write postamble length for Search 2 is set to 1 tCK.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE

This command sets or queries the DDR read/write preamble type when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE
{STATIC|DYNAMIC}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE?
```

Arguments

`STATIC` sets the DDR Read/Write preamble type to Static.

`DYNAMIC` sets the DDR Read/Write preamble type to Dynamic.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE STATIC` sets the search 2 DDR read preamble type to Static.

`SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE?` might return `SEARCH:SEARCH5:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE DYNAMIC`, indicating that the DDR read/write preamble type for Search 5 is set to dynamic mode.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH

This command sets or queries the DDR read/write data high reference level value, when the search type is DDR READWRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH
<NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments `NR3` sets the DDR read/write search data high reference value.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH 3.8E-3` sets the DDR read/write search data high reference value.

`SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:REFLevel:DATA:HIGH?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH 50E-3`, indicating that the DDR read/write search data high reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW

This command sets or queries the DDR read/write data low reference level value, when the search type is DDR READWRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW
<NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments NR3 sets the DDR read/write search data low reference value.

Examples SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW 3.8E-3 sets the DDR read/write search data low reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:REFLevel:DATA:LOW? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW 50E-3, indicating that the DDR read/write search data low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID

This command sets or queries the DDR read/write data mid reference level value, when the search type is DDR READWRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID
<NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments NR3 sets the DDR read/write search data mid reference value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID 3.8E-3 sets the DDR read/write search data mid reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:REFLevel:DATA:MID? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID 50E-3, indicating that the DDR read/write search data mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH

This command sets or queries the DDR read/write reference level strobe high value, when the search type is DDR READWRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments

NR3 sets the DDR read search strobe high value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH 3.8E-3 sets the DDR read/write search strobe high value.

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH 50E-3, indicating that the DDR read/write search strobe high value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW

This command sets or queries the DDR read/write reference level strobe low value, when the search type is DDR READWRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments NR3 sets the DDR read/write search strobe low reference value.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW 3.8E-3` sets the DDR read/write search strobe low reference value.

`SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:REFLevel:STROBE:LOW?` might return `SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW 50E-3`, indicating that the DDR read/write search strobe low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID

This command sets or queries the DDR read/write reference level strobe mid value, when the search type is DDR READWRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR read/write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

Arguments NR3 sets the DDR read/write search strobe mid reference value.

Examples SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID 3.8E-3 sets the DDR read/write search strobe mid reference value.

SEARCH:SEARCH1:TRIGger:A:DDRREAWRITE:REFLevel:STROBE:MID? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID 50E-3, indicating that the DDR read/write search strobe mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode

This command sets or queries the DDR read/write reference level mode to auto or manual, when the search type is DDR READWRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode {AUTO|MANUAL}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID

Arguments AUTO sets the DDR read/write reference level mode to auto.

MANUAL sets the DDR read/write reference level mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:REFLEVELMode MANUAL sets the DDR read/write search reference level mode to Manual for Search 3.

SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLEVELMode? might return SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:REFLEVELMode MANUAL, indicating that the DDR read/write search reference level mode set to automatic for Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard

This command sets or queries the DDR read/write search standard as DDR3 or LPDDR3.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard
{DDR3|LPDDR3}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard?
```

Arguments DDR3 sets the DDR read/write search standard as DDR3.

LPDDR3 sets the DDR read/write search standard as LPDDR3.

Examples SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:STANdard DDR3 sets the DDR standard to DDR3 for DDR read/write Search 2.

SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:STANdard? might return SEARCH:SEARCH2:TRIGger:A:DDRREADWRITE:STANdard LPDDR3 indicating that the DDR standard is set to LPDDR3 for DDR read/write Search 2.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STROBESource

This command sets or queries the DDR readwrite strobe source when the search type is DDR READWRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STROBESource
{CH<x>|CH<x>_D<x<|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STROBESource?
```

Arguments `CH<x>` specifies channel <x> as the DDR read/write strobe source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR read/write strobe source for the specified search <x>.

`Math<x>` specifies math waveform <x> as the DDR read/write strobe source for the specified search <x>.

`REF<x>` specifies reference waveform <x> as the DDR read/write strobe source for the specified search <x>.

`REF<x>_D<x>` specifies digital waveform <x> of reference waveform <x> as the DDR read/write strobe source for the specified search <x>.

Examples `SEARCH:SEARCH5:TRIGGER:A:DDRREADWRITE:STROBESource CH8` sets the Search 5 source waveform to Channel 8 for the DDR read/write strobe search.

`SEARCH:SEARCH1:TRIGGER:A:DDRREADWRITE:STROBESource?` might return `SEARCH:SEARCH1:TRIGGER:A:DDRREADWRITE:STROBESource CH3`, indicating that the Channel 3 waveform is the source for the DDR read/write strobe of Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:TOLERance

This command sets or queries the DDR read/write logic state tolerance value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:TOLERance <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:TOLERance?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod

Arguments SEARCH<x> specifies the search number.

NR3 sets the logic state tolerance value.

Examples SEARCH:SEARCH1:TRIGger:A:DDRREADWRITE:TOLERance 400E-3 sets the DDRREADWRITE logic state tolerance value of search 1 to .004.

SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:TOLERance? might return SEARCH:SEARCH3:TRIGger:A:DDRREADWRITE:TOLERance 500E-3, indicating that the DDRREADWRITE logic state tolerance value is set to 500E-3 for search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

This command sets or queries the DDRWRITE search burst detection method.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod
{DQDQS|ChipSelect|LogicState}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard

Arguments DQDQS sets the write burst detection method to DQDQS.

ChipSelect sets the write burst detection method to ChipSelect.

LogicState sets the write burst detection method to LogicState.

Examples SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTDETectmethod ChipSelect sets the burst detection method as ChipSelect, Latency + DQ/DQS Phase alignment.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTDETectmethod? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTDETectmethod ChipSelect, indicating that the DDR Write burst detection method is ChipSelect.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLatency

This command sets or queries the DDR write logic state burst latency value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLatency <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLatency?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

NR3 sets the logic state burst latency value.

Examples

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTLatency 500E-3 sets the burst latency value of search 1 to .005.

SEARCH:SEARCH3:TRIGger:A:DDRWRITE:BURSTLatency? might return SEARCH:SEARCH3:TRIGger:A:DDRWRITE:BURSTLatency 500E-3, indicating that the DDRWRITE logic burst latency value is set to 500E-3 for search 3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLENGTH

This command sets or queries the DDR write logic state burst length value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLENGTH <NR1>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLENGTH?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`NR1` sets the logic state burst length value.

Examples `SEARCH:SEARCH3:TRIGger:A:DDRWRITE:BURSTLENGTH 10` sets the DDRWRITE logic state burst length value of search 3 to 10 bursts.

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTLENGTH?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:BURSTLENGTH 8`, indicating that the DDRWRITE logic state burst length value is set to 8 bursts for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSActive

This command sets or queries the DDR write chip select active state.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSActive {LOW|HIGH}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSActive?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments

SEARCH<x> specifies the search number.

LOW sets the DDR write chip select active state to low.

HIGH sets the DDR write chip select active state to high.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:CSActive LOW sets the DDRWRITE chip select active state to LOW for search 2.

SEARCH:SEARCH4:TRIGger:A:DDRWRITE:CSActive? might return SEARCH:SEARCH4:TRIGger:A:DDRWRITE:CSActive HIGH, indicating the DDRWRITE chip select active state for search 4 is set to HIGH.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSLevel

This command sets or queries the DDR Write chip select level value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSLevel <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSLevel?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode

Arguments

SEARCH<x> specifies the search number.

NR3 sets the DDR write chip select level value.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:CSLevel 2.8E-3 sets the DDR write chip select level value to 2.8E-3

SEARCH:SEARCH5:TRIGger:A:DDRWRITE:CSLevel? might return SEARCH:SEARCH5:TRIGger:A:DDRWRITE:CSLevel 2.8E-3, indicating the DDR write search chip select level value for search 5 is set to 2.8E-3

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode

This command sets or queries the DDR Write chip select mode.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select"

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode {Auto|Manual}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`Auto` sets the DDR write chip select mode to auto.

`Manual` sets the DDR write chip select mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRWRITE:CSMode MANUAL` sets the DDR write search chip select mode to Manual for search 2.

`SEARCH:SEARCH3:TRIGger:A:DDRWRITE:CSMode?` might return `SEARCH:SEARCH3:TRIGger:A:DDRWRITE:CSMode Manual`, indicating that the DDR write search 3 chip select mode is set to manual.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSSource

This command sets or queries the DDR Write search chip select source.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSSource
{CH<x>|CH<x>_D<x>|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSSource?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments

`SEARCH<x>` specifies the search number.

`CH<x>` specifies channel <x> as the DDR write chip select source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR write strobe source for the specified search <x>.

`Math<x>` specifies math waveform <x> as the DDR write chip select source for the specified search <x>.

`REF<x>` specifies reference waveform <x> as the DDR write chip select source for the specified search <x>.

`REF<x>_D<x>` specifies digital waveform <x> of reference waveform <x> as the DDR write strobe source for the specified search <x>.

Examples

`SEARCH:SEARCH5:TRIGger:A:DDRWRITE:CSSource CH2` sets the search 5 chip select source waveform to CH2 for DDR write search.

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:CSSource?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:CSSource CH2`, indicating that the Channel 2 waveform is the source for the DDR write chip select source of search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATARate

This command sets or queries the DDR write search data rate for DDR3 and LPDRR3 standards of the specified search.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Write standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATARate
{333|800|1066|1200|1333|1466|1600|1866|2133}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATARate?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard

Arguments `333–2133` sets the data rate to the specified value.

The DDR3 standard supports the following data rates: 800|1066|1333|1600|1866|2133.

The LPDDR3 standard supports the following data rates: 333|800|1066|1200|1333|1466|1600|1866|2133.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRWRITE:DATARate 1333` sets the data rate to 1333 for DDR read Search 2.

`SEARCH:SEARCH7:TRIGger:A:DDRWRITE:DATARate?` might return `SEARCH:SEARCH7:TRIGger:A:DDRWRITE:DATARate 1466` indicating that the data rate for DDR read Search 7 is 1466.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATASource

This command sets or queries the DDR write data source when the search type is DDR Write.

Conditions Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATASource
{CH<x>|CH<x>_D<x<|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATASource?
```

Arguments `CH<x>` specifies channel <x> as the DDR write data source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR write data source for the specified search <x>.

Math<x> specifies math waveform <x> as the DDR write data source for the specified search <x>.

REF<x> specifies reference waveform <x> as the DDR write data source for the specified search <x>.

REF<x>_D<x> specifies digital waveform <x> of reference waveform <x> as the DDR write data source for the specified search <x>.

Examples

SEARCH:SEARCH1:TRIGGER:A:WRITE:DATASource CH2 sets the search 1 source waveform to Channel 2 for the DDR write search.

SEARCH:SEARCH1:TRIGGER:A:DDRWRITE:DATASource? might return SEARCH:SEARCH1:TRIGGER:A:DDRWRITE:DATASource RMATH3, indicating that the Math 1 waveform DDR write trigger data is the source for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis

This command sets or queries the DDR write hysteresis reference level value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

Arguments

NR3 sets the DDR write search hysteresis percent value in the range of 0% to 50%.

Examples

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:HYSteresis 7 sets the DDR write search hysteresis value to 7%.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:HYSteresis? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:HYSteresis 10, indicating that the DDR write search hysteresis value is set to 10%.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol

This command sets or queries the DDR write logic source 1 symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`H` sets the logic source 1 symbol to H (High).

`L` sets the logic source 1 symbol to L (Low).

`X` sets the logic source 1 symbol to X (Don't care).

Examples `SEARCH:SEARCH3:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol X` sets the DDRWRITE logic source 1 symbol of search 3 to X (don't care)

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol L`, indicating that the DDRWRITE logic source 1 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol

This command sets or queries the DDR write logic source 2 symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments

`SEARCH<x>` specifies the search number.

`H` sets the logic source 2 symbol to H (High).

`L` sets the logic source 2 symbol to L (Low).

`X` sets the logic source 2 symbol to X (Don't care).

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol X` sets the DDRWRITE logic source 2 symbol of search 3 to X (don't care)

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol L`, indicating that the DDRWRITE logic source 2 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol

This command sets or queries the DDR write logic source 3 symbol.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments SEARCH<x> specifies the search number.

H sets the logic source 3 symbol to H (High).

L sets the logic source 3 symbol to L (Low).

X sets the logic source 3 symbol to X (Don't care).

Examples SEARCH:SEARCH3:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol X sets the DDRWRITE logic source 3 symbol of search 3 to X (don't care)

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol L, indicating that the DDRWRITE logic source 3 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol

This command sets or queries the DDR write logic source 4 symbol.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments SEARCH<x> specifies the search number.

H sets the logic source 4 symbol to H (High).

L sets the logic source 4 symbol to L (Low).

X sets the logic source 4 symbol to X (Don't care).

Examples

`SEARCH:SEARCH3:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol X` sets the DDRWRITE logic source 4 symbol of search 3 to X (don't care)

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol L`, indicating that the DDRWRITE logic source 4 symbol is set to Low for search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin

This command sets or queries the DDR write margin reference level value, when the search type is DDR WRITE.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin?
```

Related Commands

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

Arguments

`NR3` sets the DDR write search margin percent value in the range of 0% to 100%.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MARgin 11` sets the DDR write search margin value to 11%.

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:MARgin?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:MARgin 5`, indicating that the DDR write search margin value is set to 5%.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MAXCAS

This command sets or queries the DDR write chip select maximum Column Access Strobe (CAS) value.

Conditions

Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MAXCAS <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MAXCAS?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments `SEARCH<x>` specifies the search number.

`NR3` sets the DDR write chip select CAS maximum value.

Examples `EARCH:SEARCH1:TRIGger:A:DDRWRITE:MAXCAS 3.8E-3` sets the DDR write chip select CAS maximum value to 3.8 milliseconds for search 1.

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MAXCAS?` might return `SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MAXCAS 3.8E-3`, indicating that the DDR write chip select CAS maximum value is set to 3.8 milliseconds for search 2.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MINCas

This command sets or queries the DDR write chip select minimum Column Access Strobe (CAS) value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Chip Select".

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MINCas <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MINCas?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments SEARCH<x> specifies the search number.

NR3 sets the DDR write chip select CAS minimum value.

Examples EARCH:SEARCH1:TRIGger:A:DDRWRITE:MINCas 3.8E-3 sets the DDR write chip select CAS minimum value to 3.8 milliseconds for search 1.

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MINCas? might return SEARCH:SEARCH2:TRIGger:A:DDRWRITE:MINCas 3.8E-3, indicating that the DDR write chip select CAS minimum value is set to 3.8 milliseconds for search 2.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth

This command sets or queries the DDR write postamble length when the search type is DDR WRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth
{500E-3|1.5|1}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth?
```

Arguments 500E-1.5 sets the postamble length to 1.5 tCK (number of clock cycles).

500E-1 sets the postamble length to 1 tCK (number of clock cycles).

500E-3 sets the postamble length to 0.5 tCK (number of clock cycles).

Examples SEARCH:SEARCH6:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth 500E-3 sets the DDR write postamble length for Search 6 to 0.5 tCK.

SEARCH:SEARCH2:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth? might return SEARCH:SEARCH2:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth 500E-1, indicating that the write postamble length for Search 2 is set to 1 tCK.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE

This command sets or queries the DDR write preamble type when the search type is DDR WRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE
{STATIC|DYNAMIC}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE?
```

Arguments `STATIC` sets the DDR Write preamble type to Static.

`DYNAMIC` sets the DDR Write preamble type to Dynamic.

Examples `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:PREAMBLE:TYPE STATIC` sets the DDR write preamble type to Static for Search 1.

`SEARCH:SEARCH4:TRIGger:A:DDRWRITE:PREAMBLE:TYPE?` might return `SEARCH:SEARCH4:TRIGger:A:DDRWRITE:PREAMBLE:TYPE DYNAMIC`, indicating that the DDR write preamble type for Search 4 is set to dynamic mode.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH

This command sets or queries the DDR write data high reference level value, when the search type is DDR WRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

Arguments NR3 sets the DDR read search data high reference value.

Examples SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH 3.8E-3 sets the DDR write search data high reference value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH 50E-3, indicating that the DDR write search data high reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW

This command sets or queries the DDR write data low reference level value, when the search type is DDR WRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

Arguments NR3 sets the DDR read search data low reference value.

Examples SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW 3.8E-3 sets the DDR write search data low reference value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW 50E-3, indicating that the DDR write search data low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:MID

This command sets or queries the DDR write data mid reference level value, when the search type is DDR WRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:MID <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:MID?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

Arguments NR3 sets the DDR read search data mid reference value.

Examples `SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:DATA:MID 3.8E-3` sets the DDR write search data mid reference value.

`SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:MID?` might return `SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:DATA:MID 50E-3`, indicating that the DDR write search data mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH

This command sets or queries the DDR write reference level strobe high value, when the search type is DDR WRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH
<NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

Arguments NR3 sets the DDR write search strobe high value.

Examples SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH 3.8E-3 sets the DDR write search strobe high value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH 50E-3, indicating that the DDR write search strobe high value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW

This command sets or queries the DDR write reference level strobe low value, when the search type is DDR WRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW
<NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

Arguments NR3 sets the DDR read search strobe low reference value.

Examples SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW 3.8E-3 sets the DDR write search strobe low reference value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW

50E-3, indicating that the DDR write search strobe low reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID

This command sets or queries the DDR write reference level strobe mid value, when the search type is DDR WRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR write reference level mode to Manual before using this command. See Related Commands.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID
<NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

Arguments NR3 sets the DDR read search strobe mid reference value.

Examples SEARCH:SEARCH2:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID 3.8E-3 sets the DDR write search strobe mid reference value.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID 50E-3, indicating that the DDR write search strobe mid reference value is set to 50E-3.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode

This command sets or queries the DDR write reference level mode to auto or manual, when the search type is DDR WRITE.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode
{AUTO|MANUAL}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID

Arguments AUTO sets the DDR write reference level mode to auto.

MANUAL sets the DDR write reference level mode to manual. Use the Related Commands to set the Reference levels when in Manual mode.

Examples SEARCH:SEARCH3:TRIGger:A:DDRWRITE:REFLEVELMode MANUAL sets the DDR write search reference level mode to Manual for Search 3.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLEVELMode? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:REFLEVELMode MANUAL, indicating that the DDR write search reference level mode set to automatic for Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard

This command sets or queries the DDR write search standard as DDR3 or LPDDR3.

Conditions Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard {DDR3|LPDDR3}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard?
```

Arguments DDR3 sets the DDR write search standard as DDR3.

LPDDR3 sets the DDR write search standard as LPDDR3.

Examples

`SEARCH:SEARCH2:TRIGger:A:DDRWRITE:STANdard DDR3` sets the DDR standard to DDR3 for DDR write Search 2.

`SEARCH:SEARCH5:TRIGger:A:DDRWRITE:STANdard?` might return `SEARCH:SEARCH5:TRIGger:A:DDRWRITE:STANdard LPDDR3` indicating that the DDR standard is set to LPDDR3 for DDR write Search 5.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STROBESource

This command sets or queries the DDR write strobe source when the search type is DDR Write.

Conditions

Requires option 6-DBDDR3

Requires 6 Series MSO instrument

Load a reference waveform on the instrument before using this command to set the search data source to a reference waveform.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STROBESource
{CH<x>|CH<x>_D<x<|Math<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STROBESource?
```

Arguments

`CH<x>` specifies channel <x> as the DDR write strobe source for the specified search <x>.

`CH<x>_D<x>` specifies digital waveform <x> of channel <x> as the DDR write strobe source for the specified search <x>.

`Math<x>` specifies math waveform <x> as the DDR write strobe source for the specified search <x>.

`REF<x>` specifies reference waveform <x> as the DDR write strobe source for the specified search <x>.

`REF<x>_D<x>` specifies digital waveform <x> of reference waveform <x> as the DDR write strobe source for the specified search <x>.

Examples

`SEARCH:SEARCH2:TRIGGER:A:DDRWRITE:STROBESource MATH4` sets the Search 2 source waveform to Math 4 for the DDR write strobe search.

`SEARCH:SEARCH1:TRIGGER:A:DDRWRITE:STROBESource?` might return `SEARCH:SEARCH1:TRIGGER:A:DDRWRITE:STROBESource CH3`, indicating that the Channel 3 waveform is the source for the DDR write strobe of Search 1.

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:TOLERance

This command sets or queries the DDR write logic state tolerance value.

Conditions Requires option 6-DBDDR3.

Requires a 6 Series MSO instrument.

Set the DDR Read standard (DDR3 or LPDDR3) before using this command. See Related Commands.

Set burst detection method as "Logic State".

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:TOLERance <NR3>
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:TOLERance?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod

Arguments SEARCH<x> specifies the search number.

NR3 sets the logic state tolerance value.

Examples SEARCH:SEARCH3:TRIGger:A:DDRWRITE:TOLERance 400E-3 sets the DDRWRITE logic state tolerance value of search 3 to .004.

SEARCH:SEARCH1:TRIGger:A:DDRWRITE:TOLERance? might return SEARCH:SEARCH1:TRIGger:A:DDRWRITE:TOLERance 500E-3, indicating that the DDRWRITE logic state tolerance value is set to 500E-3 for search 1.

SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe

This command sets or queries the slope for an edge trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe {RISe|FALL|EITher}
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe?
```

Arguments

RISe specifies a rising edge.

FALL specifies a falling edge.

EITher specifies either rising or falling edge.

Examples

SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE RISE sets the slope for search 1 to rise.

SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE? might return SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE RISE, indicating that the slope for the trigger for search 1 is rise.

SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce

This command sets or queries the source waveform for an edge trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce?
```

Arguments

CH<x> specifies one input channel as the edge source, where the channel number is specified by x.

CH<x>_D<x> specifies a digital reference waveform as the source waveform for the specified search.

MATH<x> specifies the math waveform as the search source, where the math number is specified by x.

REF<x> specifies the reference waveform as the search source, where the reference number is specified by x.

REF<x>_D<x> specifies a digital reference waveform as the source waveform for the specified search.

Examples

SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE CH2 sets the source waveform for the A trigger to Channel 2.

SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE? might return SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE CH2, indicating that the Channel 2 edge trigger is the source for search 1.

SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold

This command sets or queries the source threshold level for an edge trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold?
```

Arguments <NR3> is the source threshold level for an edge trigger search.

Examples SEARCH:SEARCH1:TRIGger:A:EDGE:THReshold 50.0e-3 sets the threshold to 50 mV.

SEARCH:SEARCH1:TRIGger:A:EDGE:THReshold? might return SEARCH:SEARCH1:TRIGGER:A:EDGE:THRESHOLD 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCk:THReshold

This command sets or queries the logic clock threshold for a logic trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCk:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCk:THReshold?
```

Arguments <NR3> is the logic clock threshold.

Examples SEARCH:SEARCH1:TRIGger:A:LOGIc:CLOCk:THReshold 50.0e-3 sets the threshold to 50 mV.

SEARCH:SEARCH1:TRIGger:A:LOGIc:CLOCk:THReshold? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:CLOCK:THRESHOLD 0.0E+0 indicating the threshold is set to 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:DELTatime

This command specifies the Logic search delta time value. The time value is used as part of the Logic search condition to determine if the duration of a logic pattern meets the specified time constraints. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:DELTatime <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:DELTatime?
```

Arguments <NR3> is delta time value.

Examples SEARCH:SEARCH1:TRIGger:A:LOGIc:DELTatime 1.5e-9 sets the delta time to 1.5 ns.

SEARCH:SEARCH1:TRIGger:A:LOGIc:DELTatime? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:DELTATIME 1.0000E-9 indicating the delta time is 1.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCtion

This command sets or queries the logic operator for a pattern or state trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCtion {AND|NANd|NOR|OR}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCtion?
```

Arguments AND places a mark if all conditions are true.

NANd places a mark if any of the conditions are false.

NOR places a mark if all conditions are false.

OR places a mark if any of the conditions are true.

Examples SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION AND sets the trigger a logic function for search 1 to AND.

SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION NOR, indicating that the logic function for search 1 is set to NOR.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce

This command specifies or queries the channel to use as the clock source for logic trigger. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce
{CH<x>|Ch<x>_D<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce?
```

Arguments Arguments are the possible input channels.

Examples SEARCH:SEARCH1:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce CH2 sets the clock source to channel 2.

SEARCH:SEARCH1:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE UNDEFINED indicating the clock source is not defined.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:CH<x>

This command sets or queries the voltage level to use for logic trigger search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:CH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:CH<x>?
```

Arguments <NR3> is the voltage level to use for logic trigger search.

Examples SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel:CH2 50.0e-3 sets the level to 50 mV.

SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel:CH2? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:CH2 0.0E+0 indicating the level is set to 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:MATH<x>

This command sets the voltage level to use for logic trigger search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:MATH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:MATH<x>?
```

Arguments <NR3> is the voltage level to use for logic trigger search.

Examples SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel:MATH1 50.0e-3 sets the level to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel:MATH1? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:MATH1 0.0E+0 indicating the level is set to 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:REF<x>

This command sets the voltage level to use for logic trigger search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:REF<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:REF<x>?
```

Arguments <NR3> is the voltage level to use for logic trigger search.

Examples SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel:REF1 50.0e-3 sets the level to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:LOGIc:LEVel:REF1? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LEVEL:REF1 0.0E+0 indicating the level is set to 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x> {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x>?
```

Arguments

H specifies triggering when the pattern is high.

L specifies triggering when the pattern is low.

X specifies triggering when the pattern is high or low.

Examples

SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:CH2 H sets the channel 2 pattern to a high.

SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:CH2? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPATTERN:CH2 X indicating channel 2 is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x>_D<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x>_D<x>
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x>_D<x>?
```

Arguments H specifies triggering when the pattern is high.

L specifies triggering when the pattern is low.

X specifies triggering when the pattern is high or low.

Examples SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:CH1_D1 H sets the channel patter to a high.

SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:CH1_D1? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPATTERN:CH1_D1 X indicating the channel patter is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:MATH<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:MATH<x>
{H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:MATH<x>?
```

Arguments H specifies triggering when the pattern is high.

L specifies triggering when the pattern is low.

X specifies triggering when the pattern is high or low.

Examples SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:MATH1 H sets the pattern to a high.

SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:MATH1? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPATTERN:MATH1 X indicating the pattern is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x> {H|L|X}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x>?
```

Arguments `H` specifies triggering when the pattern is high.

`L` specifies triggering when the pattern is low.

`X` specifies triggering when the pattern is high or low.

Examples `SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:REF1 H` sets the pattern to a high.

`SEARCH:SEARCH1:TRIGger:A:LOGIc:LOGICPattern:REF1?` might return `SEARCH:SEARCH1:TRIGGER:A:LOGIC:LOGICPATTERN:REF1 X` indicating the pattern is a don't care.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:POLarity

This command sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic search type. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:POLarity
{POSitive|NEGative|EITher}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:POLarity?
```

Arguments `POSitive` specifies using the positive clock edge.

`NEGative` specifies using negative clock edge.

`EITher` specifies using either the positive or negative clock edge.

Examples `SEARCH:SEARCH1:TRIGger:A:LOGIc:POLarity NEGATIVE` sets the polarity to negative.

`SEARCH:SEARCH1:TRIGger:A:LOGIc:POLarity?` might return `SEARCH:SEARCH1:TRIGGER:A:LOGIC:POLARITY POSITIVE` indicating the polarity is set to positive.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:USEClockedge

This command specifies whether or not Logic search uses a clock source. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:USEClockedge {OFF|ON|0|1}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:USEClockedge?
```

Arguments

`OFF` specifies not to use the clock source.

`ON` specifies to use the clock source.

`0` specifies not to use the clock source.

`1` specifies to use the clock source.

Examples

`SEARCH:SEARCH1:TRIGger:A:LOGIc:USEClockedge ON` specifies to use the clock source.

`SEARCH:SEARCH1:TRIGger:A:LOGIc:USEClockedge?` might return `SEARCH:SEARCH1:TRIGGER:A:LOGIC:USECLOCKEDGE 0` indicating not to use the clock source.

SEARCH:SEARCH<x>:TRIGger:A:LOGIc:WHEn

This command sets or queries the condition for generating an A or B logic search with respect to the defined input pattern.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:WHEn
{TRUe|FALSe|MOREThan|LESSThan|EQual|UNEQual}
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:WHEn?
```

Arguments

`TRUe` searches on an input value that is true.

`FALSe` searches on an input value that is false.

`MOREthan` searches on an input value that is greater than a set value.

`LESSthan` searches on an input value that is less than a set value.

`EQual` searches on an input value that is equal to a set value.

UNEQual searches on an input value that is not equal to a set value.

Examples SEARCH:SEARCH1:TRIGger:A:LOGIc:WHEn FALSE specifies a search on an input value that is false.

SEARCH:SEARCH1:TRIGger:A:LOGIc:WHEn? might return SEARCH:SEARCH1:TRIGGER:A:LOGIC:WHEN TRUEindicating a search on an input value that is true.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit

This command specifies the upper limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit <NR3>
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit?
```

Arguments <NR3> is the upper limit to use, in seconds, when searching for a pulse.

Examples SEARCH:SEARCH1:TRIGger:A:PULSEWidth:HIGHLimit 2.5e-9 sets the high limit to 2.5 ns.

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:HIGHLimit? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:HIGHLIMIT 2.0000E-9 indicating the pulse width high limit is 2.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOGICQUALification

This command specifies whether or not to use logic qualification for a pulse width search. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOGICQUALification
{ON|OFF}
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOGICQUALification?
```

Arguments ON specifies to use logic qualification.

OFF specifies not to use logic qualification.

Examples SEARCH:SEARCH1:TRIGger:A:PULSEWidth:LOGICQUALification ON turns on logic qualification.

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:LOGICQUALification? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:LOGICQUALIFICATION OFF indicating logic qualification is off.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit

This command specifies the lower limit to use, in seconds, when searching for a pulse whose duration is inside or outside a range of two values. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit <NR3>
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit?
```

Arguments <NR3> is the lower limit to use, in seconds, when searching for a pulse.

Examples SEARCH:SEARCH1:TRIGger:A:PULSEWidth:LOWLimit 0.5e-9 sets the low limit to 0.5 ns.

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:LOWLimit? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:LOWLIMIT 1.0000E-9 indicating the low limit is 2.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity

This command specifies the polarity for a pulse width search. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity
{POSitive|NEGative}
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity?
```

Arguments `POSitive` specifies positive polarity for a pulse width search.

`NEGative` specifies negative polarity for a pulse width search.

Examples `SEARCH:SEARCH1:TRIGger:A:PULSEWidth:POLarity NEGATIVE` sets the polarity to negative.

`SEARCH:SEARCH1:TRIGger:A:PULSEWidth:POLarity?` might return `SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:POLARITY POSITIVE` indicating the polarity is positive.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce

This command sets and queries the source for the pulse width search input. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce
{CH<x>|CH<x>_D<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce?
```

Arguments Arguments are possible sources.

Examples `SEARCH:SEARCH1:TRIGger:A:PULSEWidth:SOUrce CH1` sets the source to channel 1.

`SEARCH:SEARCH1:TRIGger:A:PULSEWidth:SOUrce?` might return `SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:SOURCE CH2` indicating channel 2 is the source.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold

Sets or queries the source threshold level for a pulse width trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold?
```

Arguments <NR3> is the source threshold level for a pulse width trigger search.

Examples SEARCH:SEARCH1:TRIGger:A:PULSEWidth:THReshold 1.0e-9 sets to 1.0 V.

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:THReshold? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:THRESHOLD 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn

This command specifies to search for a pulse with a width (duration) that is less than, greater than, equal to, or unequal to a specified value (set using SEARCh:A:PULSEWidth:WIDth), OR whose SEARCh:A:PULSEWidth:LOWLimit and SEARCh:A:PULSEWidth:HIGHLimit). The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn
{LESSthan|MOREthan| EQual|UNEQual|WIThin|OUTside}
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn?
```

Arguments LESSthan causes a search when a pulse is detected with a width less than the time set by the SEARch:A:PULSEWidth:WIDth command.

MOREthan causes a search when a pulse is detected with a width greater than the time set by the SEARch:A:PULSEWidth:WIDth command.

EQual causes a search when a pulse is detected with a width equal to the time period specified in SEARch:A:PULSEWidth:WIDth within a ±5% tolerance.

UNEQual causes a search when a pulse is detected with a width greater than or less than (but not equal) the time period specified in SEARch:A:PULSEWidth:WIDth within a ±5% tolerance.

WIThin causes a search when a pulse is detected that is within a range set by two values.

OUTside causes a search when a pulse is detected that is outside of a range set by two values.

Examples SEARCH:SEARCH1:TRIGger:A:PULSEWidth:WHEn Outside causes a search when a pulse is detected that is outside the set range.

SEARCH:SEARCH1:TRIGger:A:PULSEWidth:WHEn? might return SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WHEN WITHIN indicating that a search will occur when a pulse is detected that is within a set range.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification

This command specifies whether or not to use logic qualification for a runt search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification {ON|OFF}
SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification?
```

Arguments

ON specifies to use logic qualification for a runt search.

OFF specifies not to use logic qualification for a runt search.

Examples

SEARCH:SEARCH1:TRIGger:A:RUNT:LOGICQUALification ON turns on logic qualification.

SEARCH:SEARCH1:TRIGger:A:RUNT:LOGICQUALification? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:LOGICQUALIFICATION OFF indicating logic qualification is off.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity

This command specifies the polarity for the runt search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity
{POSitive|NEGative|EITher}
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity?
```

Arguments

POSitive specifies using positive polarity for the runt search.

NEGative specifies using negative polarity for the runt search.

EITher specifies using either positive or negative polarity for the runt search.

Examples `SEARCH:SEARCH1:TRIGger:A:RUNT:POLarity NEGATIVE` specifies using negative polarity for the runt search.

`SEARCH:SEARCH1:TRIGger:A:RUNT:POLarity?` might return `SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY POSITIVE` indicating the polarity is set to positive.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce

This command sets and queries the source for the Runt search input. The search number is specified by x.

Group Search and Mark

Syntax `SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce {CH<x>|REF<x>}`

Arguments Arguments are the available sources.

Examples `SEARCH:SEARCH1:TRIGger:A:RUNT:SOUrce CH1` sets the source to channel 1.

`SEARCH:SEARCH1:TRIGger:A:RUNT:SOUrce?` might return `SEARCH:SEARCH1:TRIGGER:A:RUNT:SOURCE CH2` indicating the source is set to channel 2.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH

This command sets or queries the source threshold HIGH level for a runt trigger search to determine where to place a mark.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH?
```

Arguments `<NR3>` is the source threshold HIGH level for a runt trigger search.

Examples `SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:HIGH 50.0E-3` sets the high threshold to 50 mV.

SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:HIGH? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:HIGH 0.0E+0 indicating the high threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW

Sets or queries the source threshold LOW level for a runt trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW?
```

Arguments <NR3> is the source threshold LOW level for a runt trigger search.

Examples SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:LOW 50.0e-3 sets the thresholdo to 50 mV.

SEARCH:SEARCH1:TRIGger:A:RUNT:THReshold:LOW? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:THRESHOLD:LOW 0.0E+0 indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn

This command sets or queries the condition setting for a runt trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn
{OCCURS|LESSthan|MOREthan|EQual}NOTEQual}
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn?
```

Arguments LESSthan argument sets the instrument to search if the a runt pulse is detected with width less than the time set by the SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth command.

MOREthan argument sets the instrument to search if the a runt pulse is detected with width more than the time set by the SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth command.

EQual argument sets the instrument to search when the pattern is true for a time period equal to the time period specified in SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth within a ±5% tolerance.

NOTEQual argument sets the instrument to search when the pattern is true for atime period greater than or less than (but not equal) the time period specified in SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth within a ±5% tolerance.

OCCURS argument specifies a search event if a runt of any detectable width occurs.

Examples

SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN MORETHAN sets the instrument to trigger when a runt pulse is detected with width wider than the time set by the SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth command.

SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN OCCURS, indicating that a mark is placed if a runt trigger event occurs.

SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth

This command sets or queries the width setting for a runt trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth <NR3>
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth?
```

Arguments

<NR3> specifies the minimum width in seconds.

Examples

SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH 400E-12 sets the runt trigger minimum width for search 1 to 0.4 nanoseconds.

SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH? might return SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH 500.0000E-12, indicating that the runt trigger minimum width for search 1 is set to 0.5 nanoseconds.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:EDGE

This command sets or queries the clock slope setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:EDGE {FALL|RISe}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:EDGE?
```

Arguments

FALL specifies the polarity as the clock falling edge.

RISe specifies the polarity as the clock rising edge.

Examples

SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:EDGE FALL sets the setup/hold trigger clock slope polarity for search 1 to falling edge.

SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:EDGE? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:EDGE RISE, indicating that the setup/hold trigger clock slope polarity for search 1 is set to rising edge.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce

This command sets or queries the clock source setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce?
```

Arguments

CH<x> specifies an input channel as the edge source, where <x> = 1, 2, 3, 4, 5, 6, 7, or 8, depending on the number of channels in your instrument.

CH<x>_D<x> specifies a digital waveform as the setup and hold clock source waveform for the specified search.

MATH<x> specifies the math waveform as the search source, where <x> = ≥1.

REF<x> specifies the reference waveform as the search source, where <x> = ≥1.

REF<x>_D<x> specifies a digital reference waveform as the setup and hold clock source waveform for the specified search.

Examples

SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE MATH1 sets the setup/hold trigger clock source setting for search 1 to MATH1.

SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE CH1, indicating that the setup/hold trigger clock source setting for search 1 is set to CH1.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:THReshold

This command sets or queries the clock threshold setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:THReshold?
```

Arguments <NR3> the clock threshold setting for a setup/hold trigger search.

Examples `SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD -1.3` sets the setup/hold trigger clock threshold setting for search 1 to -1.3 volts.
`SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD?` might return `SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD -1.3000`, indicating that the setup/hold trigger clock threshold setting for search 1 is set to -1.3 volts.

```
** CANNOT USE LocalCmd HERE BECAUSE OF VARIABLES **
```

SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime

This command sets or queries the hold time setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime?
```

Arguments <NR3> specifies the hold time setting in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

Examples `SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME 400E-12` sets the setup/hold trigger hold time setting for search 1 to 400 ps.

`SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME?` might return `SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME 500.0000E-12`, indicating that the setup/hold trigger hold time setting for search 1 is set to 0.5 ns.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x>

This command sets or queries the voltage level to use for setup & hold trigger search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x>?
```

Arguments <NR3> the voltage level to use for setup & hold trigger search.

Examples `SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:CH1 50.0e-3` sets the level to 50.0 mV.

`SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:CH1?` might return `SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LEVEL:CH1 0.0E+0` indicating the level is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x>

This command sets or queries the voltage level to use for setup & hold trigger search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x>?
```

Arguments <NR3> isi the voltage level to use for setup & hold trigger search.

Examples `SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:MATH1 50.0e-3` sets the level to 50.0 mV.

`SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:MATH1?` might return `SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LEVEL:MATH1 0.0E+0` indicating the level is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x>

This command sets or queries the voltage level to use for setup & hold trigger search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x> <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x>?
```

Arguments <NR3> is the voltage level to use for setup & hold trigger search.

Examples SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:REF1 50.0e-3 sets the level to 50.0 mV.

SEARCH:SEARCH1:TRIGger:A:SETHold:LEVel:REF1? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LEVEL:REF1 0.0E+0 indicating the level is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>
{INCLude|DONTInclude}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>?
```

Arguments INCLude specifies including the specified channel SETHOLD inputs in the specified search.

DONTInclude specifies not including the specified channel SETHOLD inputs in the specified search.

Examples SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:CH1 INCLUDE specifies including the specified channel SETHOLD inputs in the specified search.

SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:CH1? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPATTERN:CH1 DONTINCLUDE indicating the specified channel SETHOLD inputs will not be included in the specified search.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>_D<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>_D<x>
{INCLude|DONTInclude}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>_D<x>?
```

Arguments

INCLude specifies including the specified digital channel SETHOLD inputs in the specified search.

DONTInclude specifies not including the specified digital channel SETHOLD inputs in the specified search.

Examples

SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:CH1_D0 INCLUDE specifies including the specified digital channel SETHOLD inputs in the specified search.

SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:CH1_D0? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPATTERN:CH1_D0 DONTINCLUDE indicating the specified digital channel SETHOLD inputs will not be included in the specified search.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:MATH<x>

This command sets or queries the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:MATH<x>
{INCLude|DONTInclude}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:MATH<x>?
```

Arguments

`INCLude` specifies including the specified math SETHOLD inputs in the specified search.

`DONTInclude` specifies not including the specified math SETHOLD inputs in the specified search.

Examples

`SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:MATH1 INCLUDE` specifies including the specified math SETHOLD inputs in the specified search.

`SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:MATH1?` might return `SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPATTERN:MATH1 DONTINCLUDE` indicating the specified math SETHOLD inputs will not be included in the specified search.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x>

This command sets and returns the conditions used for generating an A logic pattern, with respect to the defined input pattern, and identifies the time that the selected pattern may be true and still generate the trigger. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x>
{INCLude|DONTInclude}
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x>?
```

Arguments

`INCLude` specifies including the specified reference SETHOLD inputs in the specified search.

`DONTInclude` specifies not including the specified reference SETHOLD inputs in the specified search.

Examples

`SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:REF1 INCLUDE` specifies including the specified reference SETHOLD inputs in the specified search.

`SEARCH:SEARCH1:TRIGger:A:SETHold:LOGICPattern:REF1?` might return `SEARCH:SEARCH1:TRIGGER:A:SETHOLD:LOGICPATTERN:REF1`

DONTINCLUDE indicating the specified reference SETHOLD inputs will not be included in the specified search.

SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime

This command sets or queries the setup time setting for a setup/hold trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime <NR3>
SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime?
```

Arguments <NR3> specifies the setup time for setup and hold violation triggering.

Examples SEARCH:SEARCH1:TRIGGER:A:SETHOLD:SETTIME 2E-9 sets the setup/hold trigger setup time setting for search 1 to 2 ns.

SEARCH:SEARCH1:TRIGGER:A:SETHOLD:SETTIME? might return SEARCH:SEARCH1:TRIGGER:A:SETHOLD:SETTIME 1.0000E-9, indicating that the setup/hold trigger setup time setting for search 1 is set to 1.0 ns.

SEARCH:SEARCH<x>:TRIGger:A:STATE

This command sets or queries the enabled state of the search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:STATE {<NR1>|OFF|ON}
```

Arguments <NR1> = 1 enables the search. Any other character disables the search.

ON enables the search.

OFF disables the search.

Examples SEARCH:SEARCH1:TRIGger:A:STATE OFF disables the search.

SEARCH:SEARCH1:TRIGger:A:STATE? might return SEARCH:SEARCH1:TRIGGER:A:STATE 1 indicating search 1 is enabled.

SEARCH:SEARCH<x>:TRIGger:A:STOPAcq

This command sets or queries whether acquisitions are stopped when a search hit is found. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:STOPAcq {<NR1>|OFF|ON}
SEARCH:SEARCH<x>:TRIGger:A:STOPAcq?
```

Arguments <x> is the number of the search on which to enable or disable the stop acquisition function.

<NR1> = 1 enables stopping when a search hit is found. Any other character disables the feature.

ON enables stopping when a search hit is found.

OFF disables stopping on a search hit.

Examples SEARCH:SEARCH1:TRIGger:A:STOPAcq ON enables stopping when a search hit is found.

SEARCH:SEARCH3:TRIGger:A:STOPAcq? might return :SEARCH:SEARCH3:TRIGGER:A:STOPACQ 1 indicating acquisitions are stopped when a search hit is found on search number 3.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification

This command specifies whether or not to use logic qualification for a timeout search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification
{ON|OFF}
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification?
```

Arguments ON specifies to use logic qualification.

OFF specifies not to use logic qualification.

Examples `SEARCH:SEARCH1:TRIGger:A:TIMEOut:LOGICQUALification ON` specifies to use logic qualification.

`SEARCH:SEARCH1:TRIGger:A:TIMEOut:LOGICQUALification?` might return `SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:LOGICQUALIFICATION OFF` indicating logic qualification is off.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity

The polarity to be used for a Timeout search. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity
{STAYSHigh|STAYSLow|EITher}
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity?
```

Arguments `STAYSHigh` specifies the polarity stays HIGH.

`STAYSLow` specifies the polarity stays LOW.

`EITher` specifies the polarity stays HIGH or stays LOW.

Examples `SEARCH:SEARCH1:TRIGger:A:TIMEOut:POLarity STAYSLOW` specifies the polarity stays LOW.

`SEARCH:SEARCH1:TRIGger:A:TIMEOut:POLarity?` might return `SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY STAYSHIGH` indicating the polarity stays HIGH.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce

This command sets and queries the source for timeout search input. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce
{CH<x>|CH<x>_D<x>|MATH<x>|REF<x>|REF<x>_D<x>}
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce?
```

Arguments Arguments are the available sources.

Examples `SEARCH:SEARCH1:TRIGger:A:TIMEOut:SOUrce`

`SEARCH:SEARCH1:TRIGger:A:TIMEOut:SOUrce?` might return `SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:SOURCE CH1_D0` indicating CH1_D0 is the source.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold

Sets or queries the source threshold level for a timeout trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold?
```

Arguments `<NR3>` is the source threshold level for a timeout trigger search.

Examples `SEARCH:SEARCH1:TRIGger:A:TIMEOut:THReshold 50.0e-3` sets the threshold to 50.0 mV.

`SEARCH:SEARCH1:TRIGger:A:TIMEOut:THReshold?` might return `SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:THRESHOLD 0.0E+0` indicating the threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe

This command sets or queries the time setting for a timeout trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe?
```

Arguments `<NR3>` is the time in seconds.

Examples `SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 400E-9` sets the timeout trigger time setting for search 1 to 400 ns.

SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME? might return SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 500.0000E-12, indicating that the timeout trigger time setting for search 1 is set to 500 ns.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:DELTATime

This command sets or queries the transition time setting for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:DELTATime <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:DELTATime?
```

Arguments <NR3> specifies the transition time in seconds.

Examples SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME 400E-9 sets the transition trigger time setting for search 1 to 400 ns.

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:DELTATIME? might return SEARCH:SEARCH1:TRIGGER:A:TRANSITION:DELTATIME 500.0000E-12, indicating that the transition trigger time setting for search 1 is set to 500 ps.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:LOGICQUALification

This command specifies whether or not to use logic qualification for a transition search. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:LOGICQUALification
{ON|OFF}
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:LOGICQUALification?
```

Arguments ON specifies to use logic qualification for a transition search.

OFF specifies not to use logic qualification for a transition search.

Examples SEARCH:SEARCH1:TRIGger:A:TRANsition:LOGICQUALification ON specifies to use logic qualification for a transition search.

SEARCH:SEARCH1:TRIGger:A:TRANsition:LOGICQUALification? might return SEARCH:SEARCH1:TRIGGER:A:TRANSITION:LOGICQUALIFICATION OFF indicating logic qualification is off.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:POLarity

This command specifies the polarity for the transition search. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:POLarity
{POSitive|NEGative|EITher}
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:POLarity?
```

Arguments

EITher places a mark on a transition of either polarity.

NEGative places a mark on a transition of negative polarity.

POSitive places a mark on a transition of positive polarity.

Examples

SEARCH:SEARCH1:TRIGger:A:TRANsition:POLarity NEGATIVE places a mark on a transition of negative polarity.

SEARCH:SEARCH1:TRIGger:A:TRANsition:POLarity? might return SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY POSITIVE indicating the polaity is positive.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:SOUrce

This command sets and queries the source for the transition search input. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:SOUrce
{CH<x>|MATH<x>|REF<x>}
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:SOUrce?
```

Arguments Arguments are the available sources.

Examples `SEARCH:SEARCH1:TRIGger:A:TRANsition:SOUrce CH1` sets the source to channel 1.

`SEARCH:SEARCH1:TRIGger:A:TRANsition:SOUrce?` might return `SEARCH:SEARCH1:TRIGGER:A:TRANSITION:SOURCE CH2` indicating channel 2 is the source.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:HIGH

Sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:HIGH?
```

Arguments `<NR3>` the source threshold HIGH level for a transition trigger search.

Examples `SEARCH:SEARCH1:TRIGger:A:TRANsition:THReshold:HIGH 50.0e-3` sets the high threshold to 50.0 mV.

`SEARCH:SEARCH1:TRIGger:A:TRANsition:THReshold:HIGH?` might return `SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:HIGH 0.0E+0` indicating the high threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:LOW

Sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:LOW?
```

Arguments `<NR3>` is the source threshold LOW level for a transition trigger searchk.

Examples `SEARCH:SEARCH1:TRIGger:A:TRANsition:THReshold:LOW -50.0e-3` sets the LOW threshold to -50.0 mV.

SEARCH:SEARCH1:TRIGger:A:TRANsition:THReshold:LOW? might return SEARCH:SEARCH1:TRIGGER:A:TRANSITION:THRESHOLD:LOW 0.0E+0 indicating the LOW threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:TRANsition:WHEn

This command sets or queries the condition setting for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:WHEn
{FASTERthan|SLOWERthan|EQual|UNEQual}
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:WHEn?
```

Arguments

FASTERthan sets the trigger to occur when the transitioning signal is faster than the set volts/second rate.

SLOWERthan sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.

EQual sets the trigger to occur when the transitioning signal is equal to the set volts/second rate.

UNEQual sets the trigger to occur when the transitioning signal is not equal to the set volts/second rate.

Examples

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN SLOWERTHAN sets the transition trigger condition setting for search 1 to SLOWERTHAN.

SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN? might return SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN FASTERTHAN, indicating that the transition trigger condition setting for search 1 is set to FASTERTHAN.

SEARCH:SEARCH<x>:TRIGger:A:TYPe

This command sets or queries the trigger type setting for a search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:TYPe
{EDGE|RUNT|TRANsition|PULSEWidth|TIMEOut|LOGIc|
SETHold|WINdow|Bus|DDRRead|DDRREADWrite|DDRWrite}
SEARCH:SEARCH<x>:TRIGger:A:TYPe?
```

Arguments

`EDGE` triggers when the source input signal amplitude crosses the specified level in the direction given by the slope.

NOTE. *Some trigger types are optional.*

`RUNT` triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before recrossing the first. The thresholds are set with the SEARCH:SEARCH<x>:TRIGger:A:RUNt:HIGH and SEARCH:SEARCH<x>:TRIGger:A:RUNt:LOW THRESHOLD commands.

`TRANsition` triggers when a pulse crosses both thresholds in the same direction as the specified polarity and the transition time between the two threshold crossings is greater or less than the specified time delta.

`PULSEWidth` triggers on input signal source pulses that are inside or outside of the given time range specified by SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit and SEARCH:SEARCH<x>:TRIGger:A:LOGIc:PATtern:WHEn:MORELimit. The polarity is selected using the SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity command.

`TIMEOut` triggers on an input signal source that stays above, stays below, or stays either above or beow the trigger level for a given time.

`LOGIc` specifies that a search occurs when specified conditions are met, and is controlled by the SEARCH:A:LOGIc commands.

`SETHold` triggers on a functional pattern combination of one to three data sources at the time of the clock transition.

`WINdow` triggers on an input signal source that enters or exits the horizontal band defined by the two trigger levels.

`Bus` specifies that a search occurs when a communications signal is found.

`DDRRead` triggers on a DDR Read event.

`DDRREADWrite` triggers on a DDR Read or Write event.

`DDRWrite` triggers on a DDR Write event.

Examples

`SEARCH:SEARCH1:TRIGGER:A:TYPE RUNT` sets the trigger type setting for search 1 to RUNT.

SEARCH:SEARCH1:TRIGGER:A:TYPE? might return SEARCH:SEARCH1:TRIGGER:A:TYPE EDGE, indicating that the trigger type setting for search 1 is set to EDGE.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:CROSSIng

This command sets or queries the window trigger threshold crossing of the selected trigger Source. The threshold crossing selection is only effective when :TRIGger:A:WINdow:WHEn is INSIDEGreater or OUTSIDEGreater. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:CROSSIng
{UPPer|LOWer|EITher|NONe}
```

Arguments

UPPer if :TRIGger:A:WINdow:WHEn is INSIDEGreater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGger:A:WINdow:WIDTH) and then exits through the upper threshold. If :TRIGger:A:WINdow:WHEn is OUTSIDEGreater, the instrument triggers when the signal remains above the upper threshold for longer than the time limit (:TRIGger:A:WINdow:WIDTH) and then crosses downward through the upper threshold.

LOWer if :TRIGger:A:WINdow:WHEn is INSIDEGreater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGger:A:WINdow:WIDTH) and then exits through the lower threshold. If :TRIGger:A:WINdow:WHEn is OUTSIDEGreater, the instrument triggers when the signal remains below the lower threshold for longer than the time limit (:TRIGger:A:WINdow:WIDTH) and then crosses upwards through the lower threshold.

EITher if :TRIGger:A:WINdow:WHEn is INSIDEGreater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGger:A:WINdow:WIDTH) and then exits through either the upper or lower threshold. If :TRIGger:A:WINdow:WHEn is OUTSIDEGreater, the instrument triggers when the signal remains either above the upper threshold or below the lower threshold for longer than the time limit (:TRIGger:A:WINdow:WIDTH) and then crosses a threshold.

NONe if :TRIGger:A:WINdow:WHEn is INSIDEGreater, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (:TRIGger:A:WINdow:WIDTH) without crossing through either the upper or lower threshold. If :TRIGger:A:WINdow:WHEn is OUTSIDEGreater, the instrument triggers when the signal remains outside the upper and lower

thresholds for longer than the time limit (:TRIGger:A:WINdow:WIDTH) without crossing through either the upper or lower threshold.

Examples SEARCH:SEARCH1:TRIGger:A:WINdow:CROSSIng LOWER sets the CROSSING to LOWER.

SEARCH:SEARCH1:TRIGger:A:WINdow:CROSSIng? might return SEARCH:SEARCH1:TRIGGER:A:WINDOW:CROSSING UPPER indicating that CROSSING is set to UPPER.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:LOGICQUALification

This command specifies or queries whether or not to use logic qualification for a window search. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:LOGICQUALification
{ON|OFF}
SEARCH:SEARCH<x>:TRIGger:A:WINdow:LOGICQUALification?
```

Arguments ON specifies to use logic qualification for a window search.

OFF specifies not to us logic qualification for a window search.

Examples SEARCH:SEARCH1:TRIGger:A:WINdow:LOGICQUALification ON turns ON logic qualification for a window search.

SEARCH:SEARCH1:TRIGger:A:WINdow:LOGICQUALification? might return SEARCH:SEARCH1:TRIGGER:A:WINDOW:LOGICQUALIFICATION OFF indicating logic qualification is off.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity

This command sets or queries the window trigger threshold crossing of the selected trigger Source. The search number is specified by x.

Group Search and Mark

Syntax
```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity
{UPPer|LOWer|EITher|NONe}
SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity?
```

Arguments

UPPer specifies that the instrument triggers when the signal remains above the upper threshold for longer than the time limit and then crosses downward through the upper threshold.

LOWer specifies that the instrument triggers when the signal remains below the lower threshold for longer than the time limit and then crosses upwards through the lower threshold.

EITher specifies that the instrument triggers when the signal remains either above the upper threshold or below the lower threshold for longer than the time limit and then crosses a threshold.

NONe specifies that the instrument triggers when the signal remains outside the upper and lower thresholds for longer than the time limit without crossing through either the upper or lower threshold.

Examples

SEARCH:SEARCH1:TRIGger:A:WINdow:POLarity LOWER sets the polarity to lower.

SEARCH:SEARCH1:TRIGger:A:WINdow:POLarity? might return SEARCH:SEARCH1:TRIGGER:A:WINDOW:POLARITY UPPER indicating the polarity is set to upper.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOUrce

This command sets and queries the source for the window search input. The search number is specified by x.

Group

Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOUrce
{CH<x>|MATH<x>|REF<x>}
SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOUrce?
```

Arguments

Arguments are the available sources.

Examples

SEARCH:SEARCH1:TRIGger:A:WINdow:SOUrce CH2 sets the source to channel 2.

SEARCH:SEARCH1:TRIGger:A:WINdow:SOUrce? might return SEARCH:SEARCH1:TRIGGER:A:WINDOW:SOURCE CH3 indicating the source is channel 3.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH

This command sets or queries the source threshold HIGH level for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH <NR3>
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH?
```

Arguments <NR3> is the source threshold HIGH level for a transition trigger search.

Examples `SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:HIGH 50.0e-3` sets the high threshold to 50.0 mV.

`SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:HIGH?` might return `SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:HIGH 0.0E+0` indicating the high threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW

This command sets or queries the source threshold LOW level for a transition trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW <NR3>
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW?
```

Arguments <NR3> the source threshold LOW level for a transition trigger search

Examples `SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:LOW -50.0e-3` sets the low threshold to -50.0 mV.

`SEARCH:SEARCH1:TRIGger:A:WINdow:THReshold:LOW?` might return `SEARCH:SEARCH1:TRIGGER:A:WINDOW:THRESHOLD:LOW 0.0E+0` indicating the low threshold is 0.0 V.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn

This command sets or queries the window search event. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn
{ENTERSWindow|EXITSWindow| INSIDEGreater|OUTSIDEGreater}
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn?
```

Arguments

`ENTERSWindow` specifies a window search when the signal enters the window.

`EXITSWindow` specifies a window search when the signal exits the window.

`OUTSIDEGreater` specifies a search when the signal leaves the window defined by the threshold levels for the time specified by Width.

`INSIDEGreater` specifies a search when the signal enters the window defined by the threshold levels for the time specified by Width.

Examples

`SEARCH:SEARCH1:TRIGger:A:WINdow:WHEn EXITSWINDOW` specifies a window search when the signal exits the window.

`SEARCH:SEARCH1:TRIGger:A:WINdow:WHEn?` might return `SEARCH:SEARCH1:TRIGGER:A:WINDOW:WHEN ENTERSWINDOW` indicating a window search when the signal enters the window.

SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth

This command sets or queries the width setting for a window trigger search to determine where to place a mark. The search number is specified by x.

Group Search and Mark

Syntax

```
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth <NR3>
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth?
```

Arguments

`<NR3>` specifies the minimum width in seconds.

ECL specifies a preset high level of -1.3 V and TTL specifies a preset high level of +1.4V.

Examples `SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH 400E-12` sets the window trigger width setting for search 1 to 0.4 ns.

`SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH?` might return `SEARCH:SEARCH1:TRIGGER:A:WINDOW:WIDTH 500.0000E-12`, indicating that the window trigger width setting for search 1 is set to 0.5 ns.

SEARCH:SELected

This command sets or queries the selected search, for example SEARCH1. The search number is specified by x.

Group Search and Mark

Syntax `SEARCH:SELected SEARCH1`

Arguments `SEARCH1` is the specified search.

Examples `SEARCH:SELected SEARCH1` specifies a search 1 search.

`SEARCH:SELected?` might return `:SEARCH:SELECTED SEARCH1` indicating search 1 is selected.

SEARCHTABle (No Query Form)

This command adds or deletes a new search event table in an Option 5-WIN (Microsoft Windows 10 OS) TekExpress compliance testing application.

Group Search and Mark

Syntax `SEARCHTABle {ADDNew|DELete} <qstring>`

Arguments `ADDNew` adds a new search events table in the display area.

`DELete` removes a displayed search events table from the display area.

`<qstring>` contains the name of the search table.

Examples `SEARCHTABle:ADDNew "SEARCHTABLE1"` adds the SEARCHTABLE1 in the display area.

SEARCHTABle:DELete "SEARCHTABLE1" removes the SEARCHTABLE1 from the display area.

SET? (Query Only)

This query-only command returns the commands that list the instrument settings, except for configuration information for the calibration values. You can use these commands to return the instrument to the state it was in when you made the SET? query. The SET? query always returns command headers, regardless of the setting of the HEADer command. This is because the returned commands are intended to be sent back to the instrument as a command string. The VERBose command can still be used to specify whether the returned headers should be abbreviated or full-length.

This command is identical to the *LRN? command.

Group Miscellaneous

Syntax SET?

Related Commands HEADer

*LRN?

VERBose

Examples SET? might return the following response: :ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET 1;:FASTACQ:STATE 0;:APPLICATION:GPKNOB1:ACTIVE 0;:APPLICATION:GPKNOB2:ACTIVE 0;:APPLICATION:WINDOW:HEIGHT 236;WIDTH 640;:APPLICATION:SCOPEAPP:STATE RUNNING;WINDOW FULLSCREEN;:APPLICATION:EXTAPP:STATE NOTRUNNING;:AUXOUT:SOURCE ATRIGGER;EDGE FALLING;:CMDBATCH 1;:HEADER 1;:LOCK NONE;:ROSC:SOURCE INTERNAL;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:CLOCK 1;COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT;REFCOLOR DEFAULT;:DISPLAY:FILTER SINX;FORMAT YT;GRATICULE FULL;INTENSITY:WAVEFORM 75.0000;AUTOBRIGHT 1;:DISPLAY:PERSISTENCE OFF;STYLE VECTORS;TRIGBAR SHORT;TRIGT 1;VARPERSIST 500.0000E-3;PORT FILE;:DIAG:LEVEL SUBSYS;:TRIGGER:A:MODE AUTO;TYPE EDGE;LEVEL 0.0000;HOLDOFF:BY DEFAULT;TIME 1.5000E-6;:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;.

SOCKETServer:ENAble

This command enables or disables the socket server which supports a telnet or other TCPIP socket connection to send commands and queries to the instrument. The default state is enabled.

Group Miscellaneous

Syntax

```
SOCKETServer:ENAble {0|1|OFF|ON}
SOCKETServer:ENAble?
```

Arguments

1 enables the socket server. If the state is 0 (disabled) and this command is sent to enable the socket server when the port is in use by another service, then the error event code 221 (Settings conflict) is posted to the event queue and the socket server remains disabled. In this case, select a different port number and attempt to enable the socket server again.

0 disables the socket server.

ON enables the socket server.

OFF disables the socket server.

Examples

SOCKETSERVER:ENABLE On enables the socket server.

SOCKETSERVER:ENABLE? might return :SOCKETSERVER:ENABLE 1 indicating the socket server is enabled.

SOCKETServer:PORT

This command sets the TCPIP port for the socket server connection.

Group Miscellaneous

Syntax

```
SOCKETServer:PORT <NR1>
SOCKETServer:PORT?
```

Arguments

<NR1> is the TCPIP port for the socket server connection.

If the socket server is enabled and the port specified is in use by another service, then the error event code 221 (Settings conflict) is posted to the event queue and the socket server remains in its current state (i.e. enabled/disabled and port address remain unchanged).

Similarly, if any sessions are active when the port is changed, this same error event may be posted to the event queue and the port will remain unchanged. In this case, exit all current sessions and send the :SOCKETServer:PORT command again.

Examples SOCKETSERVER:PORT 4000 sets the socket server port number to 4000.

SOCKETSERVER:PORT? might return :SOCKETSERVER:PORT 4000 indicating that the port number is 4000.

SOCKETServer:PROTOCol

This command sets or queries the protocol for the socket server.

Group Miscellaneous

Syntax
```
SOCKETServer:PROTOCol {TERMinal|NONe}
SOCKETServer:PROTOCol?
```

Arguments TERMinal specifies terminal protocol for the socket server. When set to TERMinal, a session startup message is sent to the socket and a command prompt is provided.

NONe disables the terminal features, allowing the server to be used for raw socket transactions, such as with a VISA socket server. The default setting is NONe.

Examples SOCKETSERVER:PROTOCOL NONE sets the protocol to none.

SOCKETSERVER:PROTOCOL? might return :SOCKETSERVER:PROTOCOL TERMINAL indicating the protocol is set to terminal.

*SRE

The *SRE (Service Request Enable) command sets and queries the bits in the Service Request Enable Register. For more information, refer to Registers.

Group Status and Error

Syntax
```
*SRE <NR1>
*SRE?
```

Related Commands

*CLS

DESE

*ESE

*ESR?

EVENT?

EVMsg?

FACtory

*STB?

Arguments

<NR1> is a value in the range from 0 through 255. The binary bits of the SRER are set according to this value. Using an out-of-range value causes an execution error. The power-on default for SRER is 0 if *PSC is 1. If *PSC is 0, the SRER maintains the previous power cycle value through the current power cycle.

Examples

*SRE 48 sets the bits in the SRER to binary 00110000.

*SRE? might return 32, showing that the bits in the SRER have the binary value of 00100000.

*STB? (Query Only)

The *STB? (Read Status Byte) query returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. For more information, refer to Registers.

Group

Status and Error

Syntax

*STB?

Related Commands

*CLS

DESE

*ESE

*ESR?

EVENT?

EVMsg?

FACtory

*SRE

Returns <NR1>

Examples *STB? might return 96, showing that the SBR contains the binary value 01100000.

SV:CH<x>:RF_AVErage:NUMAVg

This command sets or queries the number of acquisitions to be used when creating the Average trace for the specified spectrum trace channel in the Spectrum View. The Average spectrum trace shows the average of values from multiple acquisitions at each trace point.

Group Spectrum View

Syntax
```
SV:CH<x>:RF_AVErage:NUMAVg <NR1>
SV:CH<x>:RF_AVErage:NUMAVg?
```

Arguments CH<x> specifies the spectrum trace channel source.

<NR1> specifies the number of acquisitions to average. The range is 2 - 512, in exponential increments.

Examples SV:CH1:RF_AVErage:NUMAVg 100 sets the number of acquisitions to use for the Average spectrum trace of channel 1 to 128 (the closest power of 2 increment).

SV:CH2:RF_AVErage:NUMAVg? might return 16, indicating that the number of averages for the channel 2 Average spectrum trace is set to 16.

SV:CH<x>:RF_MAGnitude:FORMat

This command sets or queries the format of the Magnitude vs. Time trace for the specified channel.

Conditions Requires option SV-RFVT

Group Spectrum View

Syntax
```
SV:CH<x>:RF_MAGnitude:FORMat {AMPLINear|POWLINear|POWLOG}
SV:CH<x>:RF_MAGnitude:FORMat?
```

Related Commands DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:POSition

Arguments CH<x> is the channel number of the Magnitude vs. Time trace.

AMPLINear (Amplitude (linear)) sets the magnitude in Volts with square root conversion of linear power values.

POWLINear (Power (linear) sets the magnitude in Watts with direct use of linear power values.

POWLOG (Power (log) sets the magnitude in dB using log conversion of linear power values.

Examples SV:CH2:RF_MAGnitude:FORMat AMPLINear sets the Channel 2 Magnitude vs. Time trace to be shown with linear power values.

SV:CH1:RF_MAGnitude:FORMat? might return SV:CH1:RF_MAGnitude:FORMat POWLOG, indicating that the Channel 1 Magnitude vs. Time trace is displayed with log power values.

SV:CH<x>:RF_PHASe:REFerence:DEGrees

This command sets or queries the channel Phase Reference value in degrees.

Conditions Requires option SV-RFVT

Group Spectrum View

Syntax
```
SV:CH<x>:RF_PHASe:REFerence:DEGrees <NR3>
SV:CH<x>:RF_PHASe:REFerence:DEGrees?
```

Related Commands SV:CH<x>:RF_PHASe:REFerence:TIMe, SV:CH<x>:RF_PHASe:REFerence:POSition, SV:RF_PHASe:REFerence:MASTer

Arguments CH<x> is the channel number of the Phase vs. Time trace.

NR3 is the Phase Reference value, in degrees. This indicates a fixed phase value at the phase time set by SV:CH<x>:RF_PHASe:REFerence:TIMe. If CH<x> is the master phase reference, then the value is used to calculate the phase values of all other channels.

Examples SV:CH1:RF_PHASe:REFerence:DEGrees 25 sets the Phase Reference value to 25 degrees.

SV:CH2:RF_PHASe:REFerence:DEGrees? might return SV:CH2:RF_PHASe:REFerence:DEGrees 0, indicating that the Phase Reference value is 0 degrees.

SV:CH<x>:RF_PHASe:REFerence:POSition

This command sets or queries whether the channel Phase Reference is located at the Trigger position or at the Cursor A position.

Conditions Requires option SV-RFVT

Group Spectrum View

Syntax

```
SV:CH<x>:RF_PHASe:REFerence:POSition {TRIGger|CURSor}
SV:CH<x>:RF_PHASe:REFerence:POSition?
```

Related Commands SV:CH<x>:RF_PHASe:REFerence:TIMe, SV:CH<x>:RF_PHASe:REFerence:DEGrees, SV:RF_PHASe:REFerence:MASTer

Arguments

CH<x> is the channel number of the Phase vs. Time trace.

TRIGger sets the Phase Reference location to the Trigger position.

CURSor sets the channel Phase Reference location to the phase time set by SV:CH<x>:RF_PHASe:REFerence:TIMe, which defaults to the Cursor A position.

Examples

SV:CH1:RF_PHASe:REFerence:POSition TRIGger sets the Phase Reference location to the Trigger position.

SV:CH2:RF_PHASe:REFerence:POSition? might return SV:CH2:RF_PHASe:REFerence:POSition CURSor, indicating that the Phase Reference location is at the Cursor A position.

SV:CH<x>:RF_PHASe:REFerence:TIMe

This command sets or queries the channel Phase Reference time in seconds.

Conditions Requires option SV-RFVT

Group Spectrum View

Syntax

```
SV:CH<x>:RF_PHASe:REFerence:TIMe <NR3>
SV:CH<x>:RF_PHASe:REFerence:TIMe?
```

Related Commands SV:CH<x>:RF_PHASe:REFerence:DEGrees, SV:CH<x>:RF_PHASe:REFerence:POSition, SV:RF_PHASe:REFerence:MASTer

Arguments CH<x> is the channel number of the Phase vs. Time trace.

NR3 is the Phase Reference time, in seconds. This indicates the time at which the phase value set by SV:CH<x>:RF_PHASe:REFerence:DEGrees is applied. If the phase position set by SV:CH<x>:RF_PHASe:REFerence:POSition is TRIGger, then the phase time is fixed at 0 seconds and cannot be changed. If the phase position is CURSor, then the phase time may be set to any value, and is initialized to the position of Cursor A. If CH<x> is the master phase reference, then the time is used to calculate the phase values of all other channels.

Examples `SV:CH1:RF_PHASe:REFerence:TIMe 25e-9` sets the Phase Reference time to 25 ns.

`SV:CH2:RF_PHASe:REFerence:TIMe?` might return `SV:CH2:RF_PHASe:REFerence:TIMe 0`, indicating that the Phase Reference time is 0 seconds.

SV:CH<x>:RF_PHASe:WRAP:DEGrees

This command sets or queries the Phase Wrap limit for the Phase vs. Time trace for the specified channel.

Conditions Requires option SV-RFVT

Group Spectrum View

Syntax

```
SV:CH<x>:RF_PHASe:WRAP:DEGrees <NR3>
SV:CH<x>:RF_PHASe:WRAP:DEGrees?
```

Arguments CH<x> is the channel number of the Phase vs. Time trace.

NR3 sets the number of wrap degrees, from 180 degrees to infinity. When Phase Wrap is enabled, the phase values in the Phase vs Time waveform are constrained to be within ± the specified limit. Phase values below or above the limit are wrapped by repeatedly adding or subtracting 360 degrees until they are within range.

Examples `SV:CH2:RF_PHASe:WRAP:DEGrees 200` sets the Phase Wrap limit for the Channel 2 Phase vs. Time trace to 200 degrees.

`SV:CH1:RF_PHASe:WRAP:DEGrees?` might return `SV:CH1:RF_PHASe:WRAP:DEGrees 210`, indicating that the Phase Wrap limit for the Channel 1 Phase vs. Time trace is 210 degrees.

SV:CH<x>:RF_PHASe:WRAP:STATE

This command sets or queries whether Phase Wrap is applied to the Phase vs. Time trace for the specified channel.

Conditions Requires option SV-RFVT

Group Spectrum View

Syntax
```
SV:CH<x>:RF_PHASe:WRAP:STATE {ON|OFF}
SV:CH<x>:RF_PHASe:WRAP:STATE?
```

Arguments `CH<x>` is the channel number of the Phase vs. Time trace.

`ON` enables applying phase wrap on the specified Phase vs. Time channel trace.

`OFF` disables applying phase wrap on the specified Phase vs. Time channel trace.

Examples `SV:CH5:RF_PHASe:WRAP:STATE ON` turns on (enables)Phase Wrap for the Channel 5 Phase vs. Time trace.

`SV:CH1:RF_PHASe:WRAP:STATE?` might return `SV:CH1:RF_PHASe:WRAP:STATE OFF`, indicating that Phase Wrap is turned off (disabled) for the Channel 1 Phase vs. Time trace.

SV:CH<x>:SELect:RF_AVErage

This command sets or queries whether the Average trace is displayed for the specified spectrum trace channel in the Spectrum View.

Group Spectrum View

Syntax
```
SV:CH<x>:SELect:RF_AVErage {ON|1|OFF|0}
SV:CH<x>:SELect:RF_AVErage?
```

Arguments

CH<x> specifies the spectrum trace channel source.

ON, 1 enables the display of the Average trace for the specified spectrum trace channel.

OFF, 0 disables the display of the Average trace for the specified spectrum trace channel.

Examples

SV:CH3:SELect:RF_AVErage ON turns on the display of the Average trace for the channel 3 spectrum trace.

SV:CH1:SELect:RF_AVErage? might return 0, indicating that the Average trace for the channel 1 spectrum trace is turned off.

SV:CH<x>:SELect:RF_FREQuency

This command sets or queries whether the Frequency vs. Time trace for the specified channel is displayed in the Waveform View.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:SELect:RF_FREQuency {ON|OFF}
SV:CH<x>:SELect:RF_FREQuency?
```

Arguments

CH<x> is the channel number of the Frequency vs. Time trace.

ON enables display of the Frequency vs. Time trace.

OFF disables display of the Frequency vs. Time trace.

Examples

SV:CH4:SELect:RF_FREQuency ON turns on display of the Channel 4 Frequency vs. Time trace.

SV:CH1:SELect:RF_FREQuency? might return SV:CH1:SELect:RF_FREQuency OFF, indicating that the Channel 1 Frequency vs. Time trace is not displayed.

SV:CH<x>:SELect:RF_MAGnitude

This command sets or queries whether the Magnitude vs. Time trace for the specified channel is displayed in the Waveform View.

Conditions Requires option SV-RFVT

Group Spectrum View

Syntax
```
SV:CH<x>:SELect:RF_MAGnitude {ON|OFF}
SV:CH<x>:SELect:RF_MAGnitude?
```

Arguments `CH<x>` is the channel number of the Magnitude vs. Time trace.

`ON` enables display of the Magnitude vs. Time trace.

`OFF` disables display of the Magnitude vs. Time trace.

Examples `SV:CH4:SELect:RF_MAGnitude ON` turns on (enables) display of the Channel 4 Magnitude vs. Time trace.

`SV:CH1:SELect:RF_MAGnitude?` might return `SV:CH1:SELect:RF_MAGnitude OFF`, indicating that the Channel 1 Magnitude vs. Time trace is not displayed.

SV:CH<x>:SELect:RF_MAXHold

This command sets or queries whether the Max Hold trace is displayed for the specified spectrum trace channel in the Spectrum View.

Group Spectrum View

Syntax
```
SV:CH<x>:SELect:RF_MAXHold {ON|1|OFF|0}
SV:CH<x>:SELect:RF_MAXHold?
```

Arguments `CH<x>` specifies the spectrum trace channel source.

`ON, 1` enables the display of the Max Hold trace for the specified spectrum trace channel.

`OFF, 0` disables the display of the Max Hold trace for the specified spectrum trace channel.

Examples `SV:CH3:SELect:RF_MAXHold ON` turns on the display of the Max Hold trace for the channel 3 spectrum trace.

`SV:CH5:SELect:RF_MAXHold?` might return `1`, indicating that the Max Hold trace for the channel 5 spectrum trace is displayed.

SV:CH<x>:SELect:RF_MINHold

This command sets or queries whether the Min Hold trace is displayed for the specified spectrum trace channel in the Spectrum View.

Group Spectrum View

Syntax

```
SV:CH<x>:SELect:RF_MINHold {ON|1|OFF|0}
SV:CH<x>:SELect:RF_MINHold?
```

Arguments

`CH<x>` specifies the spectrum trace channel source.

`ON, 1` enables the display of the Min Hold trace for the specified spectrum trace channel.

`OFF, 0` disables the display of the Min Hold trace for the specified spectrum trace channel.

Examples

`SV:CH2:SELect:RF_MINHold ON` turns on the display of the Min Hold trace for the channel 2 spectrum trace.

`SV:CH1:SELect:RF_MINHold?` might return `0`, indicating that the Min Hold trace for the channel 1 spectrum trace is turned off.

SV:CH<x>:SELect:RF_NORMal

This command sets or queries whether the Normal trace is displayed for the specified spectrum trace channel in the Spectrum View.

Group Spectrum View

Syntax

```
SV:CH<x>:SELect:RF_NORMal {ON|1|OFF|0}
SV:CH<x>:SELect:RF_NORMal?
```

Arguments

`CH<x>` specifies the spectrum trace channel source.

`ON, 1` enables the display of the Normal trace for the specified spectrum trace channel.

`OFF, 0` disables the display of the Normal trace for the specified spectrum trace channel.

Examples `SV:CH3:SELect:RF_NORMal ON` turns on the display of the Average trace for the channel 3 spectrum trace.

`SV:CH1:SELect:RF_NORMal?` might return `0`, indicating that the Average trace for the channel 1 spectrum trace is turned off.

SV:CH<x>:SELect:RF_PHASe

This command sets or queries whether the Phase vs. Time trace for the specified channel is displayed in the Waveform View.

Conditions Requires option SV-RFVT

Group Spectrum View

Syntax
```
SV:CH<x>:SELect:RF_PHASe {ON|OFF}
SV:CH<x>:SELect:RF_PHASe?
```

Arguments `CH<x>` is the channel number of the Phase vs. Time trace.

`ON` enables display of the Phase vs. Time trace.

`OFF` disables display of the Phase vs. Time trace.

Examples `SV:CH4:SELect:RF_PHASe ON` turns on (enables) display of the Channel 4 Phase vs. Time trace.

`SV:CH1:SELect:RF_PHASe?` might return `SV:CH1:SELect:RF_PHASe OFF`, indicating that the Channel 1 Phase vs. Time trace is not displayed.

SV:CH<x>:SELTrace

This command sets or queries the spectrum trace type to show for the specified channel in the Spectrum View. Each channel's spectrum trace can display up to four traces; a Normal trace (default), a Max Hold trace, a Min Hold trace and an Average trace.

Group Spectrum View

Syntax
```
SV:CH<x>:SELTrace {NORMal|MAXHold|MINHold|AVErage}
SV:CH<x>:SELTrace?
```

Arguments

CH<x> specifies the analog channel to use as the Spectrum View source.

NORMal selects the Normal trace for the specified spectrum trace channel. If the specified spectrum trace channel does not have a Normal trace, this command is ignored.

MAXHold selects the Max Hold trace for the specified spectrum trace channel. If the specified spectrum trace channel does not have a Max Hold trace, this command is ignored.

MINHold selects the Min Hold trace for the specified spectrum trace channel. If the specified spectrum trace channel does not have a Min Hold trace, this command is ignored.

AVERage selects the Average trace for the specified spectrum trace channel. If the specified spectrum trace channel does not have an Average trace, this command is ignored.

Examples

SV:CH1:SELTrace MAXhold selects the Max Hold trace of the channel 1 spectrum trace in the Spectrum View.

SV:CH1:SELTrace? might return AVE, indicating that the Average trace of the channel 1 spectrum trace is currently selected in the Spectrum View.

SV:CH<x>:SQUELCH:STATE

This command sets or queries whether Squelch is enabled for the RF vs Time traces for the specified channel. The frequency vs. time and phase vs. time traces aren't meaningful when the transmitting signal is turned off, as they just show broadband noise that clutters up the display. With squelch, a threshold may be defined for the magnitude vs. time trace to indicate when the transmitter is on/off. Then the frequency and phase vs. time traces are shown only when the magnitude vs. time trace is above the threshold and they are blanked out when it's below the threshold.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:SQUELCH:STATE {ON|OFF}
SV:CH<x>:SQUELCH:STATE?
```

Related Commands

SV:CH<x>:SQUELCH:THReshold

Arguments

CH<x> is the channel number of the Frequency or Phase vs. Time trace.

ON enables (turns on) Squelch calculations.

OFF disables (turns off) Squelch calculations.

Examples

SV:CH6:SQUELCH:STATE ON turns on Squelch calculations for the RF vs. Time traces of Channel 6.

SV:CH1:SQUELCH:STATE? might return SV:CH1:SQUELCH:STATE OFF, indicating that Squelch calculations are turned off for the RF vs. Time traces of Channel 1

SV:CH<x>:SQUELCH:THReshold

This command sets or queries the Squelch threshold value for the RF vs Time traces for the specified channel.

Conditions

Requires option SV-RFVT

Group

Spectrum View

Syntax

```
SV:CH<x>:SQUELCH:THReshold <NR3>
SV:CH<x>:SQUELCH:THReshold?
```

Related Commands

SV:CH<x>:SQUELCH:STATE

Arguments

CH<x> is the channel number of the Magnitude vs. Time trace.

NR3 sets the threshold value, in volts, applied to the Magnitude vs. Time trace that determines whether or not to show the Frequency and Phase vs. Time traces for the same channel.

Examples

SV:CH5:SQUELCH:THReshold 0.5 sets the Squelch Threshold value to 0.5 volts for the RF vs Time traces of Channel 5.

SV:CH1:SQUELCH:THReshold? might return SV:CH1:SQUELCH:THReshold 1.25, indicating a Squelch Threshold value of 1.25 Volts for the RF vs. Time traces of Channel 1.

SV:CH<x>:UNIts

This command sets or queries the absolute logarithmic amplitude vertical scale units to show in the specified spectrum trace channel of the Spectrum View.

Group Spectrum View

Syntax
```
SV:CH<x>:UNIts {DBM|DBUW|DBMV|DBUV|DBMA|DBUA}
SV:CH<x>:UNIts?
```

Arguments `CH<x>` specifies the analog channel to use as the Spectrum View source.

`DBM` specifies Decibel milliwatts (dBm).

`DBUW` specifies Decibel microwatts (dBµW).

`DBMV` specifies Decibel millivolts (dBmV).

`DBUV` specifies Decibel microvolts (dBµV).

`DBMA` specifies Decibel milliamperes (dBmA).

`DBUA` specifies Decibel microamperes (dBµA).

Examples `SV:CH5:UNIts DBUV` sets the units to be used for the channel 5spectrum trace to decibel microvolts.

`SV:CH1:UNIts?` might return `DBM`, indicating that the vertical units for the channel 1 spectrum trace are decibel milliwatts.

SV:LOCKCenter

This command sets or queries whether the Center Frequency setting is locked across all channels in the Spectrum View.

Group Spectrum View

Syntax
```
SV:LOCKCenter {ON|1|OFF|0}
SV:LOCKCenter?
```

Related Commands CH<x>:SV:CENTERFrequency

Arguments `ON, 1` sets all spectrum traces channels in the Spectrum View window to use the same center frequency value. When the center frequency of any channel is

changed, the center frequency of all other channels is automatically changed to match that value.

OFF, 0 enables use of different center frequency values for each spectrum trace channel. The center frequencies of all channels are independent.

Examples

SV:LOCKCenter ON locks all spectrum trace channels to the same center frequency.

SV:LOCKCenter? might return 0, which indicates that each spectrum trace channel may have an independent center frequency setting.

SV:LOCKSpectrum

This command sets or queries whether the Spectrum Time value is locked across all spectrum trace channels in the Spectrum View.

Group

Spectrum View

Syntax

```
SV:LOCKSpectrum {ON|1|OFF|0}
SV:LOCKSpectrum?
```

Related Commands

CH<x>:SV:POSition

Arguments

ON, 1 sets all spectrum traces channels in the Spectrum View window to use the same Spectrum Time value. When the Spectrum Time of any channel is changed, the Spectrum Time of all other channels is automatically changed to match that value.

OFF, 0 enables use of different Spectrum Time values for each spectrum trace channel. The Spectrum Time of all channels are independent.

Examples

SV:LOCKSpectrum ON sets all spectrum trace channels to use the same spectrum time.

SV:LOCKSpectrum? might return 0, which indicates each channel may have an independent Spectrum Time setting.

SV:MARKER:PEAK:EXCURsion

This command sets or queries the minimum peak excursion value, in dB, for the Spectrum View trace peak markers.

Peak excursion refers to the minimum amount a spectrum signal needs to fall in amplitude between marked peaks to be considered another valid peak. If the peak excursion value is low, more peaks will tend to qualify as valid peaks and have associated markers. If the peak excursion value is high, fewer peaks will have associated markers.

Group Spectrum View

Syntax
```
SV:MARKER:PEAK:EXCURsion <NR3>
SV:MARKER:PEAK:EXCURsion?
```

Arguments <NR3> specifies the peak marker excursion value in dB. The range of values is 0.0 dB to 200.0 dB.

Examples SV:MARKER:PEAK:EXCURsion 12 sets the peak excursion value to 12 dB.

SV:MARKER:PEAK:EXCURsion? might return 30.00, indicating that a peak must exceed 30 dB change from other marked peaks to be marked with a peak indicator icon.

SV:MARKER:PEAK:MAXimum

This command sets or queries the maximum number of Spectrum View peak markers that can be placed on spectrum traces.

the Spectrum View window can show between 1 and 11 peak markers on all spectrum traces. The default is 5. To turn on the peak markers, use the command SV:MARKER:PEAK:STATE. The actual number of peak markers may be less than the maximum, depending on the threshold and excursion values and the spectral content of the RF signal. If more peaks than the maximum are detected that meet the threshold and excursion criteria, only the highest amplitude peaks will have markers placed on them.

Group Spectrum View

Syntax
```
SV:MARKER:PEAK:MAXimum <NR1>
SV:MARKER:PEAK:MAXimum?
```

Related Commands SV:MARKER:PEAK:STATE

Arguments <NR1> is an integer value that specifies the maximum number of peak markers to display on spectrum traces. he range of values is 1 to 11.

Examples

SV:MARKER:PEAK:MAXimum 3 sets the maximum number of peaks that can have markers to three.

SV:MARKER:PEAK:MAXimum? might return 7, indicating that the maximum number of peaks that can have markers is set to seven.

SV:MARKER:PEAK:STATE

This command sets or queries showing peak markers on spectrum traces in the Spectrum View window.

There are up to 11 markers. The maximum number of markers can be set using the command SV:MARKER:PEAK:MAXimum. The peak markers find amplitude peaks based upon threshold and excursion settings (set with the SV:MARKER:PEAK:EXCURsion and SV:MARKER:PEAK:THReshold commands.) Each peak marker has a readout associated with it. These can be absolute or delta readouts (set with the SV:MARKER:TYPe command.)

Group

Spectrum View

Syntax

```
SV:MARKER:PEAK:STATE {CH<x>}
SV:MARKER:PEAK:STATE?
```

Related Commands

SV:MARKER:PEAK:MAXimum

SV:MARKER:PEAK:EXCURsion

SV:MARKER:PEAK:THReshold

SV:MARKER:TYPe

Arguments

ON, 1 enables showing peak marker icons on spectrum trace waveforms.

OFF, 0 disables showing peak marker icons on spectrum trace waveforms.

Examples

SV:MARKER:PEAK:STATE 1 turns the peak markers on.

SV:MARKER:PEAK:STATE? might return 0, indicating that peak markers are turned off.

SV:MARKER:PEAK:THReshold

This command sets or queries the minimum peak threshold value required to mark a peak. Only peaks with an amplitude greater than the threshold value will

qualify for peak marker placement. Applies to all spectrum traces, and to each trace in its own vertical units.

Group Spectrum View

Syntax
```
SV:MARKER:PEAK:THReshold <NR3>
SV:MARKER:PEAK:THReshold?
```

Related Commands SV:CH<x>:UNIts

Arguments <NR3> specifies the peak marker threshold value. The range of values is –200.0 to 200.0.

Examples SV:MARKER:PEAK:THReshold -40 sets the peak marker threshold to -40 dBm.

SV:MARKER:PEAK:THReshold? might return -50.00, indicating that only peaks with an amplitude greater than -50 dBm qualify for peak marker placement.

SV:MARKER:REFERence (No Query Form)

This command sets the Center Frequency of the currently selected Spectrum View channel to the frequency indicated by the Reference Marker, in effect moving the Reference Marker to the center of the screen.

Group Spectrum View

Syntax
```
SV:MARKER:REFERence
```

Related Commands SV:CH<x>:UNIts

Examples SV:MARKER:REFERence moves the Reference Marker location to the center of the screen.

SV:MARKER:REFERence:AMPLITUDE? (Query Only)

This command queries the amplitude (vertical) value of the Reference Marker in user-set units. This value indicates the absolute amplitude of the Reference Marker, regardless of whether the other markers are manual or automatic.

Group Spectrum View

Syntax `SV:MARKER:REFERence:AMPLITUDE?`

Related Commands SV:CH<x>:UNIts

Returns <NR3> specifies the amplitude of the Reference Marker.

Examples `SV:MARKER:REFERence:AMPLITUDE?` might return `-65.6218`, indicating an amplitude value of -65.6218 dBm.

SV:MARKER:REFERence:FREQuency? (Query Only)

This command queries the frequency of the Reference Marker, in Hz, when the Spectrum View trace markers are on.

Group Spectrum View

Syntax `SV:MARKER:REFERence:FREQuency?`

Returns <NR3> specifies the frequency of the Reference Marker.

Examples `SV:MARKER:REFERence:FREQuency?` might return `1.6830E+9`, indicating that the Reference Marker frequency is 1.683 GHz.

SV:MARKER:TYPe

This command sets or queries the peak marker type (either DELTa or ABSolute).

An Absolute marker shows the frequency and amplitude at the location of the marker. A Delta marker shows the frequency and amplitude of the marker relative to the Reference Marker. The Reference Marker shows the absolute frequency and amplitude, regardless of this command. The marker amplitude measurements are in dBm for Absolute, or in dBc (dB below carrier amplitude) for Delta.

Group Spectrum View

Syntax

```
SV:MARKER:TYPe {DELta|ABSolute}
SV:MARKER:TYPe?
```

Arguments

`DELTa` specifies to display the frequency and amplitude of the peak markers relative to the Reference Marker. The relative amplitude is in dBc (dB below carrier amplitude); the relative frequency is in Hz.

`ABSolute` specifies to display the actual frequency and amplitude of each peak marker. The absolute amplitude is in user-set units; the absolute frequency is in Hz.

Examples

`SV:MARKER:TYPe DELta` changes the marker type to Delta.

`SV:MARKER:TYPe?` might return `ABSOLUTE`, indicating that the peak markers are measuring the absolute values of the spectrum trace at those positions.

SV:RBW

This command sets or queries the resolution bandwidth (RBW) when the RBW mode has been set to MANUAL (using the command SV:RBWMode).

The resolution bandwidth is the width of the narrowest measurable band of frequencies in a Spectrum View trace. By default, the RBW tracks the span value in a 1000:1 ratio. The RBW determines the level to which the instrument can resolve individual frequencies in the frequency domain. For example, if the input signal contains two carriers separated by 1 kHz, you will not be able to discriminate between them unless the RBW is less than 1 kHz.

Group

Spectrum View

Syntax

```
SV:RBW <NR3>
SV:RBW?
```

Related Commands

SV:RBWMode

Arguments

`<NR3>` specifies the width of the narrowest measurable band of frequencies in a Spectrum View trace, in Hz.

Examples

`SV:RBW 2.0E+04` sets the RBW to 20 kHz.

`SV:RBW?` might return `3.0E+6`, indicating that the RBW has been set to 3 MHz.

SV:RBWMode

This command sets or queries the resolution bandwidth (RBW) mode, either Automatic or Manual.

Group Spectrum View

Syntax

```
SV:RBWMode {AUTOmatic|MANual}
SV:RBWMode?
```

Related Commands SV:RBW

SV:SPANRBWRatio

Arguments `AUTOmatic` specifies the resolution bandwidth automatically as the span is changed. The default behavior is 1000:1, but you can set it to other values in a 1-2-5 sequence (e.g. 10000, 20000, 50000). To specify the RBW ratio that will be used when the mode is set to automatic, use the command SV:SPANRBWRatio.

`MANual` specifies to set the resolution bandwidth, independently from the span, using the command SV:RBW.

Examples `SV:RBWMode MANual` sets the RBW mode to Manual.

`SV:RBWMode?` might return `AUTO`, indicating that the RBW mode is set to Automatic mode (default).

SV:RF_PHASe:REFerence:MASTer

This command sets or queries the channel used as the Master Phase Reference.

Conditions Requires option SV-RFVT

Group Spectrum View

Syntax

```
SV:RF_PHASe:REFerence:MASTer {CH<x>|NONE}
SV:RF_PHASe:REFerence:MASTer?
```

Arguments `CH<x>` sets the specified channel as the Master Phase Reference source.

`NONE` indicates that there is no Master Phase Reference.

Examples `SV:RF_PHASe:REFerence:MASTer CH3` sets the Master Phase Reference source to Channel 3.

`SV:RF_PHASe:REFerence:MASTer?` might return `SV:RF_PHASe:REFerence:MASTer NONE`, indicating that there is no Master Phase Reference.

SV:SPAN

This command sets or queries the span setting for all channels in the Spectrum View. The span is the range of frequencies that can be observed centered on the center frequency. This is the width of the frequency domain trace, which is from the center frequency – ½span to the center frequency + ½ span.

Group Spectrum View

Syntax
```
SV:SPAN <NR3>
SV:SPAN?
```

Arguments `<NR3>` specifies the span value in Hz.

Examples `SV:SPAN 2E+9` sets the span to 2 GHz.

`SV:SPAN?` might return `10.0E+6`, indicating that the span is 10 MHz.

SV:SPANRBWRatio

This command sets or queries the ratio of the span to the resolution bandwidth (RBW) that will be used when the RBW Mode is set to AUTO.

The span is the width of the frequency domain trace in Hz, which is equal to the stop frequency minus the start frequency. The RBW is the width of the narrowest measurable band of frequencies in a frequency domain trace. The default RBW ratio is 1000 : 1.

Use the command SV:RBWMode to set the RBW Mode to Automatic.

Group Spectrum View

Syntax
```
SV:SPANRBWRatio <NR3>
SV:SPANRBWRatio?
```

Related Commands SV:RBWMode

Arguments <NR3> specifies the span-to-RBW ratio.

Examples SV:SPANRBWRatio 1000 sets the RBW to one one-thousandth of the span.

SV:SPANRBWRatio? might return 1.00E+3, which means the RBW has been set to one one-thousandth of the span.

SV:WINDOW

This command sets or queries the window type used by the windowing function of the Spectrum View. The windowing function is a Fast Fourier Transform (FFT) technique used to minimize the discontinuities between successive frames of an RF time domain signal. The default window type is Blackman-Harris.

Group Spectrum View

Syntax
```
SV:WINDOW {KAISerbessel|RECTangular|HAMMing|
HANNing|BLACkmanharris|FLATtop2}
SV:WINDOW?
```

Arguments KAISerbessel specifies the Kaiser-Bessel window type (a high or moderate resolution window).

RECTangular specifies the Rectangular window type (a window function is equivalent to multiplying all gate data by one).

HAMMing specifies the Hamming window type (a high or moderate resolution window based on a cosine series).

HANNing specifies the Hanning window type (a high or moderate resolution window based on a cosine series).

BLACkmanharris specifies the Blackman-Harris window type (a low-resolution (high dynamic range) window based on a cosine series).

FLATtop2 specifies the Flattop2 window type (a low-resolution (high dynamic range) window).

Examples SV:WINDOW BLACkmanharris sets the window to Blackman-Harris.

SV:WINDOW? might return KAISER, indicating that the window function is set to Kaiser-Bessel.

TEKSecure (No Query Form)

This command initializes, for the current user, both waveform and setup memories, overwriting any previously stored data.

Equivalent to invoking Teksecure from the Utility menu. This is a time-consuming operation (3 to 5 minutes) and the instrument is inoperable until the TekSecure operation is complete.

Group Miscellaneous

Syntax

```
TEKSecure
```

Examples `TEKSECURE` initializes both waveform and setup memories.

TIMe

This command sets the time in the form *hh:mm:ss* where hh refers to a two-digit hour number, mm refers to a two-digit minute number from 01 to 60, and ss refers to a two-digit second number from 01 to 60.

Conditions Time can only be changed in increments of 1 hour maximum.

There is a 24-hour total time change limit. Time changes cannot cross the 00:00:00 boundary.

On Linux, if an internet connection is present, any time changes are automatically overridden by the NTP server. This also resets the 24-hour time change limit.

On Windows, this command only succeeds if the scope application was started with administrator privileges.

Group Miscellaneous

Syntax

```
TIMe <QString>
TIMe?
```

Related Commands DATE?

Arguments <QString> is a quoted string representing the desired time.

Examples

`TIME"13:02:10"` sets the time to 1:02 p.m. and 10 seconds.

`TIME?` might return `:TIME "14:05:17"`, indicating the current time is set to 2:05 p.m. and 17 seconds.

TIMe:ZONe

This command sets the time zone to the one specified.

Group

Miscellaneous

Syntax

```
TIMe:ZONe <QString>
TIMe:ZONe?
```

Arguments

<QString> is a quoted string representing the desired time zone.

Examples

`TIME:ZONE "America/Yellowknife"` sets the time zone to UTC delta -7.

`TIME:ZONE?` might return `:TIME:ZONE "America/Los_Angeles"`.

TIMe:ZONe:UTCDELTa

This command sets or queries the time zone using the difference between the desired time zone and UTC.

Group

Miscellaneous

Syntax

```
TIMe:ZONe:UTCDELTa <NR3>
TIMe:ZONe:UTCDELTa?
```

Arguments

<NR3> is the specified number of hours difference between the desired time zone and UTC which is equivalent to GMT. The deltas supported are: -12.00, -11.00, -10.00, -9.30, -9.00, -8.30, -8.00, -7.00, -6.00, -5.00, -4.00, -3.30, -3.00, -2.00, -1.00, 0.0, 1.00, 2.00, 3.00, 3.30, 4.00, 4.30, 5.00, 5.30, 6.00, 6.30, 7.00, 8.00, 9.00, 9.30, 10.00, 10.30, 11.00, 11.30, 12.00

Examples

`TIME:ZONE:UTCDELTA -7.0e0` sets the time zone to America/Yellowknife.

`TIME:ZONE:UTCDELTA?` might return `:TIME:ZONE:UTCDELTA -8.0000`.

TOTaluptime? (Query Only)

Total number of hours the instrument has been turned on since the NV memory was last programmed, usually during the initial manufacturing process.

Group Miscellaneous

Syntax `TOTaluptime?`

Returns The total number of hours the instrument has been turned on since the NV memory was last programmed.

Examples `TOTALUPTIME?` might return :TOTALUPTIME 756 indicating the up time is 756 minutes.

TOUCHSCReen:CALibrate (No Query Form)

This command launches the touch screen calibration procedure. This command is equivalent to tapping the Calibrate touch screen control in the Utility->Self Test menu.

Group Self Test

Syntax `TOUCHSCReen:CALibrate STARt`

Arguments `STARt` launches the touch screen calibration procedure.

Examples `TOUCHSCREEN:CALIBRATE START` launches the touch screen calibration procedure.

TOUCHSCReen:STATe

This sets or queries the enabled state of the touch screen only. This command is equivalent to pushing the Touch Off button on the front panel.

To completely disable front panel operation, use the command `LOCK ALL`. To re-enable the front panel, send the command `LOCK NONE`.

Group Miscellaneous

Syntax

```
TOUCHSCReen:STATe {0|1|OFF|ON}
TOUCHSCReen:STATe?
```

Related Commands LOCk

Arguments

0 disables the touch screen.

ON enables the touch screen.

OFF disables the touch screen.

Examples

TOUCHSCReen:STATe OFF disables the touch screen.

TOUCHSCReen:STATe? might return :TOUCHSCREEN:STATE 1 indicating the touch screen is enabled.

*TRG (No Query Form)

Performs a group execute trigger on commands defined by *DDT.

Group Miscellaneous

Syntax

```
*TRG
```

Related Commands *DDT

Examples

*TRG immediately executes all commands that have been defined by *DDT.

TRIGger

This command forces a trigger event to occur. The query returns the current trigger parameters for the instrument.

Group Trigger

Syntax

```
TRIGger FORCe
TRIGger?
```

Arguments FORCe creates a trigger event. If TRIGger:STATE is set to READy, the acquisition will complete. Otherwise, this command will be ignored. This is equivalent to pressing the Force button on the front panel.

Examples TRIGGER FORCE forces a trigger event to occur.

TRIGGER? returns the current trigger parameters for the instrument.

TRIGger:{A|B|B:RESET}

This command sets the A, B, or B Reset trigger level automatically to 50% of the range of the minimum and maximum values of the trigger input signal. The query returns current trigger parameters. The trigger level is the voltage threshold through which the trigger source signal must pass to generate a trigger event. This command is equivalent to pushing the LEVEL knob on the front panel.

Group Trigger

Syntax
```
TRIGger:{A|B|B:RESET} SETLevel
TRIGger:{A|B|B:RESET}?
```

Arguments SETLevel sets the trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

Examples TRIGger:A SETLEVEL sets the trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

TRIGger:A? returns current trigger parameters.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition

This command specifies a field or condition for an ARINC429 bus to trigger on. The bus number is specified by x.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition
{SOW|LABel|DATa|LABELANDDATA |EOW|ERRor}
```

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition?
```

Arguments

SOW specifies triggering on the first bit of a word.

LABel specifies triggering on a matching label.

DATa specifies triggering on matching packet data field(s).

LABELANDDATA specifies triggering on a matching label and matching packet data field(s).

EOW specifies triggering on the 32nd bit of a word.

ERRor specifies triggering on a specified error condition.

NOTE. *The type of error triggered on is specified by TRIGger:{A|B}:BUS:B<x>:ARINC429A:ERRTYPe.*

Examples

TRIGGER:A:BUS:B1:ARINC429A:CONDITION DATA specifies triggering on packets that contain matching data field(s).

TRIGGER:A:BUS:B1:ARINC429A:CONDITION? might return SOW, indicating that the bus is triggering on the first bit of each word in the pacekt.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:HIVALue

This command sets or queries the high value when trigger on an ARINC429 data field. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA, and the data qualifier must be INrange or OUTrange.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:HIVALue?
```

Arguments

<QString> is the label value.

NOTE. *The size of the QString is dependent on the data field format selected using BUS:Bx:ARINC429A:DATAFORmat. Also, the stored QString is reset to its default value whenever the data field format is changed.*

Examples

`TRIGGER:A:BUS:B1:ARINC429A:DATA:HIVALUE "XXXXXXXXXXXXXX1000"` sets the value to XXXXXXXXXXXXXX1000.

`TRIGGER:A:BUS:B1:ARINC429A:DATA:HIVALUE?` might return "XXXXXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:QUALifier

This command sets or queries the qualifier to be used when triggering on data in the DATA field for an ARINC429 bus signal. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:
QUALifier {EQual|UNEQual|LESSthan|MOREthan
|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:QUALifier?
```

Arguments

Arguments are the available data qualifiers.

NOTE. *The trigger qualifier only applies to the bits defined as the data field via the bus data field format specifier (using* `BUS:B<x>:ARINC429A:DATAFORmat`*)*

Examples

`TRIGGER:A:BUS:B1:ARINC429A:DATA:QUALIFIER LESSTHAN` sets the data qualifier to less than.

`TRIGGER:A:BUS:B1:ARINC429A:DATA:QUALIFIER?` might return EQUAL, indicating that the data qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:VALue

This command sets or queries the low value when triggering on an ARINC429 data field. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:VALue?
```

Arguments <QString> is the label value.

NOTE. *The size of the QString is dependent on the data field format selected using* *BUS:B<x>:ARINC429A:DATAFORMat. Also, the stored QString is reset to its default value whenever the data field format is changed.*

Examples TRIGGER:A:BUS:B1:ARINC429A:DATA:VALUE "XXXXXXXXXXXXXXX1000" sets the value to XXXXXXXXXXXXXXX1000.

TRIGGER:A:BUS:B1:ARINC429A:DATA:VALUE? might return "XXXXXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:ERRTYPe

This command sets or queries the error type when triggering on an ARINC429 bus signal. The bus number is specified by x. The trigger condition must be set to ERRor.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:ERRTYPe
{ANY|PARity|WORD|GAP}
TRIGger:{A|B}:BUS:B<x>:ARINC429A:ERRTYPe?
```

Arguments ANY sets the error type to match any of the other available error types.

PARity sets the error type to match on parity errors (parity value results in even parity count for a word).

WORD sets the error type to match on word errors (any unframed or unknown decode data).

GAP sets the error type to match on gap violations (less than 4 bits idle time between two packets on the bus).

Examples `TRIGGER:A:BUS:B1:ARINC429A:ERRTYPE PARITY` sets the error type to match on parity errors.

`TRIGGER:A:BUS:B1:ARINC429A:ERRTYPE?` might return ANY, indicating that any error condition will produce a match.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue

This command sets or queries the high value when triggering on an ARINC429 label field. The bus number is specified by x. The trigger condition must be set to LABel, and the label qualifier must be INrange or OUTrange.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue?
```

Arguments `<QString>` is the label value.

Examples `TRIGGER:A:BUS:B1:ARINC429A:LABEL:HIVALUE "XXXX1010"` sets the value to XXXX1010.

`TRIGGER:A:BUS:B1:ARINC429A:LABEL:HIVALUE?` might return "XXXXXXXX", indicating that the value is XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:QUALifier

This command sets or queries the qualifier to be used when triggering on label data for an ARINC429 bus signal. The bus number is specified by x. The trigger condition must be set to LABel or LABELANDDATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:
QUALifier {EQual|UNEQual|LESSthan|MOREthan
|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:QUALifier?
```

Arguments

Arguments are the available data qualifiers.

NOTE. *If the trigger condition is set to LABELANDDATA, the label qualifier will be locked to Equal until the trigger condition is changed again.*

Examples

`TRIGGER:A:BUS:B1:ARINC429A:LABEL:QUALIFIER LESSTHAN` sets the label qualifier to less than.

`TRIGGER:A:BUS:B1:ARINC429A:LABEL:QUALIFIER?` might return EQUAL, indicating that the label qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:VALue

This command sets or queries the low value when triggering on an ARINC429 label field. The bus number is specified by x. The trigger condition must be set to LABel or LABELANDDATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:VALue?
```

Arguments

`<QString>` is the label value.

Examples

`TRIGGER:A:BUS:B1:ARINC429A:LABEL:VALUE "XXXX1010"` sets the value to XXXX1010.

`TRIGGER:A:BUS:B1:ARINC429A:LABEL:VALUE?` might return "XXXXXXXX", indicating that the value is XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue

This command sets or queries the label when triggering on an ARINC429 SDI field. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA, and the data format must be set to DATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue?
```

Arguments <QString> is the label value.

Examples TRIGGER:A:BUS:B1:ARINC429A:SDI:VALUE "X0" sets the value to X0.

TRIGGER:A:BUS:B1:ARINC429A:SDI:VALUE? might return "XX", indicating that the value is XX.

TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue

This command sets or queries the label value when triggering on an ARINC429 SSM field. The bus number is specified by x. The trigger condition must be set to DATa or LABELANDDATA, and the data format must be set to DATA or SDIDATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue?
```

Arguments <QString> is the label value.

NOTE. *The SSM field is only present when the selected data field format is DATA or SDIDATA (using* BUS:B<x>:ARINC429A:DATAFORmat*). Also, the stored QString is reset to its default value whenever the data field format is changed.*

Examples TRIGGER:A:BUS:B1:ARINC429A:SSM:VALUE "X0" sets the value to X0.

TRIGGER:A:BUS:B1:ARINC429A:SSM:VALUE? might return "XX", indicating that the value is XX.

TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition

This command sets the condition (word select, start of frame, or matching data) to be used when triggering on an audio bus signal. The bus number is specified by x.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition {SOF|DATa}
TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition?
```

Arguments SOF enables triggering on a word select or start of frame (depending on Audio Type).

DATa enables triggering on matching data.

Examples TRIGger:A:BUS:B1:AUDio:CONDition SOF sets the condition to start of frame.

TRIGger:A:BUS:B1:AUDio:CONDition? might return :TRIGGER:A:BUS:B1:AUDIO:CONDITION DATA indicating the condition is data.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDMVALue

This command sets or queries the binary data string for the high data word to be used when triggering on an TDM audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDMVALue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDMVALue?
```

Arguments <QString> is the binary data string for the high data word to be used when triggering on an TDM audio bus signal.

Examples `TRIGger:A:BUS:B1:AUDio:DATa:HITDMVALue "1100"` sets the high value to 1100.

`TRIGger:A:BUS:B1:AUDio:DATa:HITDMVALue?` might return :TRIGGER:A:BUS:B1:AUDIO:DATA:HITDMVALUE "XXXXXXXXXXXXXXXXXXX1010" indicating the high value is 1010.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue

This command sets the upper word value to be used when triggering on an audio bus signal. The trigger condition must be set to DATA using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by x.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue?
```

Arguments <QString> is the upper word value to be used when triggering on an audio bus signal.

Examples `TRIGger:A:BUS:B1:AUDio:DATa:HIVALue "11001101"` sets the hi value to XXXXXXXXXXXXXXXX11001101.

TRIGger:A:BUS:B1:AUDio:DATa:HIVALue? might return :TRIGGER:A:BUS:B1:AUDIO:DATA:HIVALUE "XXXXXXXXXXXXXXXXXXXXXXXX" indicating the hi value is set to XXXXXXXXXXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet

This command sets the data offset value to be used when triggering on an audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by x.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet?
```

Arguments <NR1> is the data offset value.

Examples `TRIGger:A:BUS:B1:AUDio:DATa:OFFSet 2` sets the data offset to 2.

`TRIGger:A:BUS:B1:AUDio:DATa:OFFSet?` might return `:TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET 1` indicating the data offset value is 1.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier

This command sets the qualifier to be used when triggering on an audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by x.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier {LESSthan|
MOREthan|EQual|UNEQual|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier?
```

Arguments

LESSthan sets the qualifier to less than.

MOREthan sets the qualifier to greater than.

EQual sets the qualifier to equal.

UNEQual sets the qualifier to not equal.

LESSEQual sets the qualifier to less than or equal.

MOREEQual sets the qualifier to greater than or equal.

INrange sets the qualifier to in range.

OUTrange sets the qualifier to out of range.

Examples

TRIGger:A:BUS:B1:AUDio:DATa:QUALifier LESSTHAN sets the qualifier to less than.

TRIGger:A:BUS:B1:AUDio:DATa:QUALifier? might return :TRIGGER:A:BUS:B1:AUDIO:DATA:QUALIFIER EQUAL indicating the qualifies is set to equal.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:TDMVALue

This command sets or queries the binary data string for the single or low data word to be used when triggering on an TDM audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B{x}:AUDio:CONDition.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:TDMVALue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:TDMVALue?
```

Arguments

<QString> is the binary data string for the single or low data word to be used when triggering on an TDM audio bus signal.

Examples

TRIGger:A:BUS:B1:AUDio:DATa:TDMVALue "1100" sets the TDMVALUE to 1100.

TRIGger:A:BUS:B1:AUDio:DATa:TDMVALue? might return :TRIGGER:A:BUS:B1:AUDIO:DATA:TDMVALUE "XXXXXXXXXXXXXXXXXXXX1010" indicating the TDM value is 1010.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue

This command sets the lower word value to be used when triggering on an audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by x.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue?
```

Arguments <QString> specifies the trigger data lower word.

Examples `TRIGger:A:BUS:B1:AUDio:DATa:VALue "11001100101"` sets the data value to XXXXXXXXXXXXX11001100101.

`TRIGger:A:BUS:B1:AUDio:DATa:VALue?` might return `:TRIGGER:A:BUS:B1:AUDIO:DATA:VALUE "XXXXXXXXXXXXXXXXXXXXXXXXX"` indicating the data value is XXXXXXXXXXXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD

This command sets the alignment of the data (left, right or either) to be used to trigger on an audio bus signal. The trigger condition must be set to DATa using TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition.

The bus number is specified by x.

Conditions Requires option 5-SRAUDIO or SUP5-SRAUDIO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD {EITher|LEFt|RIGht}
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD?
```

Arguments

`EITher` aligns the trigger data to either left or right.

`LEFt` aligns the trigger data to the left.

`RIGht` aligns the trigger data to the right.

Examples

`TRIGger:A:BUS:B1:AUDio:DATa:WORD LEFt` sets the word alignment to the left.

`TRIGger:A:BUS:B1:AUDio:DATa:WORD?` might return `:TRIGGER:A:BUS:B1:AUDIO:DATA:WORD EITHER` indicating the trigger data is aligned to either left or right.

TRIGger:{A|B}:BUS:B<x>:CAN:CONDition

This command sets the condition (start of frame, frame type, identifier, matching data, EOF, missing ACK field, bit-stuffing error) to be used when triggering on a CAN bus signal. The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:CONDition
{SOF|FRAMEtype|IDentifier|DATa|IDANDDATA|EOF|ERRor|FDBITS}
TRIGger:{A|B}:BUS:B<x>:CAN:CONDition?
```

Arguments

`SOF` enables triggering on the start of frame.

`FDBITS` enables triggering on the values of the BRS and ESI bits in an FD packet.

`FRAMEtype` enables triggering on the type of frame.

`IDentifier` enables triggering on a matching identifier.

`DATa` enables triggering on matching data.

`IDANDDATA` enables triggering on a matching identifier and matching data.

`EOF` enables triggering on the end of frame.

`ERRor` enables triggering on a specified error condition.

Examples

`TRIGGER:A:BUS:B1:CAN:CONDITION?` might return :TRIGGER:A:BUS:B1:CAN:CONDITION EOF indicating an end of file condition.

TRIGGER:A:BUS:B1:CAN:CONDITION DATA enables triggering on matching CAN data.

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:DIRection

This command sets the data direction (read, write or "nocare") to be used to search on a CAN bus signal. The trigger condition must be set to IDentifier (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:DIRection
{READ|WRITE|NOCARE}
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:DIRection?
```

Arguments

READ sets the CAN data direction to READ.

WRITE sets the CAN data direction to WRITE.

NOCARE sets the CAN data direction to either.

Examples

TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION WRITE sets the CAN data direction to Write.

TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION? might return :TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION NOCARE indicating the data direction can be either read or write.

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:OFFSet

This command sets or queries the data offset value, in bytes, to use when triggering on the CAN data field. The bus number is specified by x. The trigger condition must be set to DATA or IDANDDATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:OFFSet?
```

Related Commands

BUS:B<x>:CAN:STANDard

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue

Arguments

<NR1> is an integer whose minimum and default values are -1 (don't care), and the maximum is up to 7 (for CAN 2.0) or up to 63 (for ISO CAN FD and Non-ISO CAN FD).

The maximum is dependent on the number of bytes being matched and the CAN standard selected. Its value is calculated as [Absolute Maximum] - [Data Match Size]. For CAN 2.0, the absolute maximum is 8 bytes. For ISO CAN FD and Non-ISO CAN FD, the absolute maximum is 64 bytes. The minimum data match size is 1 byte, which produces the ranges listed above. Increasing the data match size above 1 byte will adjust the range of valid data offset values accordingly.

Examples

`TRIGGER:A:BUS:B1:CAN:DATA:OFFSET 5` sets the CAN data offset to 5 bytes.

`TRIGGER:A:BUS:B1:CAN:DATA:OFFSET?` might return 7, indicating the CAN data offset is 7 bytes.

If the CAN standard is set for CAN 2.0, and the trigger data size is set to 3, the maximum value for the data offset will be 5 (8 - 3 = 5).

If the CAN standard is set for ISO CAN FD or Non-ISO CAN FD, and the trigger data size is set to 8, the maximum value for the data offset will be 56 (64 - 8 = 56).

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:QUALifier

This command sets the qualifier (<, >, =, ≠, ≤, ≥) to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:QUALifier
{LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual}
```

```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:QUALifier?
```

Arguments

`LESSthan` sets the instrument to trigger when the data is less than the qualifier value.

`MOREthan` sets the instrument to trigger when the data is greater than the qualifier value.

`EQual` sets the instrument to trigger when the data is equal to the qualifier value.

`UNEQual` sets the instrument to trigger when the data is not equal to the qualifier value.

`LESSEQual` sets the instrument to trigger when the data is less than or equal to the qualifier value.

`MOREEQual` sets the instrument to trigger when the data is greater than or equal to the qualifier value.

Examples

`TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER LESSTHAN` sets the instrument to trigger when the data is less than the qualifier value.

`TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER?` might return `:TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER LESSTHAN`, indicating that the instrument is set to trigger when the data is less than the qualifier value.

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe

This command sets the length of the data string, in bytes, to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe?
```

Arguments

`<NR1>` is the length of the data string in bytes.

Examples `TRIGger:A:BUS:B1:CAN:DATa:SIZe 2` sets the data size to 2 bytes.

`TRIGger:A:BUS:B1:CAN:DATa:SIZe?` might return `:TRIGGER:A:BUS:B1:CAN:DATA:SIZE 1` indicating the data size is set to 1 byte.

TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue

This command sets the binary data value to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue?
```

Arguments `<QString>` is the data value in binary format. The only allowed characters in the `QString` are 0, 1, and X.

Examples `TRIGGER:A:BUS:B1:CAN:DATA:VALUE "1011"` sets the CAN data value to 1011.

`TRIGGER:A:BUS:B1:CAN:DATA:VALUE?` might return `:TRIGGER:A:BUS:B1:CAN:DATA:VALUE "XXXXXXXX"` indicating the data valule is set to XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:CAN:ERRType

This command sets or queries the type of error condition for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to ERRor.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:ERRType
{ACKMISS|BITSTUFFing|FORMERRor|ANYERRor}
TRIGger:{A|B}:BUS:B<x>:CAN:ERRType?
```

Arguments

ACKMISS specifies triggering on a missing ACK field.

BITSTUFFing specifies triggering on a bit stuffing error.

FORMERRor specifies triggering on a CAN FD form error. To use this option, the CAN standard must be set to FDISO or FDNONISO.

ANYERRor specifies triggering on any error type.

Examples

TRIGGER:A:BUS:B1:CAN:ERRTYPE ACKMISS specifies triggering on any missing ACK fields.

TRIGGER:A:BUS:B1:CAN:ERRTYPE? might return ANYERROR, indicating that the bus is triggering on all error types.

TRIGger:{A|B}:BUS:B<x>:CAN:FD:BRSBit

This command sets or queries the value of the bit rate switch bit (BRS bit) for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:FD:BRSBit {ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:CAN:FD:BRSBit?
```

Arguments

ONE filters CAN FD packets to only match those where the BRS bit has a value of 1 (fast data enabled).

ZERo filters CAN FD packets to only match those where the BRS bit has a value of 0 (fast data disabled).

NOCARE disables filtering of CAN FD packets on the BRS bit.

Examples

TRIGGER:A:BUS:B1:CAN:FD:BRSBit ONE specifies filtering CAN FD packets for those where the BRS bit has a value of 1.

TRIGGER:A:BUS:B1:CAN:FD:BRSBIT? might return NOCARE, indicating that CAN FD packets are not being filtered based on the BRS bit value.

TRIGger:{A|B}:BUS:B<x>:CAN:FD:ESIBit

This command sets or queries the value of the error state indicator bit (ESI bit) for a CAN bus to triggering on. The bus number is specified by x. The trigger condition must be set to FDBITS, and the CAN standard must be FDISO or FDNONISO.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:FD:ESIBit {ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:CAN:FD:ESIBit?
```

Arguments

ONE filters CAN FD packets to only match those where the ESI bit has a value of 1 (recessive).

ZERo filters CAN FD packets to only match those where the ESI bit has a value of 0 (dominant).

NOCARE disables filtering of CAN FD packets on the ESI bit.

Examples

TRIGGER:A:BUS:B1:CAN:FD:ESIBit ONE specifies filtering CAN FD packets for those where the ESI bit has a value of 1.

TRIGGER:A:BUS:B1:CAN:FD:ESIBIT? might return NOCARE, indicating that CAN FD packets are not being filtered based on the ESI bit value.

TRIGger:{A|B}:BUS:B<x>:CAN:FRAMEtype

This command sets the frame type (data, remote, error or overload) to be used when triggering on a CAN bus signal. The trigger condition must be set to FRAMEtype (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). B<x> is the bus number.The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:CAN:FRAMEtype
{DATa|REMote|ERRor|OVERLoad}
TRIGger:{A|B}:BUS:B<x>:CAN:FRAMEtype?
```

Arguments `DATa` specifies a data frame type.

`REMote` specifies a remote frame type.

`ERRor` specifies an error frame type.

`OVERLoad` specifies an overload frame type.

Examples `TRIGGER:A:BUS:B1:CAN:FRAMETYPE DATA` sets the CAN trigger frame type to DATA.

`TRIGGER:A:BUS:B1:CAN:FRAMETYPE?` might return `:TRIGGER:A:BUS:B1:CAN:FRAMETYPE DATA` indicating the frame type is data.

TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:MODe

This command sets the addressing mode (standard or extended format) to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:MODe
{STandard|EXTended}
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:MODe?
```

Arguments `STandard` specifies the standard addressing mode.

`EXTended` specifies the extended addressing mode.

Examples `TRIGger:A:BUS:B1:CAN:IDentifier:MODe EXTENDED` sets the addressing mode to extended.

TRIGger:A:BUS:B1:CAN:IDentifier:MODe? might return :TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE STANDARD indicating the address mode is standard.

TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:VALue

This command sets the binary address value to be used when triggering on a CAN bus signal. The trigger condition must be set to IDANDDATA or DATa (using TRIGger:{A|B}:BUS:B<x>:CAN:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:VALue?
```

Arguments <QString> is up to 29 bits specifying the binary identifier value. The only allowed characters in the QString are 0, 1, and X.

Examples TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE "1011" sets the identifier value to 1011.

TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE? might return :TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE "XXXXXXXXXXX" indicating the identifier values is XXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

This command specifies a field or condition within an Ethernet frame to trigger on. The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition
{SFD|MACADDRess|MACLENgth|IPHeader|TCPHeader|DATa|EOP|
```

IDLe|FCSError|QTAG}

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition?

Related Commands Most of the other TRIGger:A:BUS:B<x>:ETHERnet commands are impacted by the setting of this command.

Arguments SFD — Start of frame delimiter.

MACADDRess — MAC addresses field.

MACLENgth — MAC length/type field.

IPHeader — IP header field. This argument is only available when PROTOCOL is set to IPv4.

TCPHeader — TCP header field. This argument is only available when PROTOCOL is set to IPv4.

DATa — TCP/IPv4 or MAC protocol client data field. If the protocol is set to OTHER, then DATa refers to the MAC client data.

EOP — End of Packet field.

IDLe — Idle field.

FCSError — Frame Check Sequence Error (CRC) field.

QTAG — IEEE 802.1Q (VLAN) control information field. In order to use QTAG as a trigger condition, the frame type must be set to QTAG).

Examples TRIGger:A:BUS:B1:ETHERnet:CONDition MACADDRess sets the trigger field to MACADDRess.

TRIGger:A:BUS:B1:ETHERnet:CONDition? might return DATA.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue

When the Ethernet trigger condition is set to DATa, and the qualifier is set to either INrange or OUTrange, this command specifies the upper data value of the range. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue to specify the lower limit of the range.) The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

Arguments <QString> is a quoted string whose length varies depending on the size setting, up to 32 bits. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe to specify the size.) The allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples `TRIGger:A:BUS:B1:ETHERnet:DATa:HIVALue "XXXXXXXXXXXXXXXXXXXXXXXX00001000"` sets the upper limit of the range to XXXXXXXXXXXXXXXXXXXXXXXX00001000(when the trigger condition is set to `DATa`, and the qualifier is set to `INrange` or `OUTrange)`.

`TRIGger:A:BUS:B1:ETHERnet:DATa:HIVALue?` might return `"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"`

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet

When the Ethernet trigger condition is set to `DATa`, this command specifies where in the data field to look for the data trigger value. It specifies the offset into the data field, in bytes, where the value will be matched. The default is -1 (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue

Arguments <NR1> is an integer whose minimum and default values are -1 (don't care) and maximum is 1,499.

Examples TRIGger:A:BUS:B1:ETHERnet:DATa:OFFSet 36 sets the data offset to 36 bytes.

TRIGger:A:BUS:B1:ETHERnet:DATa:OFFSet? might return –1, indicating that the data offset value is the default value, -1, meaning "don't care".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:QUALifier

This command sets or queries the qualifier to be used when triggering on an Ethernet bus signal. The trigger condition must be set to DATa. The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:QUALifier {EQual|
UNEQual|LESSthan|MOREthan|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:QUALifier?
```

Arguments

LESSthan sets the qualifier to less than.

MOREthan sets the qualifier to greater than.

EQual sets the qualifier to equal.

UNEQual sets the qualifier to not equal.

LESSEQual sets the qualifier to less than or equal.

MOREEQual sets the qualifier to greater than or equal.

INrange sets the qualifier to in range.

OUTrange sets the qualifier to out of range.

Examples TRIGger:A:BUS:B1:ETHERnet:DATa:QUALifier LESSTHAN sets the qualifier to less than.

TRIGger:A:BUS:B1:ETHERnet:DATa:QUALifier? might return :TRIGGER:A:BUS:B1:ETHERNET:DATA:QUALIFIER EQUAL indicating the qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe

When the Ethernet trigger condition is set to DATa, this command specifies the number of contiguous TCP/IPv4/MAC client data bytes to trigger on. The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

Arguments The minimum and default values are 1 and maximum is 16, except when the qualifier is set to Inside Range or Outside Range. In these cases, the maximum size is 4.

Examples TRIGger:A:BUS:B1:ETHERnet:DATa:SIZe 4 sets the instrument to trigger on 4 contiguous data bytes.

TRIGger:A:BUS:B1:ETHERnet:DATa:SIZe? might return 6, indicating that the instrument is set to trigger on 6 contiguous data bytes.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue

When the Ethernet trigger condition is set to DATa, and the qualifier is set to LESSthan, MOREthan, EQual, UNEQual, LESSEQual or MOREEQual, this command specifies the value to trigger on. When the Ethernet trigger condition is set to DATa, and the qualifier is set to INrange or OUTrange, this command specifies the lower limit of the range. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue to set the upper limit of the range.) The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:OFFSet

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:SIZe

TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:HIVALue

Arguments QString is a quoted string where the allowable characters are 0, 1, and X. The allowable number of characters depends on the setting for size (using TRIGger:A:BUS:B<x>:ETHERnet:DATa:SIZe). The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples `TRIGger:A:BUS:B1:ETHERnet:DATa:VALue "00001000"` sets the binary data to trigger on to 00001000, assuming the qualifier is set to `LESSthan`, `MOREthan`, `EQual`, `UNEQual`, `LESSEQual` or `MOREEQual`, and `DATa:SIZe` is set to 1 byte.

`TRIGger:A:BUS:B1:ETHERnet:DATa:VALue "00001000"` sets the lower limit of the range to 00001000, assuming the qualifier is set to `INrange` or `OUTrange`, and`DATa:SIZe` is set to 1 byte.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue

When the Ethernet trigger condition is set to `IPHeader`, this command specifies the value of the 32–bit destination address that is to be used in the trigger (along with the source address and protocol value). The IP destination address is a standard IP address such as 192.168.0.1. The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:
VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:
VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:IPHeader:DESTinationaddr:VALue "00011001001000010110100000000001" sets the IP destination address to trigger on to 192.168.0.1.

TRIGger:A:BUS:B1:ETHERnet:IPHeader:DESTinationaddr:VALue might return "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX01".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue

When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 8–bit protocol field that is to be used in the trigger (along with the source and destination addresses). The default is all X's (don't care). The bus number is specified by x.

NOTE. *Commonly used protocol values are 1 (ICMP), 2 (IGMP), 6 (TCP) and 17 (UDP).*

Conditions

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue

Arguments

<QString> is a quoted string of up to 8 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGGER:A:BUS:B1:ETHERNET:IPHEADER:PROTOCOL:VALUE "01010010" would set the value to be used in the trigger to 01010010.

TRIGger:A:BUS:B1:ETHERnet:IPHeader:PROTOcol:VALue? might return "XXXXXXXX".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue

When the Ethernet trigger condition is set to IPHeader, this command specifies the value of the 32-bit source address that is to be used in the trigger (along with the destination address and protocol value). The IP source address is a standard IP address such as 192.168.0.1. The default is all X's (don't care). The bus number is specified by x.

Conditions

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:SOUrceaddr:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:DESTinationaddr:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:PROTOcol:VALue

Arguments

<QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

`TRIGger:A:BUS:B1:ETHERnet:IPHeader:SOUrceaddr:VALue "00011001001000010110100000000001"` sets the IP source address to trigger on to 192.168.0.1.

`TRIGger:A:BUS:B1:ETHERnet:IPHeader:SOUrceaddr:VALue` might return "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX01".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:VALue

When the Ethernet trigger condition is set to `MACADDress`, this command specifies the 48–bit MAC destination address that is to be used in the trigger (along with the source address value). The default is all X's (don't care). The bus number is specified by x.

NOTE. *MAC Addresses are 48–bit values such as 08:00:11:1E:C9:AE hex.*

Conditions

Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:
VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:
VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:SOUrce:VALue

Arguments

`<QString>` is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

`TRIGger:A:BUS:B1:ETHERnet:MAC:ADDRess:DESTination:VALue "XXXXXXXXXX0101011111110000000111101010101010101000"` would set the MAC destination address to trigger on to XX:35:FC:07:AA:C8 hex.

`TRIGger:A:BUS:B1:ETHERnet:MAC:ADDRess:DESTination:VALue?` might return "XXXXXXXXXX0101011111110000000111101010101010101000".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:SOUrce:VALue

When the Ethernet trigger condition is set to MACADDress, this command specifies the 48–bit MAC source address value that is to be used in the trigger (along with the destination address value). The default is all X's (don't care). The bus number is specified by x.

NOTE. *MAC Addresses are 48–bit values such as 08:00:11:1E:C9:AE hex.*

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:SOUrce:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:SOUrce:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:ADDRess:DESTination:VALue

Arguments <QString> is a quoted string of up to 48 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples TRIGger:A:BUS:B1:ETHERnet:MAC:ADDRess:SOUrce:VALue "XXXXXXXXXX010101111111000000011110101010101010000" would set the MAC destination address to trigger on to XX:35:FC:07:AA:C8 hex.

TRIGger:A:BUS:B1:ETHERnet:MAC:ADDRess:SOUrce:VALue? might return "XXXXXXXXXX010101111111000000011110101010101010000".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue

When the Ethernet trigger condition is set to MACLENgth, and the qualifier is set to INrange or OUTrange, this command specifies the upper data value of the range. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue to specify the lower limit of the range.) The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue

Arguments <QString> is a quoted string of up toe 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples `TRIGger:A:BUS:B1:ETHERnet:MAC:LENgth:HIVALue "XXXXXXXX00001000"` sets the upper limit of the range to the hexadecimal value XX08 (when the trigger condition is set to `MACLENgth`, and the qualifier is set to `INrange` or `OUTrange)`.

`TRIGGER:A:BUS:B1:ETHERNET:MAC:LENGTH:HIVALUE?` might return `"XXXXXXXX00001000"`.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue

When the Ethernet trigger condition is set to `MACLENgth`, and the qualifier is set to `LESSthan, MOREthan, EQual, UNEQual, LESSEQual` or `MOREEQual`, this command specifies the 16–bit value to trigger on. When the qualifier is set to `INrange` or `OUTrange`, this command specifies the lower limit of the range. (Use the command TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue to set the upper limit of the range.) The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:LENgth:HIVALue

Arguments <QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples TRIGger:A:BUS:B1:ETHERnet:MAC:LENgth:VALue "XXXXXXXX00001000" sets the MAC length/type value to trigger on the hexadecimal value XX08, assuming the qualifier is set to LESSthan, MOREthan, EQual, UNEQual, LESSEQual or MOREEQual.

TRIGger:A:BUS:B1:ETHERnet:MAC:LENgth:VALue "XXXXXXXX00001000" sets the lower limit of the range to the hexadecimal value XX08, assuming the qualifier is set to INrange or OUTrange.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:VALue

When the Ethernet trigger condition is set to QTAG, this command specifies the 32-bit Q-Tag value to trigger on. The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

Arguments <QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples TRIGGER:A:BUS:B1:ETHERNET:QTAG:VALUE "XXXXXXXXXXXXXXXXXXXX010010001010" would specify to trigger on the Q-Tag value of hexadecimal XXXXX48A.

TRIGGER:A:BUS:B1:ETHERNET:QTAG:VALUE? might return "XXXXXXXXXXXXXXXXXXXX010010001010".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue

When the Ethernet trigger condition is set to TCPHeader, this command specifies the 32-bit acknowledgement number that is to be used in the trigger (along with the destination and source port addresses and the sequence number). The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue

Arguments <QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:ACKnum:VALue "XXXXXXXXXXXXXXXXXXXX00001000" sets the acknowledgement number to be used in the trigger to hexadecimal XXXXXX08.

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:ACKnum:VALue? might return :TRIGGER:A:BUS:B1:ETHERNET:TCPHEADER:ACKNUM:VALUE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" indicating the value is set to XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue

When the Ethernet trigger condition is set TCPHeader, this command specifies the 16–bit destination port address value that is to be used in the trigger (along with the acknowledgement value, source port address and the sequence number). The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:
VALue <QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:
VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue

Arguments <QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples TRIGger:A:BUS:B1:ETHERnet:TCPHeader:DESTinationport:VALue "XXXXXXXX00100010" would set the destination port address value that is to be used in the trigger to hexadecimal XX22.

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:DESTinationport:VALue? might return "XXXXXXXXXXXXXXXX".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue

When the Ethernet trigger condition is set to TCPHeader, this command specifies the 32–bit sequence number that is to be used in the trigger (along with the destination and source port addresses and the acknowledgement value). The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue

Arguments <QString> is a quoted string of up to 32 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples TRIGger:A:BUS:B1:ETHERnet:TCPHeader:SEQnum:VALue "XXXXXXXXXXXXXXXXXXXX000100010001" would set the sequence number that is to be used in the trigger to hexadecimal XXXXX111.

TRIGger:A:BUS:B1:ETHERnet:TCPHeader:SEQnum:VALue? might return "XXXXXXXXXXXXXXXXXXXXX0010010100".

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue

When the Ethernet trigger condition is set to TCPHeader, this command specifies the 16–bit source port address that is to be used in the trigger (along with the destination port address, the sequence number and the acknowledgement number). The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRENET or SUP5-SRENET Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SOUrceport:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:ETHERnet:CONDition

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:ACKnum:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:SEQnum:VALue

TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:DESTinationport:VALue

Arguments

<QString> is a quoted string of up to 16 characters where the allowable characters are 0, 1, and X. The bits specified in the quoted string replace the least significant bits, leaving any unspecified upper bits unchanged.

Examples

```
TRIGger:A:BUS:B1:ETHERnet:TCPHeader:SOUrceport:VALue
"XXXX000010100110"
```
would set the source port address that is to be used in the trigger to hexadecimal X0A6.

```
TRIGger:A:BUS:B1:ETHERnet:TCPHeader:SOUrceport:VALue?
```
might return "XXXXX01001010110".

TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition

This command specifies the condition to use when triggering on a FlexRay bus signal (start of frame, frame type, ID, cycle count, header, data, ID and data, EOF, error). The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition
{SOF|FRAMEType|IDentifier|CYCLEcount|HEADer|DATa
|IDANDDATA|EOF|ERRor}
```

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition?
```

Arguments

SOF sets the trigger condition to start of frame.

FRAMEType sets the trigger condition to frame type.

IDentifier sets the trigger condition to identifier.

CYCLEcount sets the trigger condition to cycle count.

HEADer sets the trigger condition to header.

DATa sets the trigger condition to data.

IDANDDATA sets the trigger condition to id and data.

EOF sets the trigger condition to end of frame.

ERRor sets the trigger condition to error.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:CONDITION FRAMETYPE sets the FlexRay condition to frame type.

TRIGGER:A:BUS:B1:FLEXRAY:CONDITION? might return TRIGGER:A:BUS:B1:FLEXRAY:CONDITION SOF indicating the FlexRay condition is start of frame.

TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue

This command specifies the high value when triggering on a FlexRay bus cycle count field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue to set the low value.) The trigger condition must be set to CYCLEcount (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue?
```

Arguments

<QString> is a quoted string that is the cycle count high value.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE "110010" sets the cycle count high value to 110010.

TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE? might return TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE "XXXXXX" indicating the cycle count high value is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:QUALifier

This command specifies the qualifier (<, >, =, <=, >=, not =, in range, out of range) to use when triggering on the FlexRay bus cycle count field. The trigger condition must be set to CYCLEcount (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition).The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:QUALifier
{LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual|
INrange|OUTrange}
```

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:QUALifier?
```

Arguments

`LESSthan` sets the cycle count qualifier to less than.

`MOREthan` sets the cycle count qualifier to more than.

`EQual` sets the cycle count qualifier to equal.

`UNEQual` sets the cycle count qualifier to not equal.

`LESSEQual` sets the cycle count qualifier to less than or equal.

`MOREEQual` sets the cycle count qualifier to greater than or equal.

`INrange` sets the cycle count qualifier to in range.

`OUTrange` sets the cycle count qualifier to out of range.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER LESSTHAN` sets the cycle count qualifier to LESSTHAN.

`TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER?` might return `TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER EQUAL` indicating that the cycle count qualifier is set to EQUAL.

TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue

This command specifies the low value when triggering on the FlexRay bus cycle count field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:HIVALue to set the upper value.) The trigger condition must be set to CYCLEcount (using

TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition).The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:VALue?
```

Arguments <QString> is a quoted binary data string that represents the cycle count low value.

Examples TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE "11001101" sets the cycle count value to 11001101.

TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE? might return TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE "XXXXXX" indicating the cycle count value is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue

This command specifies the high value when triggering on the FlexRay bus data field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:VALue to set the lower value.) The trigger condition needs to be set to ID or IDANDDATA (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition).The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue?
```

Arguments <QString> is a quoted string that is the binary data high value.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:DATA:HIVALUE "11001101XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX1"` sets the binary data string high value to `"11001101XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX1"`.

`TRIGGER:A:BUS:B1:FLEXRAY:DATA:HIVALUE?` might return `TRIGGER:A:BUS:B1:FLEXRAY:DATA:HIVALUE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"` indicating the binary data string high value is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:OFFSet

This command specifies the offset of the data string, in bytes, when triggering on the FlexRay bus data field. The trigger condition needs to be set to ID or IDANDDATA (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:OFFSet?
```

Arguments

`<NR1>` is the offset of the data string in bytes. A byte offset of -1 signifies “don't care”, and no byte offset is used. The instrument will trigger on or match any byte value that fits.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET 1` sets the offset to 1.

`TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET?` might return `TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET 0` indicating that a data offset of 0.

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:QUALifier

This command specifies the qualifier (<, >, =, <=, >=, not =, in range, out of range) to use when triggering on the FlexRay bus data field. The trigger condition needs to be set to ID or IDANDDATA (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:QUALifier
{LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual|
INrange|OUTrange}
```

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:QUALifier?
```

Arguments

`LESSthan` sets the data qualifier to less than.

`MOREthan` sets the data qualifier to greater than.

`EQual` sets the data qualifier to eqaual.

`UNEQual` sets the data qualifier to not equal.

`LESSEQual` sets the data qualifier to less than or equal.

`MOREEQual` sets the data qualifier to greater than or equal.

`INrange` sets the data qualifier to in range.

`OUTrange` sets the data qualifier to out of range.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER LESSTHAN` sets the data qualifier to LESSTHAN.

`TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER?` might return `TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER EQUAL` indicating the data qualifier is EQUAL.

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:SIZe

This command specifies the length of the data string, in bytes, when triggering on the FlexRay bus data field. The trigger condition needs to be set to ID or IDANDDATA (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:SIZe?
```

Arguments

<NR1> is the FlexRay data string length, in bytes.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE 8 sets the data string size to 8 bytes.

TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE? might return TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE 1 indicating the data size is 1 byte.

TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:VALue

This command specifies the low value when triggering on the FlexRay bus data field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:HIVALue to set the upper value.) The trigger condition needs to be set to ID or IDANDDATA (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:VALue?
```

Arguments

<QString> is a quoted string.

Examples

TRIGGER:A:BUS:B1:FLEXRAY:DATA:VALUE "11001101" sets the FlexRay data value for triggering to 11001101.

TRIGGER:A:BUS:B1:FLEXRAY:DATA:VALUE? might return TRIGGER:A:BUS:B1:FLEXRAY:DATA:VALUE "XXXXXXXX" indicating the FlexRay data value is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:EOFTYPE

This command specifies the end of file type (static, dynamic or any) when triggering on the FlexRay bus EOF field. The trigger condition needs to be set to

EOF (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:EOFTYPE {STATic|DYNAMic|ANY}
TRIGger:{A|B}:BUS:B<x>:FLEXray:EOFTYPE?
```

Arguments

`STATic` specifies triggering on the STATIC end of file type.

`DYNAMic` specifies triggering on the DYNAMIC end of file type.

`ANY` specifies triggering on a STATIC or DYNAMIC end of file type.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE ANY` sets the FlexRay end of file type to ANY.

`TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE?` might return `TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE STATIC` indicating the FlexRay end of file type is STATIC

TRIGger:{A|B}:BUS:B<x>:FLEXray:ERRTYPE

This command specifies the error type when triggering on the FlexRay bus signal. The trigger condition needs to be set to ERROR (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:ERRTYPE
{CRCHeader|CRCTrailer|SYNCFrame|STARTupnosync|NULLFRStatic|
NULLFRDynamic}
TRIGger:{A|B}:BUS:B<x>:FLEXray:ERRTYPE?
```

Arguments

`CRCHeader` sets the error type to CRCHeader.

`CRCTrailer` sets the error type to CRCTrailer.

`SYNCFrame` sets the error type to SYNCFrame.

`STARTupnosync` sets the error type to STARTupnosync.

`NULLFRStatic` sets the error type to NULLFRStatic.

`NULLFRDynamic` sets the error type to NULLFRDynamic.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE SYNCFRAME` sets the trigger type is SYNCFRAME.

`TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE?` might return `TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE CRCHEADER` indicating the FlexRay trigger type is CRCHeader.

TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue

This command specifies the high value when triggering on the FlexRay bus frame ID field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALue to set the low value.) The trigger condition needs to be set to IDentifier (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue?
```

Arguments

`<QString>` is a quoted string that is the binary frame ID high value.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE "11001100101"` sets the frame ID high value to 11001100101.

`TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE?` might return `TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE "XXXXXXXXXXX"` indicating the frame ID high value is "don't care".

TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:QUALifier

This command specifies the qualifier to use when triggering on the FlexRay bus frame ID field. The trigger condition needs to be set to IDentifier (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:QUALifier {LESSthan|
MOREthan|EQual|UNEQual|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:QUALifier?
```

Arguments

`LESSthan` sets the frame ID qualifier to less than.

`MOREthan` sets the frame ID qualifier to greater than.

`EQual` sets the frame ID qualifier to equal.

`UNEQual` sets the frame ID qualifier to not equal.

`LESSEQual` sets the frame ID qualifier to less than or equal.

`MOREEQual` sets the frame ID qualifier to greater than or equal.

`INrange` sets the frame ID qualifier to in range.

`OUTrange` sets the frame ID qualifier to out of range.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER LESSTHAN` sets the frame ID qualifier to less than.

`TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER?` might return `TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER EQUAL` indicating the frame ID qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALue

This command specifies the low value when triggering on the FlexRay bus frame ID field. (Use TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:HIVALue to set the high value.) The trigger condition needs to be set to IDentifier (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:VALue?
```

Arguments <QString> is a quoted string that is the FlexRay frame ID low value.

Examples `TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE "11001100101"` sets the frame ID value to 11001100101.

`TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE?` might return `TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE "XXXXXXXXXXX"` indicating the frame ID value is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEType

This command specifies the frame type (normal, payload, null, sync or startup) when triggering on the FlexRay bus signal. The trigger condition needs to be set to FRAMEType (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEType
{NORMal|PAYLoad|NULL|SYNC|STARTup}

TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEType?
```

Arguments `NORMal` specifies the normal frame type.

`PAYLoad` specifies the payload frame type.

`NULL` specifies the null frame type.

`SYNC` specifies the sync frame type.

`STARTup` specifies the startup frame type.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE PAYLOAD` sets the frame type to payload.

`TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE?` might return `TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE NORMAL` indicating the frame type is set to normal.

TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CRC

This command specifies the CRC portion of the binary header string when triggering on the FlexRay bus signal. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CRC <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CRC?
```

Arguments

`<QString>` is a quoted string that is the CRC portion of the binary header string.

Examples

`TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC "11001100101"` sets the CRC portion of the binary header string to 11001100101.

`TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC?` might return `TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC "XXXXXXXXXXX"` indicating the CRC portion of the binary header string is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CYCLEcount

This command specifies the cycle count portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CYCLEcount <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:CYCLEcount?
```

Arguments <QString> is a quoted string that is the cycle count portion of the binary header string.

Examples TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT "110010" sets the cycle count to 110010.

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT? might return TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT "XXXXXX" indicating the cycle count is don't care.

TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:FRAMEID

This command specifies the frame ID portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:FRAMEID <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:FRAMEID?
```

Arguments <QString> is a quoted string that represents the frame ID portion of the binary header string.

Examples TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID "11001100101" sets the frame ID portion of the binary header string to 11001100101.

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID? might return TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID "XXXXXXXXXXX" indicating the frame ID portion of the binary header string is "don't care".

TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:INDBits

This command specifies the indicator bits portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition). The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:INDBits <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:INDBits?
```

Arguments <QString> is a quoted string that is the indicator bits portion of the binary header string.

Examples `TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS "11001"` sets the indicator bits portion of the header string to 11001.

`TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS?` might return `TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS "XXXXX"` indicating that the indicator bits portion of the header string are "don't cares".

TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:PAYLength

This command specifies the payload length portion of the binary header string when triggering on the FlexRay bus header. The trigger condition needs to be set to HEADer (using TRIGger:{A|B}:BUS:B<x>:FLEXray:CONDition).The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:PAYLength <QString>
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:PAYLength?
```

Arguments <QString> is the length of the payload portion of the Binary header string.

Examples TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH "1100101" sets the FlexRay header paylength to 1100101.

TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH? might return TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH "XXXXXXX" indicating the FlexRay header paylength is don't care.

TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:MODe

This command specifies the I²C address mode to 7 or 10-bit. The bus number is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:MODe {ADDR7|ADDR10}
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:MODe?
```

Arguments ADDR7 specifies the 7-bit I²C address mode.

ADDR10 specifies the 10-bit I²C address mode.

Examples TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE ADDR10 sets the I2C address mode to 10-bit.

TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE? might return :TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE ADDR7 indicating the address mode is set to the 7-bit mode.

TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:VALue

This command specifies the binary address string used for the I²C trigger if the trigger condition is ADDRESS or ADDRANDDATA. The bus number is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:VALue?
```

Arguments <QString> is up to 7 or 10-bits depending on the address mode that specifies the address. The only allowed characters in the QString are 0, 1, and X.

Examples TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE "1011" sets the I²C address value to XXX1011.

TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE? might return :TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE "XXXXXXX" indicating the address value is set to XXXXXXX.

TRIGger:{A|B}:BUS:B<x>:I2C:CONDition

This command specifies the trigger condition for an I²C trigger. The bus number is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:I2C:CONDition
{STARt|STOP|REPEATstart|ACKMISS|ADDRess|DATa|ADDRANDDATA}

TRIGger:{A|B}:BUS:B<x>:I2C:CONDition?
```

Arguments

STARt specifies a search based on start condition.

STOP specifies a search based on stop condition.

REPEATstart specifies a search based on repeat of start condition.

ACKMISS specifies a search based on missing acknowledgement condition.

ADDRess specifies a search based on address.

DATa specifies a search based on data.

ADDRANDDATA specifies a search based on address and data.

Examples `TRIGGER:A:BUS:B1:I2C:CONDITION START` specifies start as the I²C trigger condition.

`TRIGGER:A:BUS:B1:I2C:CONDITION?` might return `:TRIGGER:A:BUS:B1:I2C:CONDITION START` indicating the condition is set to the start condition.

TRIGger:{A|B}:BUS:B<x>:I2C:DATa:DIRection

This command specifies the I²C trigger type to be valid on a Read, Write, or Either condition. Read or write is indicated by the R/W bit in the I²C protocol. The bus number is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:DIRection
{READ|WRITE|NOCARE}
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:DIRection?
```

Arguments `READ` specifies read as the data direction.

`WRITE` specifies write as the data direction.

`NOCARE` specifies either as the data direction.

Examples `TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION WRITE` specifies write as the I²C data direction.

`TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION?` might return `:TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION NOCARE` indicating the data direction is either read or write.

TRIGger:{A|B}:BUS:B<x>:I2C:DATa:SIZe

This command specifies the length of the data string in bytes to be used for an I2C trigger if the trigger condition is DATA or ADDRANDDATA. Applies to bus <x>, where the bus number is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:SIZe?
```

Arguments <NR1> is the length of the data string in bytes.

Examples TRIGger:A:BUS:B1:I2C:DATa:SIZe 1 sets the data size to 1 byte.

TRIGger:A:BUS:B1:I2C:DATa:SIZe? might return :TRIGGER:A:BUS:B1:I2C:DATA:SIZE 1 indicating the size is set to 1 byte.

TRIGger:{A|B}:BUS:B<x>:I2C:DATa:VALue

This command specifies the binary data string used for I2C triggering if the trigger condition is DATA or ADDRANDDATA. The bus number is specified by x.

Conditions Requires option 5-SREMBD or SUP5-SREMBD Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:VALue?
```

Arguments <QString> is the binary data string, where the number of bits is 8 times the number of bytes specified. The only allowed characters in the string are 0, 1, and X.

Examples TRIGger:A:BUS:B1:I2C:DATa:VALue "11001101" sets the data value to 1100101.

TRIGger:A:BUS:B1:I2C:DATa:VALue? might return :TRIGGER:A:BUS:B1:I2C:DATA:VALUE "XXXXXXXX" indicating the data value is XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:LIN:CONDition

This command specifies the trigger condition for LIN. The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:LIN:CONDition
{SYNCfield|IDentifier|DATa|IDANDDATA|WAKEup|SLEEP|ERRor}

TRIGger:{A|B}:BUS:B<x>:LIN:CONDition?
```

Arguments

`SYNCfield` sets the LIN trigger condition to sync field.

`IDentifier` sets the LIN trigger condition to identifier.

`DATa` sets the LIN trigger condition to data.

`IDANDDATA` sets the LIN trigger condition to id and data.

`WAKEup` sets the LIN trigger condition to wake up.

`SLEEP` sets the LIN trigger condition to sleep.

`ERRor` sets the LIN trigger condition to error.

Examples

`TRIGGER:A:BUS:B1:LIN:CONDITION ERROR` sets the LIN trigger condition to error.

`TRIGGER:A:BUS:B1:LIN:CONDITION?` might return `TRIGGER:A:BUS:B1:LIN:CONDITION SYNCFIELD` indicating the LIN trigger condition is sync field.

TRIGger:{A|B}:BUS:B<x>:LIN:DATa:HIVALue

This command specifies the high data value string used for a LIN bus trigger when the trigger condition is DATA or IDANDDATA and the data qualifier is INRANGE or OUTRANGE. The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:HIVALue?
```

Arguments <QString> is a quoted string that is the binary data string used for LIN trigger if the trigger condition is ID or IDANDDATA.

Examples `TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE "11001010"` sets the high value to 11001010.

`TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE?` might return `TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE "XXXXXXXX"`indicating the high value is don't care.

TRIGger:{A|B}:BUS:B<x>:LIN:DATa:QUALifier

This command specifies the LIN data qualifier. This only applies if the trigger condition is IDANDDATA or DATA. The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:QUALifier
{LESSthan|MOREthan|EQual|UNEQual|LESSEQual|MOREEQual|
INrange|OUTrange}

TRIGger:{A|B}:BUS:B<x>:LIN:DATa:QUALifier?
```

Arguments `LESSthan` sets the LIN data qualifier to less than.

`MOREthan` sets the LIN data qualifier to greater than.

`EQual` sets the LIN data qualifier to equal.

`UNEQual` sets the LIN data qualifier to not equal.

`LESSEQual` sets the LIN data qualifier to less than or equal.

`MOREEQual` sets the LIN data qualifier to greater than or equal.

`INrange` sets the LIN data qualifier to in range.

OUTrange sets the LIN data qualifier to out of range.

Examples TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER OUTRANGE sets the data qualifier to out of range.

TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER? might return TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER EQUAL indicating the data qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe

This command specifies the length of the data string in bytes to be used for LIN trigger. The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe?
```

Arguments <NR1> is the size of the data string in bytes.

Examples TRIGGER:A:BUS:B1:LIN:DATA:SIZE 8 sets the data size to 8 bytes.

TRIGGER:A:BUS:B1:LIN:DATA:SIZE? might return TRIGGER:A:BUS:B1:LIN:DATA:SIZE 1 indicating the data size is 1 byte.

TRIGger:{A|B}:BUS:B<x>:LIN:DATa:VALue

This command specifies the binary data string to be used for LIN trigger condition if trigger condition is ID or IDANDDATA. The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:VALue?
```

Arguments

<QString> is a quoted string that is the LIN trigger data value.

Examples

TRIGGER:A:BUS:B1:LIN:DATA:VALUE "11001101"sets the data value to 11001101.

TRIGGER:A:BUS:B1:LIN:DATA:VALUE? might return TRIGGER:A:BUS:B1:LIN:DATA:VALUE "XXXXXXXX" indicating the data value is don't care.

TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE

This command specifies the error type be used for LIN trigger. The bus number is specified by x.

Conditions

Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE {SYNC|PARity|CHecksum}
TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE?
```

Arguments

SYNC sets the LIN error type to SYNC.

PARity sets the LIN error type to parity.

CHecksum sets the LIN error type to checksum.

Examples

TRIGGER:A:BUS:B1:LIN:ERRTYPE CHECKSUM sets the LIN error type to checksum.

TRIGGER:A:BUS:B1:LIN:ERRTYPE? might return TRIGGER:A:BUS:B1:LIN:ERRTYPE SYNC indicating the LIN error type is SYNC.

TRIGger:{A|B}:BUS:B<x>:LIN:IDentifier:VALue

This command specifies the binary address string used for LIN bus trigger if the trigger condition is ID or IDANDDATA. The bus number is specified by x.

Conditions Requires option 5-SRAUTO or SUP5-SRAUTO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:LIN:IDentifier:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:LIN:IDentifier:VALue?
```

Arguments <QString> is the binary address string used for LIN trigger if the trigger condition is ID or IDANDDATA.

Examples `TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE "110010"` sets the identifier value to 110010.

`TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE?` might return `TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE "XXXXXX"` indicating the identifier value is XXXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:HIVALue

This command sets or queries the high value when triggering on command word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to COMMAND, and the address qualifier must be INrange or OUTrange.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:HIVALue
<QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:HIVALue?
```

Arguments <QString> is the address value.

Examples `TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:HIVALUE "X1000"` sets the value to X1000.

`TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:HIVALUE?` might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:QUALifier

This command sets or queries the qualifier to be used when triggering on command word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:
QUALifier {EQual|UNEQual|LESSthan|MOREthan
|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:QUALifier?
```

Arguments Arguments are the available address qualifiers.

Examples `TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:QUALIFIER LESSTHAN` sets the address qualifier to less than.

`TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:QUALIFIER?` might return EQUAL, indicating that the address qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:VALue

This command sets or queries the low value when triggering on command word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to COMMAND

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:ADDRess:VALue?
```

Arguments `<QString>` is the address value.

Examples `TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:VALUE "X1000"` sets the value to X1000.

`TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:VALUE?` might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:COUNt

This command sets or queries the value of the command word "word count" field for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:COUNt <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:COUNt?
```

Arguments `<QString>` is the word count value.

Examples `TRIGGER:A:BUS:B1:MIL1553B:COMMAND:COUNT "X1000"` sets the value to X1000.

`TRIGGER:A:BUS:B1:MIL1553B:COMMAND:COUNT?` might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:PARity
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:PARity?
```

Arguments

ONE filters command words to only match those where the parity bit has a value of 1.

ZERo filters command words to only match those where the parity bit has a value of 0.

NOCARE disables filtering of command words on the parity bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:PARITY ONE specifies filtering command words for those where the parity bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:PARITY? might return NOCARE, indicating that command words are not being filtered based on the parity bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:SUBADdress

This command sets or queries the value of the command word subaddress field for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:SUBADdress <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:SUBADdress?
```

Arguments

<QString> is the subaddress value.

Examples

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:SUBADDRESS "X1000" sets the value to X1000.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:SUBADDRESS? might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:TRBit

This command sets or queries the value of the command word Transmit / Receive bit for a MIL-STD-1553 bus to trigger on. The bus number is specified by x. The trigger condition must be set to COMMAND.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:TRBit {RX|TX|X}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:COMMAND:TRBit?
```

Arguments

RX filters command words to only match those that are receive packets.

TX filters command words to only match those that are transmit packets.

X disables filtering of command words on the R/T bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:TRBIT TX specifies filtering command words for only transmit messages.

TRIGGER:A:BUS:B1:MIL1553B:COMMAND:TRBIT? might return X, indicating that command words are not being filtered based on the R/T bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:CONDition

This command sets or queries the field or condition for a MIL-STD-1553 bus to trigger on. The bus number is specified by x.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:CONDition
{SYNC|COMMAND|STATus|DATA |TIMe|ERRor}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:CONDition?
```

Arguments

SYNC specifies triggering on the sync pulse of any word.

COMMAND specifies triggering on a matching command word.

STATus specifies triggering on a matching status word.

DATA specifies triggering on a matching data word.

TIMe specifies triggering on the response time or intermessage gap between words.

ERRor specifies triggering on a specified error condition.

Examples

TRIGGER:A:BUS:B1:MIL1553B:CONDITION DATA specifies finding matching data word(s).

TRIGGER:A:BUS:B1:MIL1553B:CONDITION? might return SYNC, indicating that the bus is triggering on sync pulses found in any word.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:PARity

This command sets or queries the value of the command word parity bit for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to DATA.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:PARity
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:PARity?
```

Arguments

ONE filters data words to only match those where the parity bit has a value of 1.

ZERo filters data words to only match those where the parity bit has a value of 0.

NOCARE disables filtering of data words on the parity bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:DATA:PARITY ONE specifies filtering data words for those where the parity bit has the value 1.

TRIGGER:A:BUS:B1:MIL1553B:DATA:PARITY? might return NOCARE, indicating that data words are not being filtered based on the parity bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:VALue

This command sets or queries the value when triggering on data words for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to DATA.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:VALue?
```

Arguments <QString> is the data value.

Examples TRIGGER:A:BUS:B1:MIL1553B:DATA:VALUE "XXXXXXXXXXXX1000" sets the value to XXXXXXXXXXXX1000.

TRIGGER:A:BUS:B1:MIL1553B:DATA:VALUE? might return "XXXXXXXXXXXXXXXX", indicating that the value is XXXXXXXXXXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:ERRTYPe

This command sets or queries the type of error condition for a MIL-STD-1553 bus to trigger on. The bus number is specified by x. The trigger condition must be set to ERRor.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:ERRTYPe {PARity|SYNC|DATA}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:ERRTYPe?
```

Arguments PARity specifies triggering on an incorrectly calculated parity bit in any word.

SYNC specifies triggering on any sync pulse that does not transition in the middle of the pulse as required.

DATA specifies triggering on any non-contiguous data words.

Examples

TRIGGER:A:BUS:B1:MIL1553B:ERRTYPE DATA specifies triggering on non-contiguous data words.

TRIGGER:A:BUS:B1:MIL1553B:ERRTYPE? might return PARITY, indicating that the bus is triggering on parity errors in any word.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:HIVALue

This command sets or queries the high value when triggering on status word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to STATus and the address qualifier must be INrange or OUTrange.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:HIVALue
<QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:HIVALue?
```

Arguments

<QString> is the address value.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:HIVALUE "X1000" sets the value to X1000.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:HIVALUE? might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:QUALifier

This command sets or queries the qualifier to be used when triggering on status word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:
ADDRess:QUALifier {EQual|UNEQual|LESSthan
|MOREthan|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:QUALifier?
```

Arguments Arguments are the available address qualifiers.

Examples `TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:QUALIFIER LESSTHAN` sets the address qualifier to less than.

`TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:QUALIFIER?` might return EQUAL, indicating that the address qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:VALue

This command sets or queries the low value when triggering on status word addresses for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:VALue
<QString>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:ADDRess:VALue?
```

Arguments `<QString>` is the address value.

Examples `TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:VALUE "X1000"` sets the value to X1000.

`TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:VALUE?` might return "XXXXX", indicating that the value is XXXXX.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BCR

This command sets or queries the value of the broadcast command received bit (BCR bit, bit 15) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BCR
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BCR?
```

Arguments `ONE` filters status words to only match those where the BCR bit has a value of 1.

`ZERo` filters status words to only match those where the BCR bit has a value of 0.

`NOCARE` disables filtering of status words on the BCR bit.

Examples `TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BCR ONE` specifies filtering status words for those where the BCR bit has a value of 1.

`TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BCR?` might return NOCARE, indicating that status words are not being filtered based on the BCR bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BUSY

This command sets or queries the value of the busy bit (BUSY bit, bit 16) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BUSY
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:BUSY?
```

Arguments

ONE filters status words to only match those where the BUSY bit has a value of 1.

ZERo filters status words to only match those where the BUSY bit has a value of 0.

NOCARE disables filtering of status words on the BUSY bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BUSY ONE specifies filtering status words for those where the BUSY bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BUSY? might return NOCARE, indicating that status words are not being filtered based on the BUSY bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:DBCA

This command sets or queries the value of the dynamic bus control acceptance bit (DBCA bit, bit 18) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:DBCA
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:DBCA?
```

Arguments

ONE filters status words to only match those where the DBCA bit has a value of 1.

ZERo filters status words to only match those where the DBCA bit has a value of 0.

NOCARE disables filtering of status words on the DBCA bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:DBCA ONE specifies filtering status words for those where the DBCA bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:DBCA? might return NOCARE, indicating that status words are not being filtered based on the DBCA bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:INSTR

This command sets or queries the value of the instrumentation bit (INSTR bit, bit 10) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:INSTR
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:INSTR?
```

Arguments ONE filters status words to only match those where the INSTR bit has a value of 1.

ZERo filters status words to only match those where the INSTR bit has a value of 0.

NOCARE disables filtering of status words on the INSTR bit.

Examples TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:INSTR ONE specifies filtering status words for those where the INSTR bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:INSTR? might return NOCARE, indicating that status words are not being filtered based on the INSTR bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME

This command sets or queries the value of the message error bit (ME bit, bit 9) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:ME?
```

Arguments ONE filters status words to only match those where the ME bit has a value of 1.

ZERo filters status words to only match those where the ME bit has a value of 0.

NOCARE disables filtering of status words on the ME bit.

Examples

`TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:ME ONE` specifies filtering status words for those where the ME bit has a value of 1.

`TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:ME?` might return NOCARE, indicating that status words are not being filtered based on the ME bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SRQ

This command sets or queries the value of the status word service request bit (SRQ bit, bit 11) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SRQ
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SRQ?
```

Arguments

`ONE` filters status words to only match those where the SRQ bit has a value of 1.

`ZERo` filters status words to only match those where the SRQ bit has a value of 0.

`NOCARE` disables filtering of status words on the SRQ bit.

Examples

`TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SRQ ONE` specifies filtering status words for those where the SRQ bit has a value of 1.

`TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SRQ?` might return NOCARE, indicating that status words are not being filtered based on the SRQ bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SUBSF

This command sets or queries the value of the subsystem flag bit (SUBSF bit, bit 17) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SUBSF
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:SUBSF?
```

Arguments ONE filters status words to only match those where the SUBSF bit has a value of 1.

ZERo filters status words to only match those where the SUBSF bit has a value of 0.

NOCARE disables filtering of status words on the SUBSF bit.

Examples TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SUBSF ONE specifies filtering status words for those where the SUBSF bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SUBSF? might return NOCARE, indicating that status words are not being filtered based on the SUBSF bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:TF

This command sets or queries the value of the terminal flag bit (TF bit, bit 19) in a status word for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:TF
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:TF?
```

Arguments ONE filters status words to only match those where the TF bit has a value of 1.

ZERo filters status words to only match those where the TF bit has a value of 0.

NOCARE disables filtering of status words on the TF bit.

Examples TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:TF ONE specifies filtering status words for those where the TF bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:TF? might return NOCARE, indicating that status words are not being filtered based on the TF bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:PARity

This command sets or queries the value of the status word parity bit for a MIL-STD-1553 bus to triggering on. The bus number is specified by x. The trigger condition must be set to STATus.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:PARity
{ONE|ZERo|NOCARE}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:PARity?
```

Arguments

ONE filters status words to only match those where the parity bit has a value of 1.

ZERo filters status words to only match those where the parity bit has a value of 0.

NOCARE disables filtering of status words on the parity bit.

Examples

TRIGGER:A:BUS:B1:MIL1553B:STATUS:PARITY ONE specifies filtering status words for those where the parity bit has a value of 1.

TRIGGER:A:BUS:B1:MIL1553B:STATUS:PARITY? might return NOCARE, indicating that status words are not being filtered based on the parity bit value.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:LESSLimit

This command sets or queries the lower limit to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to TIMe.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:LESSLimit <NR3>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:LESSLimit?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier

Arguments

<NR3> is a floating point number that specifies the lower bound for measuring remote terminal response time (RT) or the inter-message gap (IMG) in seconds.

Examples

TRIGGER:A:BUS:B1:MIL1553B:TIME:LESSLIMIT 2.0000e-6 sets the lower bound for comparison to 2 microseconds.

TRIGGER:A:BUS:B1:MIL1553B:TIME:LESSLIMIT? might return 4.0000E-6, indicating that the lower bound for comparison is set to 4 microseconds.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:MORELimit

This command sets or queries the upper limit to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to TIMe.

Conditions

Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:MORELimit <NR3>
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:MORELimit?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier

Arguments

<NR3> is a floating point number that specifies the upper bound for measuring remote terminal response time (RT) or the inter-message gap (IMG) in seconds.

Examples

TRIGGER:A:BUS:B1:MIL1553B:TIME:MORELIMIT 15.0000e-6 sets the upper bound for comparison to 15 microseconds.

TRIGGER:A:BUS:B1:MIL1553B:TIME:MORELIMIT? might return 12.0000E-6, indicating that the upper bound for comparison is set to 12 microseconds.

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier

This command sets or queries the qualifier to be used when triggering on response time / inter message gap time for a MIL-STD-1553 bus. The bus number is specified by x. The trigger condition must be set to TIMe.

Conditions Requires option 5-SRAERO or SUP5-SRAERO Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier
{LESSthan|MOREthan|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:QUALifier?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:LESSLimit

TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:MORELimit

Arguments Arguments are the available comparison qualifiers.

Examples `TRIGGER:A:BUS:B1:MIL1553B:TIME:QUALIFIER LESSTHAN` sets the time comparison qualifier to less than.

`TRIGGER:A:BUS:B1:MIL1553B:TIME:QUALIFIER?` might return OUTRANGE, indicating that the qualifier is set to out-of-range comparison.

TRIGger:{A|B}:BUS:B<x>:PARallel:DATa:VALue

This command specifies the binary data string used for a Parallel Bus trigger. The bus number is specified by x.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:PARallel:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:PARallel:DATa:VALue?
```

Arguments `<QString>` is the binary data string used for a Parallel Bus trigger.

Examples

`TRIGger:A:BUS:B1:PARallel:DATa:VALue "11001101"` sets the data value to 11001101.

`TRIGger:A:BUS:B1:PARallel:DATa:VALue?` might return `:TRIGGER:A:BUS:B1:PARALLEL:DATA:VALUE "X"` indicating the data value is X.

TRIGger:{A|B}:BUS:B<x>:RS232C:CONDition

This command specifies the condition for an RS-232C trigger, where the bus number is specified by< >x.

Conditions

Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:RS232C:CONDition
{STARt|EOp|DATa|PARItyerror}
TRIGger:{A|B}:BUS:B<x>:RS232C:CONDition?
```

Arguments

`STARt` sets the Trigger on condition to Start.

`EOp` sets the Trigger on condition to End of Packet.

`DATa` sets the Trigger on condition to Data.

`PARItyerror` sets the Trigger on condition to Parity Error.

Examples

`TRIGger:A:BUS:B1:RS232C:CONDition DATA` sets the trigger on condition to data.

`TRIGger:A:BUS:B1:RS232C:CONDition?` might return `:TRIGGER:A:BUS:B1:RS232C:CONDITION START` indicating start is the trigger condition.

TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:SIZe

This command sets or queries the length of the data string in bytes to be used for an RS-232C trigger when the trigger condition is Data. The bus number is specified by x.

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:SIZe <NR3>
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:SIZe?
```

Arguments <NR3> specifies the data size in bytes.

Examples `TRIGGER:A:BUS:B1:RS232C:DATA:SIZE 3` sets three bytes data size for the RS-232C bus B1 trigger.

`TRIGGER:A:BUS:B1:RS232C:DATA:SIZE?` might return `:TRIGGER:A:BUS:B1:RS232C:DATA:SIZE 2`, indicating that the data size for the RS-232C bus B1 trigger is set to two bytes.

TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:VALue

This command sets or queries the data address string used for the RS-232 bus trigger when the trigger condition is set to Data. The bus number is specified by x.

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:VALue?
```

Arguments <QString> specifies the address value. The argument is a string of 0, 1, or X representing a binary number.

Examples `TRIGGER:A:BUS:B1:RS232C:DATA:VALUE "011XX11"` sets the data address string used for the RS-232 bus trigger to 011XX11.

`TRIGGER:A:BUS:B1:RS232C:DATA:VALUE?` might return `:TRIGGER:A:BUS:RS232C:DATA:VALUE "XXXXXX01"`, indicating that the data address string used for the RS-232 bus trigger is set to "XXXXXX01"

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

This command sets or queries the trigger condition for a SENT bus.

Conditions Requires option SRAUTOSEN.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:CONDition
{START|FAST|SLOW|ERRor}
TRIGger:{A|B}:BUS:B<x>:SENT:CONDition?
```

Arguments B<x> is the number of the bus waveform.

START sets triggering on start of packet.

FAST sets triggering on fast channel packets.

SLOW sets triggering on slow channel packets.

ERRor sets triggering on errors.

Examples TRIGGER:A:BUS:B1:SENT:CONDITION START specifies triggering on start of packet.

TRIGGER:A:BUS:B1:SENT:CONDITION? might return TRIGGER:A:BUS:B1:SENT:CONDITION ERROR, indicating the bus is triggering on errors.

TRIGger:{A|B}:BUS:B<x>:SENT:ERRType

This command sets or queries the error type to be used when triggering on SENT data.

Conditions Requires option 5SRAUTOSEN.

The SENT bus trigger condition is set to ERRor.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType CRC
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

Arguments B<x> is the number of the bus waveform.

CRC specifies triggering on CRC errors.

Examples TRIGGER:A:BUS:B1:SENT:ERRTYPE CRC sets the bus to trigger on CRC errors.

TRIGGER:A:BUS:B1:SENT:ERRTYPE? might return :TRIGGER:A:BUS:B1:SENT:ERRTYPE CRC to indicate the bus is triggering on CRC errors.

TRIGger:{A|B}:BUS:B<x>:SENT:ERRType:CRC

This command sets or queries the CRC error type to be used when triggering on SENT data.

Conditions Requires option SRAUTOSEN.

The SENT bus trigger condition is set to ERRor.

The ERRType is set to CRC.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType:CRC {FAST|SLOW}
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType:CRC?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

TRIGger:{A|B}:BUS:B<x>:SENT:ERRType

Arguments B<x> is the number of the bus waveform.

FAST specifies triggering on CRC errors in only the Fast Channel.

SLOW specifies triggering on CRC errors in only the slow channel.

Examples TRIGGER:A:BUS:B1:SENT:ERRTYPE:CRC FAST sets the bus to trigger on Fast Channel CRC errors.

TRIGGER:A:BUS:B1:SENT:ERRTYPE:CRC? might return TRIGGER:A:BUS:B1:SENT:ERRTYPE:CRC FAST to indicate that the bus is triggering on Fast Channel CRC errors.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue

This command sets or queries the high binary fast channel 1 value to use when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to FAST.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:HIVALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

Arguments `Bus<x>` is the bus identifier number.

`<Qstring>` sets the Fast Channel 1 binary data high value.

Examples `TRIGger:A:BUS:B12:SENT:FAST:CHAN1A:HIVALue "XXXXXXXXXXXX"` sets the SENT bus B12 Fast Channel 1 high value to XXXXXXXXXXXX, or "don't care."

`TRIGger:A:BUS:B3:SENT:FAST:CHAN1A:HIVALue?` might return `TRIGGER:A:BUS:B3:SENT:FAST:CHAN1A:HIVALUE "0101XXXX1111"`, indicating the SENT bus B3 Fast Channel 1 high is set to the binary value 0101XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT fast packet bus data for device channel 1.

Conditions Requires option SRAUTOSEN

The SENT bus trigger condition must be set to FAST.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:
QUALifier {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue

Arguments

B<x> is the number of the bus.

EQUal sets the qualifier as Equal.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQual sets the qualifier as More Than or Equal to.

MOREThan sets the qualifier as More Than.

UNEQual sets the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

TRIGGER:A:BUS:B2:SENT:FAST:CHAN1A:QUALIFIER UNEQUAL sets the fast channel 1 data qualifier trigger to not equal for bus 2.

TRIGGER:A:BUS:B5:SENT:FAST:CHAN1A:QUALIFIER? might return TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:QUALIFIER MOREEQUAL to indicate that the fast channel 1 data qualifier trigger is set to greater than or equal for bus 5.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue

This command sets or queries the binary fast channel 1 value to be used when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The search trigger condition must be set to FAST.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:QUALifier

Arguments

B<x> is the number of the bus waveform.

<Qstring> is the Fast Channel 1 value on which to trigger.

Examples

TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE "XXXXXXXXXXXX" sets the Fast Channel 1 value to trigger on XXXXXXXXXXXX, or "don't care."

TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE? might return :TRIGGER:A:BUS:B1:SENT:FAST:CHAN1A:VALUE "0000XXXX1111" to indicate the Fast Channel 1 binary trigger value is 0000XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue

This command sets or queries the high binary fast channel 2 value to use when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to FAST.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:HIVALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

Arguments

Bus<x> is the bus identifier number.

<Qstring> sets the Fast Channel 2 high binary data value.

Examples

TRIGger:A:BUS:B2:SENT:FAST:CHAN2B:HIVALue "100000000000" sets the SENT bus B2 Fast Channel 2 high value to 100000000000.

TRIGger:A:BUS:B4:SENT:FAST:CHAN2B:HIVALue? might return TRIGGER:A:BUS:B4:SENT:FAST:CHAN2B:HIVALUE "0101XXXX1111", indicating the SENT bus B4 Fast Channel 2 high is set to the binary value 0101XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT fast packet bus data for device channel 2.

Conditions

Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to FAST.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:
QUALifier {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:QUALifier?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALue

Arguments

B<x> is the number of the bus.

EQUal sets the qualifier as Equal.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQual sets the qualifier as More Than or Equal to.

MOREThan sets the qualifier as More Than.

UNEQual sets the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

TRIGGER:A:BUS:B2:SENT:FAST:CHAN2B:QUALIFIER UNEQUAL sets the fast channel 2 data qualifier trigger to not equal for bus 2.

TRIGGER:A:BUS:B5:SENT:FAST:CHAN2B:QUALIFIER? might return TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:QUALIFIER MOREEQUAL to indicate that the fast channel 2 data qualifier trigger is set to greater than or equal for bus 5.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALue

This command sets or queries the binary fast channel 2 value to be used when triggering on a SENT bus signal. The trigger condition must be set to FAST.

Conditions Requires option SRAUTOSEN.

The search condition must be set to FAST.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier

Arguments B<x> is the number of the bus waveform.

<Qstring> is the Fast Channel 2 binary value on which to trigger.

Examples TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:VALUE "111111111111" sets the Fast Channel 2 value on which to trigger to 111111111111.

TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:VALUE? might return :TRIGGER:A:BUS:B1:SENT:FAST:CHAN2B:VALUE "000000000000" to indicate the Fast Channel 2 binary value on which to trigger is 000000000000.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:HIVALue

This command sets or queries the high binary fast message counter value to be used when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:HIVALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

BUS:B<x>:SENT:NUMCHANnel

BUS:B<x>:SENT:NIBBLECount

Arguments `Bus<x>` is the bus identifier number.

`<Qstring>` sets the Fast Channel 1 counter binary value.

Examples `TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:HIVALUE "XXXXXXXX"` sets the Fast Channel 1 secure counter high value to "don't care."

`TRIGGER:A:BUS:B3:SENT:FAST:COUNTER:HIVALUE?` might return `TRIGGER:A:BUS:B3:SENT:FAST:COUNTER:HIVALUE "11110000"` to indicate a binary value of 11110000.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT fast packet bus data for the secure format counter.

Conditions Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to FAST.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:
QUALifier {EQual|UNEQual|LESSthan|MOREthan|
LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:QUALifier?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue

Arguments

B<x> is the number of the bus.

EQUal sets the qualifier as Equal.

LESSEQual sets the qualifier as Less Than or Equal to.

LESSThan sets the qualifier as Less Than.

MOREEQual sets the qualifier as More Than or Equal to.

MOREThan sets the qualifier as More Than.

UNEQual sets the qualifier as Unequal.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

TRIGGER:A:BUS:B2:SENT:FAST:COUNTer:QUALIFIER UNEQUAL sets the fast channel 2 counter qualifier trigger to not equal for bus 2.

TRIGGER:A:BUS:B5:SENT:FAST:COUNTER:QUALIFIER? might return TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:QUALIFIER MOREEQUAL to indicate that the fast channel 2 counter qualifier trigger is set to greater than or equal for bus 5.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue

This command sets or queries the binary fast message counter value to be used when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

BUS:B<x>:SENT:NIBBLECount

BUS:B<x>:SENT:NUMCHANnel

Arguments

B<x> is the number of the bus waveform.

<Qstring> is the Fast Channel 1 fast message counter binary value on which to trigger.

Examples

TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:VALUE "XXXXXXXX" sets the Fast Channel 1 fast message counter value to "don't care."

TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:VALUE? might return :TRIGGER:A:BUS:B1:SENT:FAST:COUNTER:VALUE "00001111" to indicate the Fast Channel 1 fast message counter binary value on which to trigger is 00001111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue

This command sets or queries the binary fast message inverted nibble value to be used when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The search condition must be set to FAST.

The number of channels must be set to 1.

The nibble count must be set to 6.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue
<Qstring>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

BUS:B<x>:SENT:NIBBLECount

BUS:B<x>:SENT:NUMCHANnel

Arguments

B<x> is the number of the bus waveform.

<Qstring> is the Fast Channel 1 inverted nibble binary value on which to trigger.

Examples

`TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "XXXX"` sets the Fast Channel 1 secure inverted nibble value to "don't care."

`TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE?` might return `:TRIGGER:A:BUS:B1:SENT:FAST:INVERTNIBBLE:VALUE "1111"` to indicate a binary value of 1111.

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue

This command sets or queries the binary status value to be used when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN.

The search trigger condition must be set to FAST.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

Arguments

`B<x>` is the number of the bus waveform.

`<Qstring>` is the binary status value on which to trigger.

Examples

`TRIGger:A:BUS:B1:SENT:FAST:STATus:VALue "XXXX"` sets the binary status trigger value to XXXX.

`TRIGGER:A:BUS:B1:SENT:FAST:STATUS:VALUE?` might return `:TRIGGER:A:BUS:B1:SENT:FAST:STATUS:VALUE "1010"` to indicate a binary trigger value is 1010.

TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier

This command sets or queries the qualifier to be used when triggering on SENT pause pulses.

Conditions

Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to PAUSE.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier {EQual|UNEQual|
LESSthan|MOREthan|LESSEQual|MOREEQual|Inrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue

SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue

Arguments `B<x>` is the bus identifier number.

`EQUal` sets the qualifier as Equal.

`INrange` sets the qualifier to in range.

`LESSEQual` sets the qualifier as Less Than or Equal to.

`LESSThan` sets the qualifier as Less Than.

`MOREEQual` sets the qualifier as More Than or Equal to.

`MOREThan` sets the qualifier as More Than.

`OUTrange` sets the qualifier to out of range.

`UNEQual` sets the qualifier as Unequal.

Examples `TRIGGER:A:BUS:B2:SENT:PAUSE:QUALifier UNEQUAL` sets the pause pulse qualifier trigger to not equal for bus 2.

`TRIGGER:A:BUS:B1:SENT:PAUSE:QUALIFIER?` might return `:TRIGGER:A:BUS:B1:SENT:PAUSE:QUALIFIER INRANGE` to indicate that the pause pulse qualifier is set to trigger inside the range of ticks low and ticks high on bus 1.

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue

This command sets or queries the high binary Slow channel data value to use when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN.

The SENT bus trigger condition must be set to SLOW.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

Arguments `Bus<x>` is the bus identifier number.

`<Qstring>` sets the binary Slow channel data value.

Examples `TRIGGER:A:BUS:B1:SENT:SLOW:DATA:HIVALUE "XXXXXXXX"` sets the Slow data high value to "don't care."

`TRIGGER:A:BUS:B1:SENT:SLOW:DATA:HIVALUE?` might return `TRIGGER:A:BUS:B1:SENT:SLOW:DATA:HIVALUE "11110001"`, to indicate triggering on Slow data values that match 11110001.

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier

This command sets or queries the binary identifier value to use when triggering on a SENT bus signal.

Conditions Requires option SRAUTOSEN

The SENT bus trigger condition must be set to SLOW.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier
{EQual|UNEQual| LESSthan|MOREthan|LESSEQual|MOREEQual}
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

Arguments `B<x>` is the bus identifier number.

`EQUal` specifies the qualifier as Equal.

`UNEQual` specifies the qualifier as Not Equal to.

`LESSthan` specifies the qualifier as Less Than.

MOREthan specifies the qualifier as More Than.

LESSEQual specifies the qualifier as Less Than or Equal to.

MOREEQual specifies the qualifier as More Than or Equal to.

INrange sets the qualifier to inside a range.

OUTrange sets the qualifier to outside a range.

Examples

TRIGger:{A|B}:BUS:B2:SENT:SLOW:DATA:QUALIFIER LESSEQUAL sets the slow channel data qualifier to less than or equal for bus 2.

TRIGger:{A|B}:BUS:B5:SENT:SLOW:DATA:QUALIFIER? might return :TRIGger:{A|B}:BUS:B5:SENT:SLOW:DATA:QUALIFIER EQUAL to indicate that the slow channel data qualifier is set to equal on bus 5.

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALue

This command sets or queries the binary slow channel value to use when triggering on a SENT bus signal.

Conditions

Requires option SRAUTOSEN

The SENT bus trigger condition must be set to SLOW.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

Arguments

B<x> is the bus identifier number.

<Qstring> is the binary slow channel data value.

Examples

TRIGGER:A:BUS:B2:SENT:SLOW:DATA:VALUE "XXXXXXXX" sets the slow data value to "don't care" for bus 2.

TRIGGER:A:BUS:B4:SENT:SLOW:DATA:VALUE? might return :TRIGGER:A:BUS:B4:SENT:SLOW:DATA:VALUE "XXXX1111", to indicate triggering on slow data values that match XXXX1111 on bus 4.

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDentifier:VALue

This command sets or queries the qualifier to use when triggering on SENT slow packet bus data.

Conditions Requires option SRAUTOSEN

The SENT bus trigger condition must be set to SLOW.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDentifier:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDentifier:VALue
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SENT:CONDition

TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALue

Arguments `B<x>` is the bus identifier number.

`<Qstring>` is the binary identifier value.

Examples `TRIGGER:A:BUS:B3:SENT:SLOW:IDENTIFIER:VALUE "XXXX"` sets the identifier value to "don't care" for bus 3.

`TRIGGER:A:BUS:B1:SENT:SLOW:IDENTIFIER:VALUE?` might return `:TRIGGER:A:BUS:B1:SENT:SLOW:IDENTIFIER:VALUE "0001"` to indicate the slow channel identifier is set to 0001 on bus 1.

TRIGger:{A|B}:BUS:B<x>:SPI:CONDition

This command specifies the trigger condition for a SPI trigger. The bus number is specified by x.

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPI:CONDition {SS|STARTofframe|DATa}
TRIGger:{A|B}:BUS:B<x>:SPI:CONDition?
```

Related Commands BUS:B<x>:SPI:IDLETime

BUS:B<x>:SPI:FRAMING

Arguments B<x> is the bus identifier number.

SS specifies the Slave Selection condition.

STARTofframe is applicable when BUS:B<x>:SPI:FRAMING is set to IDLEtime. When the trigger condition is set to STARTofframe, the instrument triggers on the first SPI clock after an idle time when there are no clocks.

DATa sets the trigger condition to Master-In Slave-Out and Master-Out Slave-In.

TRIGger:{A|B}:BUS:B<x>:SPI:DATa:SIZe

This command specifies the length of the data string to be used for a SPI trigger if the trigger condition is DATa. The bus number is specified by x.

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:SIZe?
```

Arguments B<x> is the bus identifier number.

<NR1> is the length of the data string in bytes.

Examples TRIGger:A:BUS:B1:SPI:DATa:SIZe 1 sets the data size to 1 byte.

TRIGger:A:BUS:B1:SPI:DATa:SIZe? might return :TRIGGER:A:BUS:B1:SPI:DATA:SIZE 1 indicating the data size is 1 byte.

TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue

This command specifies the binary data string used for SPI triggering if the trigger condition is DATA. The bus number is specified by x.

Conditions Requires option 5-SRCOMP or SUP5-SRCOMP Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue?
```

Arguments B<x> is the bus identifier number.

<QString> specifies the data value in the specified valid format. The valid characters are 0, 1, and X for binary format.

Examples `TRIGger:A:BUS:B1:SPI:DATa:VALue "11011010"` sets the data value to 11011010.

`TRIGger:A:BUS:B1:SPI:DATa:VALue?` might return `:TRIGGER:A:BUS:B1:SPI:DATA:VALUE "XXXXXXXX"` indicating the data value is XXXXXXXX.

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

This command sets or queries the trigger condition for the specified SPMI bus.

Conditions Requires option SRPM

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition {SSC|RESet|SLEep|
SHUTdown|WAKeup|MASTERREAd|MASTERWRIte|REGREAd|REGWRIte|
DEVICEDESCMASTERREAd|DEVICEDESCSLAVEREAd|EXTREGREAd|
EXTREGWRIte|LONGEXTREGREAd|LONGEXTREGWRIte|REGOWRIte|
AUTHenticate|TRANSferbusownership|PARItyerror}
TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition?
```

Arguments Bus<x> is the bus identifier number.

SSC specifies triggering on the Sequence Start Condition.

RESet specifies triggering on the Reset command sequence.

SLEep specifies triggering on the Sleep command sequnce.

SHUTdown specifies triggering on the Shutdown command sequence.

WAKeup specifies triggering on the Wakeup command sequence.

MASTERREAd specifies triggering on the Master Read command sequence.

MASTERWRIte specifies triggering on the Master Write command sequence.

REGREAd specifies triggering on the Register Read command sequence.

REGWRIte specifies triggering on the Register Write command sequence.

DEVICEDESCMASTERREAd specifies triggering on the Device Descriptor Block Master Read command sequence

DEVICEDESCSLAVEREAd specifies triggering on the Device Descriptor Block Slave Read command sequence.

EXTREGREAd specifies triggering on the Extended Register Read command sequence.

EXTREGWRIte specifies triggering on the Extended Register Write command sequence.

LONGEXTREGREAd specifies triggering on the Extended Register Read Long command sequence.

LONGEXTREGWRIte specifies triggering on the Extended Register Write Long command sequence.

REG0WRIte specifies triggering on the Register 0 Write command sequence.

AUTHenticate specifies triggering on the Authentication command sequence.

TRANSferbusownership specifies triggering on the Transfer Bus Ownership (TBO) command sequence.

PARItyerror specifies triggering on the parity errors.

Examples

TRIGGER:A:BUS:B5:SPMI:CONDITION SLEEP specifies triggering on a Sleep command sequence for bus B5.

TRIGGER:A:BUS:B2:SPMI:CONDITION? might return :TRIGGER:A:BUS:B2:SPMI:CONDITION MASTERREAD, indicating that bus B2 is triggering on Master Read command sequences.

TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:SIZe

This command sets or queries the length of the data string, in bytes, to be used when triggering on an SPMI bus signal.

Conditions

Requires option SRPM.

The trigger condition must be set to EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, or LONGEXTREGWRIte.

Note that this a floating point value. If the condition is set to REG0WRIte, the query return value will be a fractional value (0.875) because the data size is less than a byte.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:SIZe <NR2>
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:SIZe?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue

Arguments `Bus<x>` is the bus identifier number.

`<NR2>` is the size of the data string in bytes.

Examples `TRIGGER:A:BUS:B1:SPMI:DATA:SIZE 1` sets the data value to 1 byte.

`TRIGGER:A:BUS:B1:SPMI:DATA:SIZE?` might return `:TRIGGER:A:BUS:B1:SPMI:DATA:SIZE 4` to indicate the data pattern has four bytes of data.

TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue

This command sets or queries the binary data string used for SPMI triggering if the trigger condition is MASTERREAd, MASTERWRIte, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, LONGEXTREGWRIte, or REG0WRIte.

Conditions Requires option SRPM

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

Arguments

`Bus<x>` is the bus identifier number.

`<Qstring>` is the binary data string that identifies the data value on which to trigger.

Examples

`TRIGGER:A:BUS:B1:SPMI:DATA:VALUE "XXXXXXXX"` sets the binary data value on which to trigger for bus B1 to "don't care."

`TRIGGER:A:BUS:B5:SPMI:DATA:VALUE?` might return `TRIGGER:A:BUS:B5:SPMI:DATA:VALUE "XXXX1111"` to indicate the binary data value on which to trigger for bus B5 is set to XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SPMI:MASTERADDRess:VALue

This command sets or queries the binary data string that identifies the master address used in SPMI triggering if the trigger condition is MASTERREAd, MASTERWRIte, or DEVICEDESCMASTERREAd.

Conditions

Requires option SRPM

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:MASTERADDRess:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SPMI:MASTERADDRess:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

Arguments

`Bus<x>` is the bus identifier number.

`<Qstring>` is the binary data string that identifies the master address on which to trigger.

Examples

`TRIGGER:A:BUS:B3:SPMI:MASTERADDRESS:VALUE "XX"` sets the binary master address trigger of bus B3 to "don't care."

`TRIGGER:A:BUS:B1:SPMI:MASTERADDRESS:VALUE?` might return `TRIGGER:A:BUS:B1:SPMI:MASTERADDRESS:VALUE "11"` to indicate the binary master address on which to trigger for bus B1 is set to binary 11.

TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse

This command sets or queries whether or not to trigger on No Response frames.

Conditions Requires option SRPM

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse <Qstring>
TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

Arguments `Bus<x>` is the bus identifier number.

`<Qstring>` is either TRUE or FALSE boolean value.

Examples `TRIGGER:A:BUS:B4:SPMI:NORESPONSE TRUE` set the bus B4 to look for No Response frames.

`TRIGGER:A:BUS:B1:SPMI:NORESPONSE?` might return `:TRIGGER:A:BUS:B1:SPMI:NORESPONSE FALSE` to indicate that bus B4 is not looking for No Response frames.

TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue

This command sets or queries the binary data string that identifies the register address used in SPMI triggering if the trigger condition is MASTERREAd, MASTERWRIte, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, or LONGEXTREGWRIte.

Conditions Requires option SRPM

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

Arguments `Bus<x>` is the bus identifier number.

`<Qstring>` is the binary data string that identifies the register address.

Examples

`TRIGGER:A:BUS:B2:SPMI:REGISTERADDRESS:VALUE "XXXXXXXX"` sets the binary register address of bus B2 to "don't care."

`TRIGGER:A:BUS:B1:SPMI:REGISTERADDRESS:VALUE?` might return `TRIGGER:A:BUS:B1:SPMI:REGISTERADDRESS:VALUE "XXXX1111"` to indicate the binary register address on which to trigger for bus B1 is XXXX1111.

TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRess:VALue

This command sets or queries the binary data string that identifies the slave address used in SPMI triggering if the trigger condition is RESet, SLEep, SHUTdown, WAKeup, AUTHenticate, REGREAd, REGWRIte, EXTREGREAd, EXTREGWRIte, LONGEXTREGREAd, LONGEXTREGWRIte, DEVICEDESCSLAVEREAd, or REG0WRIte.

Conditions

Requires option SRPM

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRess:VALue <Qstring>
TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRess:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition

Arguments

`Bus<x>` is the bus identifier number.

`<Qstring>` is the binary data string that identifies the slave address on which to trigger.

Examples

`TRIGGER:A:BUS:B1:SPMI:SLAVEADDRESS:VALUE "0011"` sets the binary slave address of bus B1 to 0011.

`:TRIGGER:A:BUS:B4:SPMI:SLAVEADDRESS:VALUE?` might return `TRIGGER:A:BUS:B4:SPMI:SLAVEADDRESS:VALUE "1010"` to indicate that bus B4 is triggering on binary slave address1010.

TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:HIVALue

This command specifies the binary address string for the upper limit for inside-of-range and outside-of-range qualifiers for the USB trigger. Use the command TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue to set the lower limit. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:HIVALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:CONDition

TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue

Arguments `B<x>` is the bus identifier number.

`<QString>` within the range 0000000 to 1111111 (00 hex to 7F hex).

Examples `TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE "0001000"` sets the upper limit to binary 0001000 (08 hex).

`TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE?` might return `:TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE "1111111"`, which indicates that the upper limit is 1111111 (7F hex).

TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue

This command specifies the binary address string to be used for USB trigger.

The trigger condition must be set to TOKEN. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments B<x> is the bus identifier number.

<QString> within the range 0000000 to 1111111 (00 hex to 7F hex).

Examples TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE "0001000" sets the binary address to 0001000 (08 hex).

TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE? might return :TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE "1000000", which indicates that the binary address is 100000 (40 hex).

TRIGger:{A|B}:BUS:B<x>:USB:CONDition

This command specifies the trigger condition for the USB trigger. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:CONDition
{SYNC|RESET|SUSPEND|RESUME|EOP|TOKENPacket|DATAPacket|
HANDSHAKEPacket|SPECIALPacket|ERRor}
```

```
TRIGger:{A|B}:BUS:B<x>:USB:CONDition?
```

Arguments SYNC indicates triggering on a Sync field of a packet.

RESET sets triggering on a reset condition.

SUSPEND sets triggering on a suspend condition.

RESUME sets triggering on a resume condition.

EOP indicates triggering on an end-of-packet signal.

TOKENPacket indicates triggering on a token packet.

DATAPacket indicates triggering on a data packet

HANDSHAKEPacket indicates triggering on a handshake packet.

SPECIALPacket indicates triggering on a special status packet.

ERRor indicates triggering on an error condition.

Examples

`TRIGGER:A:BUS:B1:USB:CONDITION TOKENPACKET` sets the trigger condition to be a token packet.

`TRIGGER:A:BUS:B1:USB:CONDITION?` might return `:TRIGGER:A:BUS:B1:USB:CONDITION SYNC`, which indicates that the trigger condition is a sync field.

TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue

This command specifies the binary data string for the upper limit for inside-of-range and outside-of-range qualifiers for the USB trigger when the trigger condition is DATA. Use the command TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue to set the lower limit. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition

TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue

Arguments

`<QString>` within the range 00000000 to 11111111 (00 hex to FF hex).

Examples

`TRIGGER:A:BUS:B1:USB:DATA:HIVALUE "00001000"` sets the upper limit to binary 00001000 (08 hex).

`TRIGGER:A:BUS:B1:USB:DATA:HIVALUE?` might return `:TRIGGER:A:BUS:B1:USB:DATA:HIVALUE "01111111"`, which indicates that the upper limit is 01111111 (7F hex).

TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet

This command specifies the data offset in bytes to trigger on. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet <NR1>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments <NR1> is number in the range 0 to 1024.

Examples `TRIGGER:A:BUS:B1:USB:DATA:OFFSET 36` sets the data offset to 36.

`TRIGGER:A:BUS:B1:USB:DATA:OFFSET?` might return `:TRIGGER:A:BUS:B1:USB:DATA:OFFSET 0`, indicating that the data offset is the default value, 0.

TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier

This command sets the qualifier to be used when triggering on a USB bus signal. The trigger condition must be set to DATAPACKET. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier {EQual|UNEQual|
LESSthan|MOREthan|LESSEQual|MOREEQual|INrange|OUTrange}
TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier?
```

Arguments

`EQUal` specifies the qualifier as Equal.

`INrange` specifies the qualifier as Inside Range.

`LESSEQual` specifies the qualifier as Less Than or Equal to.

`MOREEQual` specifies the qualifier as More Than or Equal to.

`OUTrange` specifies the qualifier as Out of Range.

`UNEQual` specifies the qualifier as Not Equal to.

`LESSthan` specifies the qualifier as Less Than.

MOREthan specifies the qualifier as More Than.

Examples

TRIGger:A:BUS:B1:USB:DATa:QUALifier LESSTHAN sets the qualifier to less than.

TRIGger:A:BUS:B1:USB:DATa:QUALifier? might return :TRIGGER:A:BUS:B1:USB:DATA:QUALIFIER EQUAL indicating the qualifier is set to equal.

TRIGger:{A|B}:BUS:B<x>:USB:DATa:SIZe

This command specifies the number of contiguous data bytes to trigger on. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:SIZe <NR1>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:SIZe?
```

Arguments

<NR1> is number in the range 1 to 16.

Examples

TRIGGER:A:BUS:B1:USB:DATA:SIZE 4 sets the instrument to trigger on four contiguous data bytes.

TRIGGER:A:BUS:B1:USB:DATA:SIZE? might return :TRIGGER:A:BUS:B1:USB:DATA:SIZE 6 indicating that the instrument will trigger on 6 contiguous data bytes.

TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPe

This command specifies the data type for a USB trigger. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPe
{ANY|DATA0|DATA1|DATA2|MDATA}
TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPe?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments

ANY indicates either a DATA0 or DATA1 data packet type.

DATA0 indicates a DATA0 data packet type.

DATA1 indicates a DATA1 data packet type.

DATA2 indicates a DATA2 data packet type when on HIGH speed.

MDATA indicates a MDATA data packet type when on HIGH speed.

Examples

`TRIGGER:A:BUS:B1:USB:DATA:TYPE DATA0` sets the instrument to trigger on a DATA0 data packet type.

`TRIGGER:A:BUS:B1:USB:DATA:TYPE?` might return `:TRIGGER:A:BUS:B1:USB:DATA:TYPE DATA1`, indicating that the data packet type is set to DATA1.

TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue

This command specifies the binary data string to be used when triggering on a USB trigger.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments

<QString> within the range 00000000 to 11111111 (00 hex to FF hex).

Examples

`TRIGGER:A:BUS:B1:USB:DATA:VALUE "00001000"` sets the binary address to 00001000 (08 hex).

TRIGGER:A:BUS:B1:USB:DATA:VALUE? might return :TRIGGER:A:BUS:B1:USB:DATA:VALUE "01000000", which indicates that the binary address is 0100000 (40 hex).

TRIGger:{A|B}:BUS:B<x>:USB:ENDPoint:VALue

This command specifies the binary endpoint string to be used for the USB trigger. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:ENDPoint:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:ENDPoint:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments <QString> within the range 0000 to 1111 (00 hex to 0F hex).

Examples TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE "1000" sets the binary address to 1000 (08 hex).

TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE? might return :TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE "1001", which indicates that the binary address is 1001 (09 hex).

TRIGger:{A|B}:BUS:B<x>:USB:ERRType

This command specifies the error type to be used when the trigger condition is set to ERRor. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:ERRType
{PID|CRC5|CRC16|BITSTUFFing}
TRIGger:{A|B}:BUS:B<x>:USB:ERRType?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments

`PID` indicates the error type is set to packet ID.

`CRC5` indicates the error type is set to 5-bit CRC.

`CRC16` indicates the error type is set to 16-bit CRC.

`BITSTUFFing` indicates the error type is set to bit stuffing.

Examples

`TRIGGER:A:BUS:B1:USB:ERRTYPE PID` sets the error trigger condition to packet ID.

`TRIGGER:A:BUS:B1:USB:ERRTYPE?` might return `:TRIGGER:A:BUS:B1:USB:TYPE PID`, indicating the error trigger condition is packet ID.

TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType

This command specifies the handshake type for the USB trigger. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType
{ANY|NAK|ACK|STALL|NYET}
TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments

`ANY` indicates the instrument will trigger on any handshake type.

`NAK` indicates the instrument will trigger when a device cannot send or receive data.

`ACK` indicates the instrument will trigger when a packet is successfully received.

STALL indicates the instrument will trigger when a device requires intervention from the host.

NYET specifies the handshake type as No response Yet (0110).

Examples

TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE ACK sets the handshake type to acknowledgement.

TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE? might return :TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE STALL, indicating that the handshake type is set to stall.

TRIGger:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber

This command specifies the binary data string to be used for start of frame number, when the trigger condition is Token Packet and the token type is Start of Frame. The bus number is specified by x.

Conditions

Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group

Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber <QString>
TRIGger:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber?
```

Related Commands

TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments

<QString> within the range 000 0000 0000 to 111 1111 1111 (000 hex to 7FF hex).

Examples

TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER "00000001000" sets the start of frame number to 00000001000 (008 hex).

TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER? might return :TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER "00000001001", which indicates that the start of frame number is 00000001001 (009 hex).

TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType

This command specifies the packet ID (PID) for the special packet. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType
{ANY|ERROr|PING|PREamble|RESERVed|SPLit}
TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments `ANY` specifies the PID value as Any (XX00).

`ERROr` specifies the PID value as ERR (1100).

`PING` specifies the PID value as PING (0100).

`PREamble` specifies the PID value as PRE (1100).

`RESERVed` specifies the PID value as Reserved (0000).

`SPLit` specifies the PID value as Split (1000).

Examples `TRIGGER:A:BUS1:USB:SPECIALTYPE PREAMBLE` sets the special packet type to preamble.

`TRIGGER:A:BUS1:USB:SPECIALTYPE?` might return `:TRIGGER:A:BUS1:USB:SPECIALTYPE PREAMBLE`, indicating that the special type is set to preamble.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue

When triggering on a high-speed USB split transaction, this command specifies the split transaction endpoint type value to trigger on. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue
{NOCARE|CONTRol|ISOchronous|BULK|INTERRUPT}
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType

Arguments NOCARE — any endpoint type.

CONTRol — control endpoint type.

ISOchronous — isochronous endpoint type.

BULK — bulk endpoint type (BULK-IN or BULK-OUT).

INTERRUPT — interrupt endpoint type (Interrupt-IN).

Examples TRIGger:A:BUS:B1:USB:SPLit:ET:VALue iso sets the endpoint type to trigger on to isochronous.

TRIGger:A:BUS:B1:USB:SPLit:ET:VALue? might return BULK, indicating that the bulk endpoint type has been specified to trigger on.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:HUB:VALue

When triggering on a high-speed USB split transaction, this command specifies the split transaction hub address value to trigger on. The trigger condition must be set to Special with packet type SPLIT. The value can be up to 7 characters long. The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:HUB:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:HUB:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType

Arguments QString is a quoted string of up to 7 characters. The valid characters are 0 and 1.

Examples TRIGger:A:BUS:B1:USB:SPLit:HUB:VALue "001010" sets the split transaction hub address to trigger on to 001010.

TRIGger:A:BUS:B1:USB:SPLit:HUB:VALue? might return XXXXXXX, indicating that the hub address value to trigger on doesn't matter.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue

When triggering on a high-speed USB split transaction, this command specifies the split transaction port address value to trigger on. The trigger condition must be set to Special with a packet type SPLIT. The value can be up to 7 characters long. The default is all X's (don't care). The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue <QString>
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType

Arguments `QString` is a quoted string of up to 7 characters. The valid characters are 0 and 1.

Examples `TRIGger:A:BUS:B1:USB:SPLit:PORT:VALue` "001010" sets the split transaction hub address to trigger on to 001010.

`TRIGger:A:BUS:B1:USB:SPLit:PORT:VALue?` might return XXXXXXX, indicating that the hub address value to trigger on doesn't matter.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue

When triggering on a high-speed USB split transaction, this command specifies whether to trigger on the start or complete phase of the split transaction, based on the Start/Complete bit field value. (0 = Start, 1 = Complete). The default is NOCARE. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue
{NOCARE|SSPLIT|CSPLIT}
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType

Arguments

NOCARE — trigger on either the start or complete phase of the split transaction.

SSPLIT — trigger on the start phase of the split transaction.

CSPLIT — trigger on the complete phase of the split transaction.

Examples

TRIGger:A:BUS:B1:USB:SPLit:SC:VALue SSPLIT specifies to trigger on the start phase of the split transaction.

TRIGger:A:BUS:B1:USB:SPLit:SC:VALue? might indicate NOCARE, specifying that it doesn't matter whether to trigger on the start or complete phase of the split transaction.

TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SE:VALue

When triggering on a high-speed USB split transaction, this command specifies the split transaction start/end bit value to trigger on. The bus number is specified by x.

NOTE. *The start and end bits are interpreted based on the type of split transaction:*

For Interrupt and control transactions, the S bit means Speed: 0 = Full Speed, 1 = Low Speed.

For bulk IN/OUT and isochronous IN start-split transactions, the S field must be 0.

For bulk/control IN/OUT, interrupt IN/OUT, and isochronous IN start-split transactions, the E field must be 0.

For full-speed isochronous OUT start-split transactions, the S (Start) and E (End) fields specify how the high-speed data payload corresponds to data for a full-speed data packet as shown below:

S E High-speed to Full-speed Data Relation

0 0 High-speed data is the middle of the full-speed data payload.

0 1 High-speed data is the end of the full-speed data payload.

1 0 High-speed data is the beginning of the full-speed data payload.

1 1 High-speed data is all of the full speed data payload.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SE:VALue
{NOCARE|FULLSPEED|LOWSPEED|ISOSTART|ISOMID|ISOEND|ISOALL}
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SE:VALue?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType

Arguments

`NOCARE` — any combination of S and E bits.

`FULLSPEED` — S bit = 0, E bit = 0.

`LOWSPEED` — S bit = 1, E bit = 0.

`ISOSTART` — S bit = 1, E bit = 0.

`ISOMID` — see note above.

`ISOEND` — see note above.

`ISOALL` — see note above.

Examples

`TRIGger:A:BUS:B1:USB:SPLit:SE:VALue ISOEND` specifies to trigger on the ISOEND split transaction value.

`TRIGger:A:BUS:B1:USB:SPLit:SE:VALue?` might return FULLSPEED.

TRIGger:{A|B}:BUS:B<x>:USB:TOKENType

This command specifies the token type for the USB trigger. The bus number is specified by x.

Conditions Requires option 5-SRUSB2 or SUP5-SRUSB2 Triggering and Analysis application.

Group Trigger

Syntax
```
TRIGger:{A|B}:BUS:B<x>:USB:TOKENType {ANY|SOF|OUT|IN|SETUP}
TRIGger:{A|B}:BUS:B<x>:USB:TOKENType?
```

Related Commands TRIGger:{A|B}:BUS:B<x>:USB:CONDition

Arguments

`ANY` indicates any of the token types.

`SOF` indicates a SOF (start-of-frame) token type

`OUT` indicates an OUT token type.

`IN` indicates an IN token type.

`SETUP` indicates a SETUP token type.

Examples

`TRIGGER:A:BUS:B1:USB:TOKENTYPE SETUP` sets the token type to SETUP.

`TRIGGER:A:BUS:B1:USB:TOKENTYPE?` might return `:TRIGGER:A:BUS:B1:USB:TOKENTYPE SOF` if the token type is SOF.

TRIGger:{A|B}:BUS:SOUrce

This command sets or queries the source bus for a bus trigger.

Group Trigger

Syntax

```
TRIGger:{A|B}:BUS:SOUrce B<x>
TRIGger:{A|B}:BUS:SOUrce?
```

Arguments

`B<x>` sets the selected source to the bus.

Examples

`TRIGGER:A:BUS:SOURCE B2` sets the selected source for the bus trigger to Bus 2.

`TRIGGER:A:BUS:SOURCE?` might return `TRIGGER:A:BUS:SOURCE B1`, indicating that the selected source for the bus trigger is set to Bus 1.

TRIGger:{A|B}:EDGE:COUPling

This command sets or queries the type of coupling for the edge trigger. This command is equivalent to selecting Edge from the Trigger Type drop-down in the Trigger setup context menu, and choosing from the Coupling drop-down list.

Group Trigger

Syntax

```
TRIGger:{A|B}:EDGE:COUPling {DC|HFRej|LFRej|NOISErej}
TRIGger:{A|B}:EDGE:COUPling?
```

Related Commands

TRIGger:{A|B}:EDGE:SOUrce

TRIGger:{A|B}:EDGE:SLOpe

Arguments

`DC` selects DC trigger coupling, which passes all input signals to the trigger circuitry.

`HFREj` coupling attenuates signals above 50 kHz before passing the signals to the trigger circuitry.

`LFREj` coupling attenuates signals below 80 kHz before passing the signals to the trigger circuitry.

`NOISEREj` coupling provides stable triggering by increasing the trigger hysteresis. Increased hysteresis reduces the trigger sensitivity to noise but can require greater trigger signal amplitude.

Examples

`TRIGGER:A:EDGE:COUPLING DC` sets the A edge trigger coupling to DC.

`TRIGGER:A:EDGE:COUPLING?` might return `:TRIGGER:A:EDGE:COUPLING DC`, indicating that the A edge trigger coupling is set to DC.

TRIGger:{A|B}:EDGE:SLOpe

This command sets or queries the slope for the edge trigger. This command is equivalent to selecting Edge from the Trigger Type drop-down in the Trigger setup context menu, and then choosing the desired Slope. This command is also equivalent to pressing the front-panel Slope button.

Group

Trigger

Syntax

```
TRIGger:{A|B}:EDGE:SLOpe {RISe|FALL|EITher}
TRIGger:{A|B}:EDGE:SLOpe?
```

Related Commands

TRIGger:{A|B}:EDGE:SOUrce

TRIGger:{A|B}:EDGE:COUPling

TRIGger:B:STATE

Arguments

`RISe` specifies to trigger on the rising or positive edge of a signal.

`FALL` specifies to trigger on the falling or negative edge of a signal.

`EITHER` specifies to trigger on either the rising or falling edge of a signal.

Examples

`TRIGGER:A:EDGE:SLOPE RISE` sets the A edge trigger slope to positive, which triggers on the rising edge of the signal.

`TRIGGER:A:EDGE:SLOPE?` might return `:TRIGGER:A:EDGE:SLOPE FALL`, indicating that the A edge trigger slope is negative.

TRIGger:{A|B}:EDGE:SOUrce

This command sets or queries the source for the edge trigger. For instruments that have an Auxiliary Input (such as the MSO58LP), AUXiliary can be selected as trigger source.

Group

Trigger

Syntax

```
TRIGger:{A|B}:EDGE:SOUrce {CH<x>|CH<x>_D<y>|LINE|AUXiliary}
TRIGger:{A|B}:EDGE:SOUrce?
```

Related Commands

TRIGger:{A|B}:EDGE:SLOpe

TRIGger:{A|B}:EDGE:COUPling

TRIGger:B:STATE

Arguments

`CH<x>` specifies an analog channel as the edge trigger source.

`CH<x>_D<y>` specifies a digital channel as the edge trigger source.

`LINE` specifies AC line voltage, and is a valid source when B trigger is inactive.

`AUXiliary` specifies the Auxiliary Input.

Examples

`TRIGGER:A:EDGE:SOURCE CH2` sets the A edge trigger source to input channel 2.

`TRIGGER:A:EDGE:SOURCE?` might return `:TRIGGER:A:EDGE:SOURCE CH1`, indicating that the A edge trigger source is set to input channel 1.

TRIGger:{A|B}:LEVel:CH<x>

This command sets or queries the CH<x> trigger level for an Edge, Pulse Width, Runt or Rise/Fall (Transition and Slew Rate) trigger when triggering on an analog channel waveform. Each channel can have an independent trigger level. The <x> is the channel number.

Group Trigger

Syntax
```
TRIGger:{A|B}:LEVel:CH<x> <NR3>
TRIGger:{A|B}:LEVel:CH<x>?
```

Arguments <NR3> specifies the trigger level in user units (usually volts).

Examples `TRIGGER:A:LEVEL:CH1 1.5` sets the A trigger level for Channel 1 to 1.5 V.

`TRIGGER:A:LEVEL:CH2?` might return `:TRIGGER:A:LEVEL:CH2 1.3000E+00`, indicating that the A trigger level for Channel 2 is set to 1.3 V.

TRIGger:{A|B}:LOGIc:DELTatime

This command specifies or queries the Logic trigger delta time value. The time value is used as part of the Logic trigger condition to determine if the duration of a logic pattern meets the specified time constraints.

Group Trigger

Syntax
```
TRIGger:{A|B}:LOGIc:DELTatime <NR3>
TRIGger:{A|B}:LOGIc:DELTatime?
```

Arguments <NR3> the Logic trigger delta time value.

Examples `TRIGger:A:LOGIc:DELTatime 4.5e-9` sets the delta time to 4.5 ns.

`TRIGger:A:LOGIc:DELTatime?` might return `:TRIGGER:A:LOGIC:DELTATIME 4.0E-9` indicating the delta time is set to 4 ns.

TRIGger:{A|B}:LOGIc:FUNCtion

This command sets or queries the logical combination of the input channels for logic triggers. This command is equivalent to selecting Logic for the Trigger Type, and setting or viewing the Define Logic.

Group Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:FUNCtion {AND|NANd|NOR|OR}
TRIGger:{A|B}:LOGIc:FUNCtion?
```

Arguments

AND specifies to trigger if all conditions are true.

NANd specifies to trigger if any of the conditions are false.

NOR specifies to trigger if all conditions are false.

OR specifies to trigger if any of the conditions are true.

Examples

TRIGGER:A:LOGIC:FUNCTION AND sets the logical combination of channels to be true when all conditions are true.

TRIGGER:A:LOGIC:FUNCTION? might return :TRIGGER:A:LOGIC:FUNCTION NAND, indicating that the instrument will trigger if the AND logic conditions are false.

TRIGger:{A|B}:LOGIc:INPut:CLOCk:SOUrce

This command specifies the channel to use as the clock source for logic trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:INPut:CLOCk:SOUrce {CH<x>|CH<x>_D<y>}
TRIGger:{A|B}:LOGIc:INPut:CLOCk:SOUrce?
```

Arguments

CH<x> specifies an analog channel as the clock source. Number of channels depends on instrument configuration.

CH<x>_D<y> specifies a digital channel as the clock source. Number of channels depends on instrument configuration.

Examples

TRIGger:A:LOGIc:INPut:CLOCk:SOUrce CH3 sets the clock source to channel 3.

TRIGger:A:LOGIc:INPut:CLOCk:SOUrce? might return TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE CH3 indicating the clock source is set to channel 3.

TRIGger:{A|B}:LOGIc:POLarity

This command sets or queries the polarity for the clock channel when Use Clock Edge is set to Yes for Logic trigger type.

Group Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:POLarity {POSitive|NEGative|EITher}
TRIGger:{A|B}:LOGIc:POLarity?
```

Arguments

`NEGative` specifies negative polarity.

`POSITIVe` specifies positive polarity.

`EITher` specifies either polarity.

Examples

`TRIGger:A:LOGIc:POLarity EITHER` sets the polarity to either positive or negative.

`TRIGger:A:LOGIc:POLarity?` might return `:TRIGGER:A:LOGIC:POLARITY POSITIVE` indicating the polarity is set to positive.

TRIGger:{A|B}:LOGIc:USECLockedge

This command specifies whether or not Logic trigger type uses clock source.

Group Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:USECLockedge {OFF|ON|0|1}
TRIGger:{A|B}:LOGIc:USECLockedge?
```

Arguments

`ON` specifies that logic trigger type uses clock source.

`OFF` specifies that logic trigger type does not use clock source.

`<NR1>` = 0 specifies that logic trigger type does not use clock source; any other value uses clock source.

Examples

`TRIGger:A:LOGIc:USECLockedge OFF` specifies that the clock edge will not be used.

TRIGger:A:LOGIc:USECLockedge? might return :TRIGGER:A:LOGIC:USECLOCKEDGE 1 indicating that the clock edge will be used.

TRIGger:{A|B}:LOGIc:WHEn

This command sets or queries the condition for generating an A or B logic trigger with respect to the defined input pattern. This command is equivalent to selecting Logic for Trigger Type, Use Clock Edge to No, and choosing a trigger condition from the Logic Pattern drop-down list.

Group Trigger

Syntax

```
TRIGger:{A|B}:LOGIc:WHEn
{TRUe|FALSe|MOREThan|LESSThan|EQual| UNEQual}
TRIGger:{A|B}:LOGIc:WHEn?
```

Arguments

TRUe triggers on an input pattern that is true.

FALSe triggers on an input pattern that is false.

MOREthan triggers on an input pattern that is true for a time period greater than a user defined Time Limit (DELTatime) value.

LESSthan triggers on an input pattern that is true for a time period less than a user defined Time Limit (DELTatime) value.

EQual triggers on an input pattern that is true for a time period equal to a user defined Time Limit (DELTatime) value.

UNEQual triggers on an input pattern that is true for a time period not equal to a user defined Time Limit (DELTatime) value.

Examples

TRIGger:A:LOGIc:WHEn EQual specifies triggering when the input pattern is true for a time period equal to a user defined Time Limit value.

TRIGger:A:LOGIc:WHEn? might return :TRIGGER:A:LOGIC:WHEN TRUE indicating a trigger when the logic is true.

TRIGger:{A|B}:LOGICPattern:{CH<x>|CH<x>_D<x>}

This command sets or queries the Logic Pattern that is used along with the Define Logic choice (LOGIc:FUNCtion) to determine when the logic trigger occurs.

Group Trigger

Syntax

```
TRIGger:{A|B}:LOGICPattern:{CH<x>|CH<x>_D<x>} {HIGH|LOW|X}
TRIGger:{A|B}:LOGICPattern:{CH<x>|CH<x>_D<x>}?
```

Arguments

HIGH specifies the logic high.

LOW specifies the logic low.

X specifies a don't care state.

Examples

TRIGger:A:LOGICPattern:CH1 HIGH specifies triggering on a logic high.

TRIGger:A:LOGICPattern:CH1? might return :TRIGGER:A:LOGICPATTERN:CH1 X indicating a don't care state for channel 1.

TRIGger:A:LOGICPattern:CH1_D0 HIGH specifies a logic high.

TRIGger:A:LOGICPattern:CH1_D0? might return :TRIGGER:A:LOGICPATTERN:CH1_D0 X indicating a don't care.

TRIGger:{A|B}:LOWerthreshold:CH<x>

This command sets or queries the A or B lower trigger level threshold for the channel, specified by x.

Group

Trigger

Syntax

```
TRIGger:{A|B}:LOWerthreshold:CH<x> <NR3>
TRIGger:{A|B}:LOWerthreshold:CH<x>?
```

Related Commands

TRIGger:{A|B}:UPPerthreshold:CH<x>

Arguments

<NR3> specifies the threshold voltage in user units.

Examples

TRIGGER:A:LOWERTHRESHOLD:CH2 1.3 sets the A trigger threshold voltage for Channel 2 to 1.3 V.

TRIGGER:A:LOWERTHRESHOLD:CH3? might return TRIGGER:A:LOWERTHRESHOLD:CH3 1.2000E+00, indicating that the A trigger threshold voltage for Channel 3 is 1.2 V.

TRIGger:{A|B}:PULSEWidth:HIGHLimit

This command specifies the upper limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values. (Use TRIGger:{A|B}:PULSEWidth:LOWLimit to specify the lower value of the range.)

Group Trigger

Syntax
```
TRIGger:{A|B}:PULSEWidth:HIGHLimit <NR3>
TRIGger:{A|B}:PULSEWidth:HIGHLimit?
```

Related Commands TRIGger:{A|B}:PULSEWidth:WHEn

TRIGger:{A|B}:PULSEWidth:LOWLimit

Arguments <NR3> is a floating point number that represents the higher value of the range.

Examples `TRIGger:A:PULSEWidth:HIGHLimit 110.0E-9` sets the high limit to 110.0 ns.

`TRIGger:A:PULSEWidth:HIGHLimit?` might return `TRIGGER:A:PULSEWIDTH:HIGHLIMIT 178.88000E-9` indicates the high limit is set to 178.88 ns.

TRIGger:{A|B}:PULSEWidth:LOGICQUALification

This command specifies whether or not to use logic qualification for a pulse width trigger.

Group Trigger

Syntax
```
TRIGger:{A|B}:PULSEWidth:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:PULSEWidth:LOGICQUALification?
```

Arguments ON specifies that the pulse width trigger type uses logic qualification.

OFF specifies that the pulse width trigger type does not use logic qualification.

Examples `TRIGger:A:PULSEWidth:LOGICQUALification ON` turns on logic qualification.

`TRIGger:A:PULSEWidth:LOGICQUALification?` might return `TRIGGER:A:PULSEWIDTH:LOGICQUALIFICATION OFF` indicating logic qualification is off.

TRIGger:{A|B}:PULSEWidth:LOWLimit

This command specifies the lower limit to use, in seconds, when triggering on detection of a pulse whose duration is inside or outside a range of two values. (Use TRIGger:{A|B}:PULSEWidth:HIGHLimit to specify the upper limit of the range.)

This command also specifies the single limit to use, in seconds, when triggering on detection of a pulse whose duration is less than, greater than, equal to, or not equal to this time limit.

Group Trigger

Syntax

```
TRIGger:{A|B}:PULSEWidth:LOWLimit <NR3>
TRIGger:{A|B}:PULSEWidth:LOWLimit?
```

Related Commands TRIGger:{A|B}:PULSEWidth:WHEn

TRIGger:{A|B}:PULSEWidth:HIGHLimit

Arguments <NR3> is a floating point number that represents the lower value of the range.

Examples `TRIGger:A:PULSEWidth:LOWLimit 100.0E-9` sets the low limit to 100.0 ns.

`TRIGger:A:PULSEWidth:HIGHLimit?` might return `TRIGGER:A:PULSEWIDTH:LOWLIMIT 77.7600E-9` indicating the low limit is set to 77.76 ns.

TRIGger:{A|B}:PULSEWidth:POLarity

This command specifies the polarity for a pulse width trigger.

Group Trigger

Syntax

```
TRIGger:{A|B}:PULSEWidth:POLarity {NEGative|POSitive}
TRIGger:{A|B}:PULSEWidth:POLarity?
```

Arguments

NEGative specifies a negative pulse.

POSitive specifies a positive pulse.

Examples

TRIGger:A:PULSEWidth:POLarity NEGATIVE sets the pulse polarity to negative.

TRIGger:A:PULSEWidth:POLarity? might return TRIGGER:A:WIDTH:POLARITY POSITIVE indicating a positive pulse.

TRIGger:{A|B}:PULSEWidth:SOUrce

This command specifies the source waveform for a pulse width trigger.

Group

Trigger

Syntax

TRIGger:{A|B}:PULSEWidth:SOUrce {CH<x>|CH<x>_D<y>}

TRIGger:{A|B}:PULSEWidth:SOUrce?

Arguments

CH<x> specifies an analog input channel as the pulse-width trigger source.

CH<x>_D,y> specifies an digital input channel as the pulse-width trigger source.

Examples

TRIGGER:A:PULSEWIDTH:SOURCE CH1 sets channel 1 as the pulse width source.

TRIGGER:A:PULSEWIDTH:SOURCE? might return :TRIGGER:A:PULSEWIDTH:SOURCE CH1 indicating that channel 1 is the pulse width trigger source.

TRIGger:{A|B}:PULSEWidth:WHEn

This command specifies to trigger when a pulse is detected with a width (duration) that is less than, greater than, equal to, or unequal to a specified value (set using TRIGger:{A|B}:PULSEWidth:LOWLimit), OR whose width falls outside of or within a specified range of two values (set using TRIGger:{A|B}:PULSEWidth:LOWLimit and TRIGger:{A|B}:PULSEWidth:HIGHLimit).

Group

Trigger

Syntax

```
TRIGger:{A|B}:PULSEWidth:WHEn
{LESSthan|MOREthan|EQual|UNEQual|WIThin|OUTside}
TRIGger:{A|B}:PULSEWidth:WHEn?
```

Related Commands

TRIGger:{A|B}:PULSEWidth:LOWLimit

TRIGger:{A|B}:PULSEWidth:HIGHLimit

TRIGger:{A|B}:PULSEWidth:SOUrce

Arguments

`LESSthan` causes a trigger when a pulse is detected with a width less than the time set by the TRIGger:{A|B}:PULSEWidth:LOWLimit command.

`MOREthan` causes a trigger when a pulse is detected with a width greater than the time set by the TRIGger:{A|B}:PULSEWidth:LOWLimit command.

`EQual` causes a trigger when a pulse is detected with a width equal to the time period specified in TRIGger:{A|B}:PULSEWidth:LOWLimit within a ±5% tolerance.

`UNEQual` causes a trigger when a pulse is detected with a width greater than or less than (but not equal) the time period specified in TRIGger:{A|B}:PULSEWidth:LOWLimit within a ±5% tolerance.

`WIThin` causes a trigger when a pulse is detected that is within a range set by two values.

`OUTside` causes a trigger when a pulse is detected that is outside of a range set by two values.

Examples

`:TRIGger:B:PULSEWidth:WHEn LESSthan` causes the instrument to trigger when a pulse is detected that is shorter than the time specified by TRIGger:{A|B}:PULSEWidth:LOWLimit.

`TRIGger:A:PULSEWidth:WHEn ?` might return `TRIGGER:A:PULSEWIDTH:WHEN GREATER THAN 2.0000E-9`, indicating that a trigger is generated when a pulse is detected greater than 2 ns.

`TRIGger:B:PULSEWidth:WHEn?` might return `TRIGger:B:PULSEWidth:WHEn MOREthan` indicating that a trigger is generated when a pulse is detected that is greater than the time specified by TRIGger:{A|B}:PULSEWidth:LOWLimit.

TRIGger:{A|B}:RUNT:LOGICQUALification

This command specifies whether or not to use logic qualification for a runt trigger.

Group Trigger

Syntax

```
TRIGger:{A|B}:RUNT:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:RUNT:LOGICQUALification?
```

Arguments

ON specifies that the runt trigger type uses logic qualification.

OFF specifies that the runt trigger type does not use logic qualification.

Examples

TRIGger:A:RUNT:LOGICQUALification ON turns on logic qualification.

TRIGger:A:RUNT:LOGICQUALification? might return :TRIGGER:A:RUNT:LOGICQUALIFICATION OFF indicating logic qualification is off.

TRIGger:{A|B}:RUNT:POLarity

This command specifies the polarity for the runt trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:RUNT:POLarity {EITher|NEGative|POSitive}
TRIGger:{A|B}:RUNT:POLarity?
```

Arguments

POSitive indicates that the rising edge crosses the low threshold and the falling edge recrosses the low threshold without either edge ever crossing the high threshold.

NEGative indicates that the falling edge crosses the high threshold and the rising edge recrosses the high threshold without either edge ever crossing the low threshold.

EITher triggers on a runt of either polarity.

Examples

TRIGger:A:RUNT:POLarity NEGATIVE specifies that the polarity of the A pulse runt trigger is negative.

TRIGger:A:RUNT:POLarity? might return :TRIGGER:A:RUNT:POLARITY POSITIVE indicating that the polarity of the A pulse runt trigger is positive.

TRIGger:{A|B}:RUNT:SOUrce

This command specifies the source waveform for the runt trigger.

NOTE. *Digital channels are not supported as runt trigger sources.*

Group Trigger

Syntax
```
TRIGger:{A|B}:RUNT:SOUrce {CH<x>}
TRIGger:{A|B}:RUNT:SOUrce?
```

Arguments `CH<x>` specifies the analog channel number to use as the source waveform for the runt trigger. To specify the threshold levels when using CH<x> as the source, use TRIGger:{A|B}:LOWerthreshold:CH<x> and TRIGger:{A|B}:UPPerthreshold:CH<x>.

Examples `TRIGger:A:RUNT:SOUrce CH4` sets channel 4 as the trigger source.

`TRIGger:A:RUNT:SOUrce?` might return `:TRIGGER:A:RUNT:SOURCE CH2` indicating that channel 2 is the trigger source.

TRIGger:{A|B}:RUNT:WHEn

This command specifies the type of pulse width the trigger checks for when it detects a runt.

Group Trigger

Syntax
```
TRIGger:{A|B}:RUNT:WHEn
{LESSthan|MOREthan|EQual|UNEQual|OCCURS}
TRIGger:{A|B}:RUNT:WHEn?
```

Related Commands TRIGger:{A|B}:RUNT:WIDth

Arguments `OCCURS` sets the instrument to trigger if a runt signal of any detectable width occurs.

`LESSthan` sets the instrument to trigger if the a runt pulse is detected with width less than the time set by the TRIGger:{A|B}:RUNT:WIDth command.

`MOREthan` sets the instrument to trigger if the a runt pulse is detected with width greater than the time set by the TRIGger:{A|B}:RUNT:WIDth command.

`EQual` sets the instrument to trigger if a runt pulse is detected with width equal to the time period specified in TRIGger:{A|B}:RUNT:WIDth within a ±5% tolerance.

`NOTEQual` sets the instrument to trigger if a runt pulse is detected with width greater than or less than (but not equal to) the time period specified in TRIGger:{A|B}:RUNT:WIDth within a ±5% tolerance.

Examples

`TRIGger:A:RUNT:WHEn MORETHAN` sets the runt trigger to occur when the instrument detects a runt in a pulse wider than the specified width.

`TRIGger:A:RUNT:WHEn?` might return `:TRIGGER:A:PULSE:RUNT:WHEN OCCURS` indicating that a runt trigger will occur if the instrument detects a runt of any detectable width.

TRIGger:{A|B}:RUNT:WIDth

This command specifies the width, in seconds, for a runt trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:RUNT:WIDth <NR3>
TRIGger:{A|B}:RUNT:WIDth?
```

Related Commands

TRIGger:{A|B}:RUNT:WHEn

Arguments

`<NR3>` is a floating point number that specifies the minimum width, in seconds.

Examples

`TRIGger:A:RUNT:WIDth 15E-6` sets the minimum width of the pulse runt trigger to 15 µs.

`TRIGger:A:RUNT:WIDth?` might return `:TRIGGER:A:PULSE:RUNT:WIDTH 2.0000E-09` indicating that the minimum width of a pulse runt trigger is 2 ns.

TRIGger:{A|B}:SETHold:CLOCk:EDGE

This command specifies the clock edge polarity for setup and hold triggering.

Group

Trigger

Syntax

```
TRIGger:{A|B}:SETHold:CLOCk:EDGE {FALL|RISe}
TRIGger:{A|B}:SETHold:CLOCk:EDGE?
```

Arguments

`FALL` specifies polarity as the clock falling edge.

`RISe` specifies polarity as the clock rising edge.

Examples `TRIGger:A:SETHold:CLOCk:EDGE RISE` specifies the polarity as the clock rising edge.

`TRIGger:A:SETHold:CLOCk:EDGE?` might return `:TRIGGER:A:SETHOLD:CLOCK:EDGE RISE` indicating that polarity is specified as the clock rising edge.

TRIGger:{A|B}:SETHold:CLOCk:SOUrce

This command specifies the clock source for the setup and hold triggering. You cannot specify the same source for both clock and data.

Group Trigger

Syntax

```
TRIGger:{A|B}:SETHold:CLOCk:SOUrce {CH<x>|CH<x>_D<y>}
TRIGger:{A|B}:SETHold:CLOCk:SOUrce?
```

Arguments `CH<x>` specifies the analog channel to use as the clock source waveform.

`CH<x>_D<y>` specifies the digital channel to use as the clock source waveform.

Examples `TRIGger:A:SETHold:CLOCk:SOUrce CH1` specifies channel 1 as the clock source for a setup and hold trigger operation.

`TRIGger:A:SETHold:CLOCk:SOUrce?` might return `:TRIGGER:A:SETHOLD:CLOCK:SOURCE CH4` indicating that channel 4 is the clock source for a setup and hold trigger operation.

TRIGger:{A|B}:SETHold:HOLDTime

This command specifies the hold time for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Hold Time.

Group Trigger

Syntax

```
TRIGger:{A|B}:SETHold:HOLDTime <NR3>
TRIGger:{A|B}:SETHold:HOLDTime?
```

Arguments <NR3> is a floating point number that specifies the hold time setting, in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

Examples TRIGger:A:SETHold:HOLDTime 3.0E-3 sets the hold time for the setup and hold trigger to 3 ms.

TRIGger:A:SETHold:HOLDTime? might return :TRIGGER:A:SETHOLD:HOLDTIME 2.0000E-09 indicating that the current hold time for the setup and hold trigger is 2 ns.

TRIGger:{A|B}:SETHold:SETTime

This command specifies the setup time for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Setup Time.

Group Trigger

Syntax
```
TRIGger:{A|B}:SETHold:SETTime <NR3>
TRIGger:{A|B}:SETHold:SETTime?
```

Arguments <NR3> is a floating point number that specifies the setup time for setup and hold violation triggering.

Examples TRIGger:A:SETHold:SETTime 3.0E-6 specifies that the current setup time for setup and hold trigger is 3 µs.

TRIGger:A:SETHold:SETTime? might return :TRIGGER:A:LOGIC:SETHOLD:SETTIME 2.0000E-09 indicating that the current setup time for setup and hold trigger is 2 ns.

TRIGger:{A|B}:SETHOLDLogicval:{CH<x>|CH<x>_D<x>}

This command sets or queries whether the specified channel is included (INCLude) or not included (DONTInclude) in the Setup & Hold trigger input configuration. The channel number is specified by x.

Group Trigger

Syntax

```
TRIGger:{A|B}:SETHOLDLogicval:{CH<x>|CH<x>_D<x>}
{INCLude|DONTInclude}
TRIGger:{A|B}:SETHOLDLogicval:{CH<x>|CH<x>_D<x>}?
```

Arguments

`INCLude` specifies including the specified channel in the Setup & Hold trigger input configuration.

`DONTInclude` specifies not including the specified channel in the Setup & Hold trigger input configuration.

Examples

`TRIGger:A:SETHOLDLogicval:CH1 INCLUDE` specifies including the specified channel in the Setup & Hold trigger input configuration.

`TRIGger:A:SETHOLDLogicval:CH1?` might include `:TRIGGER:A:SETHOLDLOGICVAL:CH1 DONTINCLUDE` indicating not to include the channel in the configuration.

`TRIGger:A:SETHOLDLogicval:CH1_D0` include specifies including the specified channel in the setup and hold trigger input configuration.

`TRIGger:A:SETHOLDLogicval:CH1_D0?` might return `:TRIGGER:A:SETHOLDLOGICVAL:CH1_D0 DONTINCLUDE` indicating the channel will not be included in the configuration.

TRIGger:{A|B}:TIMEOut:LOGICQUALification

This command specifies whether or not to use logic qualification for a timeout trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TIMEOut:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:TIMEOut:LOGICQUALification?
```

Arguments

`ON` specifies that the timeout trigger type uses logic qualification.

`OFF` specifies that the timeout trigger type does not use logic qualification.

Examples

`TRIGger:A:TIMEOut:LOGICQUALification ON` specifies using logic qualification.

`TRIGger:A:TIMEOut:LOGICQUALification?` might include `:TRIGGER:A:TIMEOUT:LOGICQUALIFICATION OFF` indicating that logic qualification is off.

TRIGger:{A|B}:TIMEOut:POLarity

When triggering using the TIMEOut trigger type, this commands specifies the polarity to be used.

Group Trigger

Syntax

```
TRIGger:{A|B}:TIMEOut:POLarity {STAYSHigh|STAYSLow|EITher}
TRIGger:{A|B}:TIMEOut:POLarity?
```

Related Commands TRIGger:{A|B}:TIMEOut:SOUrce

TRIGger:{A|B}:TIMEOut:TIMe

Arguments STAYSHigh — Trigger when the signal stays high during the timeout time specified by the command TRIGger:{A|B}:TIMEOut:TIMe.

STAYSLow — Trigger when the signal stays low during the timeout time specified by the command TRIGger:{A|B}:TIMEOut:TIMe.

EITher — Trigger when the signal is either high or low during the timeout time specified by the command TRIGger:{A|B}:TIMEOut:TIMe.

Examples `TRIGger:A:TIMEOut:POLarity STAYSHigh` specifies to trigger when the signal stays high during the timeout time when triggering using the TIMEOut trigger type.

`TRIGger:A:TIMEOut:POLarity?` might return EITHER, indicating that the signal stays either high or low during the timeout time.

TRIGger:{A|B}:TIMEOut:SOUrce

When triggering using the TIMEOut trigger type, this command specifies the source. The available sources are live channels and digital channels. The default is channel 1. The timeout trigger type is selected using TRIGger:{A|B}:TYPe TIMEOut.

Group Trigger

Syntax

```
TRIGger:{A|B}:TIMEOut:SOUrce {CH<x>|CH<x>_D<y>}
TRIGger:{A|B}:TIMEOut:SOUrce?
```

Related Commands SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce

TRIGger:{A|B}:TIMEOut:TIMe

TRIGger:{A|B}:TIMEOut:POLarity

Arguments `CH<x>` specifies an analog channel as the timeout trigger source.

`CH<x>_D<y>` specifies a digital channel as the timeout trigger source.

Examples `TRIGger:A:TIMEOut:SOUrce CH4` specifies to use channel 4 as the source for the timeout trigger.

`TRIGger:A:TIMEOut:SOUrce?` might return CH1, indicating that channel 1 has been set as the source for the timeout trigger.

TRIGger:{A|B}:TIMEOut:TIMe

When triggering using the `TIMEOut` trigger type, this command specifies the timeout time, in seconds. This command is equivalent to selecting Timeout from the Trig menu and setting a value for Time Limit. The timeout trigger type is selected using TRIGger:{A|B}:TYPe TIMEOut

Group Trigger

Syntax
```
TRIGger:{A|B}:TIMEOut:TIMe <NR3>
TRIGger:{A|B}:TIMEOut:TIMe?
```

Related Commands TRIGger:{A|B}:TIMEOut:POLarity

TRIGger:{A|B}:TIMEOut:SOUrce

Arguments `<NR3>` is a floating point number that specifies the timeout time, in seconds.

Examples `:TRIGger:A:TIMEOut:TIMe 4.0E-9` specifies the timeout time of 4.0 nsec.

`:TRIGger:A:TIMEOut:TIMe?` might return 8.0000E-9.

TRIGger:{A|B}:TRANsition:DELTatime

This command specifies the delta time (that is Time Limit) used in calculating the transition value for the transition (that is Rise or Fall Time) trigger.

Group Trigger

Syntax

```
TRIGger:{A|B}:TRANsition:DELTatime <NR3>
TRIGger:{A|B}:TRANsition:DELTatime?
```

Arguments

<NR3> is a floating point number that specifies the delta time, in seconds.

Examples

TRIGGER:A:TRANSITION:DELTATIME 15E-6 sets the delta time of the transition trigger to 15 µs.

TRIGGER:A:TRANSITION:DELTATIME? might return :TRIGGER:A:TRANSITION:DELTATIME 2.0000E-09 indicating that the delta time of the transition trigger is set to 2 ns.

TRIGger:{A|B}:TRANsition:LOGICQUALification

This command specifies whether or not to use logic qualification for a transition trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TRANsition:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:TRANsition:LOGICQUALification?
```

Arguments

ON specifies that the transition trigger type uses logic qualification.

OFF specifies that the transition trigger type does not use logic qualification.

Examples

TRIGger:A:TRANsition:LOGICQUALification ON specifies using logic qualification.

TRIGger:A:TRANsition:LOGICQUALification? might return :TRIGGER:A:TRANSITION:LOGICQUALIFICATION OFF indicating logic qualification is off.

TRIGger:{A|B}:TRANsition:POLarity

This command specifies the polarity for the transition trigger.

Group

Trigger

Syntax

```
TRIGger:{A|B}:TRANsition:POLarity {EITher|NEGative|POSitive}
```

TRIGger:{A|B}:TRANsition:POLarity?

Arguments

POSitive indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.

NEGative indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

EITher indicates either positive or negative polarity.

Examples

TRIGGER:A:TRANSITION:POLARITY NEGATIVE sets the transition polarity to negative.

TRIGGER:A:TRANSITION:POLARITY? might return :TRIGGER:A:TRANSITION:POLARITY EITHER indicating that the polarity can be either positive or negative.

TRIGger:{A|B}:TRANsition:SOUrce

This command specifies the source waveform for a transition trigger.

NOTE. *Digital channels are not supported as transition trigger sources.*

Group

Trigger

Syntax

TRIGger:{A|B}:TRANsition:SOUrce {CH<x>}
TRIGger:{A|B}:TRANsition:SOUrce?

Arguments

CH<x> specifies one of the analog channels to be used as the source for a transition trigger.

Examples

TRIGGER:A:TRANSITION:SOURCE CH4 sets channel 4 as the source for the transition trigger.

TRIGGER:A:TRANSITION:SOURCE? might return :TRIGGER:A:TRANSITION:SOURCE CH2 indicating that channel 2 is the source for the transition trigger.

TRIGger:{A|B}:TRANsition:WHEn

This command specifies whether to check for a transitioning signal that is faster or slower than the specified delta time.

Group Trigger

Syntax

```
TRIGger:{A|B}:TRANsition:WHEn {SLOWer|FASTer|EQual|UNEQual}
TRIGger:{A|B}:TRANsition:WHEn?
```

Arguments

`FASTer` sets the trigger to occur when the signal transition time is faster than the time set by `TRIGger:A:TRANsition:DELTatime`.

`SLOWer` sets the trigger to occur when the signal transition time is slower than the time set by.`TRIGger:A:TRANsition:DELTatime`.

`EQual` sets the trigger to occur when the signal transition time is equal to the time set by `TRIGger:A:TRANsition:DELTatime`.

`UNEQual` sets the trigger to occur when the signal transition time is not equal to the time set by `TRIGger:A:TRANsition:DELTatime`.

Examples

`TRIGGER:A:TRANSITION:WHEN SLOWER` sets the trigger to occur when the signal transition time is slower than the time set by the `TRIGger:A:TRANsition:DELTatime` command.

`TRIGGER:A:TRANSITION:WHEN?` might return
`:TRIGGER:A:TRANSITION:WHEN FASTER`

TRIGger:{A|B}:TYPe

This command sets or queries the type of A or B trigger.

Group Trigger

Syntax

```
TRIGger:{A|B}:TYPe {EDGE|WIDth|TIMEOut|RUNt|WINdow|LOGIc|
SETHold|TRANsition|BUS}
TRIGger:{A|B}:TYPe?
```

Arguments

`EDGE` is a normal trigger. A trigger event occurs when a signal passes through a specified voltage level in a specified direction and is controlled by the TRIGger:A:`EDGE` commands.

`WIDth` specifies that the trigger occurs when a pulse with a specified with is found.

`TIMEOut` specifies that a trigger occurs when a pulse with the specified timeout is found.

RUNt specifies that a trigger occurs when a pulse with the specified parameters is found.

WINdow specifies that a trigger occurs when a signal with the specified window parameters is found.

LOGIc specifies that a trigger occurs when specified conditions are met and is controlled by the TRIGger:{A|B}:LOGIc commands.

SETHold specifies that a trigger occurs when a signal is found that meets the setup and hold parameters.

Transition specifies that a trigger occurs when a specified pulse is found that meets the transition trigger parameters.

BUS specifies that a trigger occurs when a signal is found that meets the specified bus setup parameters.

Examples

TRIGGER:A:TYPE EDGE sets the A trigger type to EDGE.

TRIGGER:A:TYPE? might return :TRIGGER:A:TYPE RUNT indicating that the A trigger type is a runt trigger.

TRIGger:{A|B}:UPPerthreshold:CH<x>

This command sets or queries the specified channel upper trigger level. The CH<x> range is 1 to 8 and depends on the number of analog channels on your instrument.

Group

Trigger

Syntax

```
TRIGger:{A|B}:UPPerthreshold:CH<x> <NR3>
TRIGger:{A|B}:UPPerthreshold:CH<x>?
```

Related Commands

TRIGger:{A|B}:LOWerthreshold:CH<x>

Arguments

<NR3> specifies the trigger level in user units (usually volts).

Examples

TRIGGER:A:UPPERTHRESHOLD:CH1 1.3 This command sets the A trigger level for Channel 1 to 1.3 V level.

TRIGGER:A:UPPERTHRESHOLD:CH2? might return TRIGGER:A:UPPERTHRESHOLD:CH2 1.3000E+00, indicating that the A trigger level for Channel 2 is set to 1.3 V.

TRIGger:{A|B}:WINdow:CROSSIng

This command sets or queries the window trigger threshold crossing of the selected trigger source. The threshold crossing selection is only effective when :TRIGger:{A|B}:WINdow:WHEn is INSIDEGreater or OUTSIDEGreater.

Group Trigger

Syntax `TRIGger:{A|B}:WINdow:CROSSIng {UPPer|LOWer|EITher|NONe}`

Arguments

`UPPer` if `:TRIGger:{A|B}:WINdow:WHEn` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then exits through the upper threshold. If `:TRIGger:{A|B}:WINdow:WHEn` is `OUTSIDEGreater`, the instrument triggers when the signal remains above the upper threshold for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then crosses downward through the upper threshold.

`LOWer` if `:TRIGger:{A|B}:WINdow:WHEn` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then exits through the lower threshold. If `:TRIGger:{A|B}:WINdow:WHEn` is `OUTSIDEGreater`, the instrument triggers when the signal remains below the lower threshold for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then crosses upwards through the lower threshold.

`EITher` if `:TRIGger:{A|B}:WINdow:WHEn` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then exits through either the upper or lower threshold. If `:TRIGger:{A|B}:WINdow:WHEn` is `OUTSIDEGreater`, the instrument triggers when the signal remains either above the upper threshold or below the lower threshold for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) and then crosses a threshold.

`NONe` if `:TRIGger:{A|B}:WINdow:WHEn` is `INSIDEGreater`, the instrument triggers when the signal remains between the upper and lower thresholds for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) without crossing through either the upper or lower threshold. If `:TRIGger:{A|B}:WINdow:WHEn` is `OUTSIDEGreater`, the instrument triggers when the signal remains outside the upper and lower thresholds for longer than the time limit (`:TRIGger:{A|B}:WINdow:WIDTH`) without crossing through either the upper or lower threshold.

Examples `TRIGger:A:WINdow:CROSSIng LOWER` sets crossing to lower.

`TRIGger:A:WINdow:CROSSIng?` might return `TRIGGER:A:WINDOW:CROSSING UPPER` indicating the crossing is set to upper.

TRIGger:{A|B}:WINdow:LOGICQUALification

This command specifies whether or not to use logic qualification for a window trigger.

Group Trigger

Syntax
```
TRIGger:{A|B}:WINdow:LOGICQUALification {ON|OFF}
TRIGger:{A|B}:WINdow:LOGICQUALification?
```

Arguments `ON` specifies that the window trigger type uses logic qualification.

`OFF` specifies that the window trigger type does not use logic qualification.

Examples `TRIGger:A:WINdow:LOGICQUALification ON` turns on logic qualification.

`TRIGger:A:WINdow:LOGICQUALification?` might return `TRIGGER:A:WINDOW:LOGICQUALIFICATION OFF` indicating logic qualification is off.

TRIGger:{A|B}:WINdow:SOUrce

This command sets or queries the source for a window trigger.

NOTE. *Digital channels are not supported as window trigger sources.*

Group Trigger

Syntax
```
TRIGger:{A|B}:WINdow:SOUrce {CH<x>}
TRIGger:{A|B}:WINdow:SOUrce?
```

Arguments The window trigger source channel.

Examples `TRIGger:A:WINdow:SOUrce CH2` sets the source to channel 2.

`TRIGger:A:WINdow:SOUrce?` might return `:TRIGGER:A:WINDOW:SOURCE CH3` indicating the source is channel 3.

TRIGger:{A|B}:WINdow:WHEn

This command sets or queries the window trigger event. This command is equivalent to selecting Window Setup from the Trig menu and selecting from the Window Trigger When box.

Group Trigger

Syntax
```
TRIGger:{A|B}:WINdow:WHEn {ENTERSWindow|EXITSWindow|
INSIDEGreater|OUTSIDEGreater}
TRIGger:{A|B}:WINdow:WHEn?
```

Arguments `OUTSIDEGreater` specifies a trigger event when the signal leaves the window defined by the threshold levels for the time specified by Width.

`INSIDEGreater` specifies a trigger event when the signal enters the window defined by the threshold levels for the time specified by Width.

`ENTERSWindow` specifies a trigger event when the signal enters the window defined by the threshold levels.

`EXITSWindow` specifies a trigger event when the signal leaves the window defined by the threshold levels.

Examples `TRIGger:A:WINdow:WHEn EXITSWINDOW` specifies a trigger event when the signal leaves the window defined by the threshold levels.

`TRIGger:A:WINdow:WHEn?` might return `:TRIGGER:A:WINDOW:WHEN ENTERSWINDOW` indicating a trigger when the signal enters the window.

TRIGger:{A|B}:WINdow:WIDth

This command sets or queries the minimum width for a window violation. This command is equivalent to selecting Window Setup from the Trig menu, selecting Inside > Limit or Outside > Limit in the Trigger When box, and setting the Time Limit.

Group Trigger

Syntax `TRIGger:{A|B}:WINdow:WIDth <NR3>`

Arguments `<NR3>` is the minimum width for a window violation.

Examples `TRIGger:A:WINdow:WIDTH 4.5e-9` sets the minimum width for a window violation to 4.5 ns.

`TRIGger:A:WINdow:WIDTH?` might return `:TRIGGER:A:WINDOW:WIDTH 4.0E-9` indicating that 4.0 ns is the minimum width for a window violation.

TRIGger:A:HOLDoff:BY

This command sets or queries the type of holdoff for the A trigger. Holdoff types are expressed as either user-specified time (TIMe) or by an internally calculated random time value (RANDom). This command is equivalent to selecting Mode & Holdoff from the Trig menu and then setting the Holdoff type.

Group Trigger

Syntax
```
TRIGger:A:HOLDoff:BY {TIMe|RANDom}
TRIGger:A:HOLDoff:BY?
```

Related Commands TRIGger:A:HOLDoff:TIMe

Arguments `TIMe` enables you to set the holdoff time via the TRIGger:A:HOLDoff:TIMe command.

`RANDom` specifies a random time value.

Examples `TRIGGER:A:HOLDOFF:BY TIME` sets the holdoff to the "by time" setting. This enables you to set the holdoff time.

`TRIGGER:A:HOLDOFF:BY?` might return `:TRIGGER:A:HOLDOFF:BY TIME`, indicating that you will set the holdoff time.

TRIGger:A:HOLDoff:TIMe

This command sets or queries the A trigger holdoff time. This command is equivalent to selecting Mode & Holdoff from the Trig menu, selecting Time, and then setting the desired Holdoff Time.

Group Trigger

Syntax
```
TRIGger:A:HOLDoff:TIMe <NR3>
TRIGger:A:HOLDoff:TIMe?
```

Related Commands TRIGger:A:HOLDoff:BY

Arguments <NR3> specifies the holdoff time in seconds. The range is from 0 seconds through 10 seconds.

Examples TRIGGER:A:HOLDOFF:TIME 10 sets the A trigger holdoff time to 10 s.

TRIGGER:A:HOLDOFF:TIME? might return :TRIGGER:A:HOLDOFFTIME 1.2000E-06, indicating that the A trigger holdoff time is set to 1.2 µs.

TRIGger:A:LOGICQUALification

This command sets or queries the type of logic qualification to perform.

Group Trigger

Syntax
```
TRIGger:A:LOGICQUALification {AND|OR|NAND|NOR}
TRIGger:A:LOGICQUALification?
```

Arguments AND specifies to trigger if all conditions are true.

NANd specifies to trigger if any of the conditions are false.

NOR specifies to trigger if all conditions are false.

OR specifies to trigger if any of the conditions are true.

Examples TRIGGER:A:LOGICQUALIFICATION OR specifies to trigger if any of the conditions are true.

TRIGGER:A:LOGICQUALIFICATION? might return TRIGGER:A:LOGICQUALIFICATION AND indicating the instrument will trigger if all conditions are true.

TRIGger:A:MODe

This command sets or queries the A trigger mode. This command is equivalent to pushing the Mode button on the front panel.

Group Trigger

Syntax

```
TRIGger:A:MODe {AUTO|NORMal}
TRIGger:A:MODe?
```

Arguments `AUTO` generates a trigger if one is not detected within a specified time period.

`NORMal` waits for a valid trigger event.

Examples `TRIGGER:A:MODE NORMAL` sets the trigger mode to normal.

`TRIGGER:A:MODE?` might return `TRIGGER:A:MODE AUTO`, indicating that the trigger mode is auto.

TRIGger:AUXLevel

For those instruments that have an Auxiliary Input (such as an MSO58LP), this command sets or queries the Auxiliary Input voltage level to use for an edge trigger.

Group Trigger

Syntax

```
TRIGger:AUXLevel {<NR3>|ECL|TTL}
TRIGger:AUXLevel?
```

Arguments `<NR3>` is trigger level in Volts.

`ECL` sets trigger level to -1.3 Volts.

`TTL` sets trigger level to 1.4 Volts.

Returns `<NR3>` is trigger level in Volts.

Examples `TRIGger:AUXLevel 1.2` sets trigger level for the Auxiliary Input to 1.2 Volts.

TRIGger:B:BY

This command selects or returns whether the B trigger occurs after a specified number of events or a specified period of time after the A trigger.

Group Trigger

Syntax

```
TRIGger:B:BY {EVENTS|TIMe}
TRIGger:B:BY?
```

Related Commands TRIGger:B:EVENTS:COUNt

TRIGger:B:TIMe

TRIGger:B:STATE

Arguments `EVENTS` sets the B trigger to take place following a set number of trigger events after the A trigger occurs. The number of events is specified by `TRIGger:B:EVENTS:COUNt`.

`TIMe` sets the B trigger to occur a set time after the A trigger event. The time period is specified by `TRIGger:B:TIMe`.

Examples `TRIGGER:B:BY TIME` sets the B trigger to occur at a set time after the A trigger event.

`TRIGGER:B:BY?` might return `TRIGGER:B:BY EVENTS`, indicating that the B trigger takes place following a set number of trigger events after the A trigger occurs.

TRIGger:B:EVENTS:COUNt

This command sets or queries the number of events that must occur before the B trigger. The B trigger event count applies only if TRIGger:B:BY is set to EVENTS.

Group Trigger

Syntax

```
TRIGger:B:EVENTS:COUNt <NR1>
TRIGger:B:EVENTS:COUNt?
```

Related Commands TRIGger:B:STATE

Arguments <NR1> is the number of B trigger events, which can range from 1 to 65,471.

Examples TRIGGER:B:EVENTS:COUNT 4 sets the number of B trigger events to four.

TRIGGER:B:EVENTS:COUNT? might return TRIGGER:B:EVENTS:COUNT 2, indicating that two events must occur after the A trigger before the B trigger can occur.

TRIGger:B:RESET (No Query Form)

This command sets the B reset trigger level to 50%.

Group Trigger

Syntax
```
TRIGger:B:RESET SETLevel
```

Arguments SETLevel sets the B reset trigger level to 50%.

Examples TRIGger:B:RESET SETLevel sets the B reset trigger level to 50%.

TRIGger:B:RESET:EDGE:COUPling

Sets or queries the trigger coupling for a sequential edge trigger reset when the Source is set to an analog channel.

Group Trigger

Syntax
```
TRIGger:B:RESET:EDGE:COUPling {DC|HFRej|LFRej|NOISErej}
TRIGger:B:RESET:EDGE:COUPling?
```

Related Commands

Arguments DC selects DC trigger coupling.

HFRej selects high frequency low sensitivity.

LFRej selects low frequency low sensitivity.

NOISErej selects DC low sensitivity.

Examples `TRIGGER:B:RESET:EDGE:COUPLING` HFREJ sets high frequency low sensitivity.

`TRIGGER:B:RESET:EDGE:COUPLING`? might return TRIGGER:B:RESET:EDGE:COUPLING DC indicating DC trigger coupling is selected.

TRIGger:B:RESET:EDGE:LEVel

This command sets the voltage level to use for an Edge Reset trigger when triggering on an analog channel waveform.

Group Trigger

Syntax

```
TRIGger:B:RESET:EDGE:LEVel <NR3>
```

Arguments `<NR3>` is the voltage level to use for an Edge Reset trigger when triggering on an analog channel waveform.

Examples `TRIGGER:B:RESET:EDGE:LEVEL` 50.0e-3 sets the level to 50.0 mV.

`TRIGGER:B:RESET:EDGE:LEVEL`? might return TRIGGER:B:RESET:EDGE:LEVEL 0.0E+0 indicating the level is set to 0.0 V.

TRIGger:B:RESET:EDGE:SLOpe

This command sets or queries the trigger slope for a sequential edge trigger reset.

Group Trigger

Syntax

```
TRIGger:B:RESET:EDGE:SLOpe {RISe|FALL|EIther}
TRIGger:B:RESET:EDGE:SLOpe?
```

Arguments `RISe` specifies to reset the trigger on the rising or positive edge of a signal.

`FALL` specifies to reset the trigger on the falling or negative edge of a signal.

`EITHER` specified to reset the trigger on either the rising or falling edge of a signal.

Examples `TRIGGER:B:RESET:EDGE:SLOPE` FALL specifies to reset the trigger on the falling or negative edge of a signal.

TRIGGER:B:RESET:EDGE:SLOPE? might return TRIGGER:B:RESET:EDGE:SLOPE RISE indicating the instrument is set to reset the trigger on the rising or positive edge of a signal.

TRIGger:B:RESET:EDGE:SOUrce

This command sets or queries the trigger source for the A→B sequential edge trigger reset feature.

Group Trigger

Syntax

```
TRIGger:B:RESET:EDGE:SOUrce {CH<x>|CH<x>_D<y>}
TRIGger:B:RESET:EDGE:SOUrce?
```

Arguments The source channel for the trigger reset.

Examples TRIGGER:B:RESET:EDGE:SOURCE CH4 sets Channel 4 as the input source for the trigger reset.

TRIGGER:B:RESET:EDGE:SOURCE? might return TRIGGER:B:RESET:EDGE:SOURCE CH1, indicating that the current input source for the trigger reset is Channel 1.

TRIGger:B:RESET:TIMEOut:TIMe

This command sets or queries the reset timer for a sequential timeout trigger reset.

Group Trigger

Syntax

```
TRIGger:B:RESET:TIMEOut:TIMe <NR3>
TRIGger:B:RESET:TIMEOut:TIMe?
```

Arguments <NR3> is the reset timer for a sequential timeout trigger reset.

Examples TRIGGER:B:RESET:TIMEOUT:TIME 100e-9 sets the time to 100 ns.

TRIGGER:B:RESET:TIMEOUT:TIME? might return TRIGGER:B:RESET:TIMEOUT:TIME 20.0E-9 indicating the timeout time is set to 20 ns.

TRIGger:B:RESET:TYPe

This command sets or queries the type of A→B sequential trigger reset. If the B trigger reset is active, the reset criteria are part of the B triggering sequence. If the reset conditions defined by the reset type are met, the instrument must start over searching for a new occurrence of the A event.

You must identify a trigger Source and Threshold for each reset type, except for the Timeout trigger type.

NOTE. *If a reset condition occurs, the reset criteria itself is reset and must start over.*

Group Trigger

Syntax

```
TRIGger:B:RESET:TYPe {NONe|TIMEOut|EDGE}
TRIGger:B:RESET:TYPe?
```

Arguments

`NONe` defeats the trigger reset feature.

`TIMEOut` initiates a reset if the timeout conditions specified by TRIGger:B:RESET:TIMEOut are met.

`EDGE` initiates a reset if the edge trigger conditions are met.

Examples

`TRIGGER:B:RESET:TYPE NONE` deactivates the sequential trigger reset.

`TRIGGER:B:RESET:TYPE?` This query might return `TRIGGER:B:RESET:TYPe TIMEOUT`, indicating that the sequential trigger reset is active following a timeout.

TRIGger:B:STATE

This command sets or queries the state of B trigger activity. If the B trigger state is on, the B trigger is part of the triggering sequence. If the B trigger state is off, then only the A trigger causes the trigger event.

Group Trigger

Syntax

```
TRIGger:B:STATE {ON|OFF|<NR1>}
TRIGger:B:STATE?
```

Related Commands TRIGger:A:MODe

Arguments

`ON` indicates that the B trigger is active and causes trigger events with the A trigger.

`OFF` indicates that only the A trigger causes trigger events.

`<NR1>` is an integer number. 0 turns off the B trigger; any other value activates the B trigger.

Examples

`TRIGGER:B:STATE ON` sets the B trigger to active, making it capable of causing trigger events.

`TRIGGER:B:STATE?` might return `TRIGGER:B:STATE 0`, indicating that the B trigger is inactive and that only the A trigger causes trigger events.

TRIGger:B:TIMe

This command sets or queries B trigger delay time, in seconds. The B Trigger time applies only if TRIGger:B:BY is set to TIMe.

Group

Trigger

Syntax

```
TRIGger:B:TIMe <NR3>
TRIGger:B:TIMe?
```

Related Commands

TRIGger:B:BY

TRIGger:B:EVENTS:COUNt

TRIGger:B:STATE

Arguments

`<NR3>` is the B trigger delay time in seconds.

Examples

`TRIGGER:B:TIME 4E-6` sets the B trigger delay time to 4 µs.

`TRIGGER:B:TIME?` might return `TRIGGER:B:TIME 16.0000E-9`, indicating that the B trigger time is set to 16 ns.

TRIGger:STATE? (Query Only)

This query-only command returns the current state of the triggering system.

Group

Trigger

Syntax `TRIGger:STATE?`

Related Commands TRIGger:A:MODe

Returns `ARMED` indicates that the instrument is acquiring pretrigger information.

`AUTO` indicates that the instrument is in the automatic mode and acquires data even in the absence of a trigger.

`READY` indicates that all pretrigger information is acquired and that the instrument is ready to accept a trigger.

`SAVE` indicates that the instrument is in save mode and is not acquiring data.

`TRIGGER` indicates that the instrument triggered and is acquiring the post trigger information.

Examples `TRIGGER:STATE?` might return `TRIGGER:STATE ARMED`, indicating that the pretrigger data is being acquired.

*TST? (Query Only)

Tests (self-test) the interface and returns a 0.

Group Miscellaneous

Syntax `*TST?`

Examples `*TST?` always returns 0.

UNDO (No Query Form)

Reverts the instrument settings to a state before the previous command or user interface action.

Group Miscellaneous

Syntax `UNDO`

Examples UNDO reverts the instrument settings to a state before the previous command or user interface action.

UNLock (No Query Form)

This command (no query form) unlocks the front panel controls only. To unlock the front panel controls and the touch screen use the LOCk NONe command. The command TOUCHSCReen:STATE ON enables the touch screen only.

NOTE. *If the instrument is in the Remote With Lockout State (RWLS), the UNLock command has no effect. For more information, see the ANSI-IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.*

Group Miscellaneous

Syntax
```
UNLock ALL
```

Related Commands LOCk

Arguments ALL specifies that all front panel buttons and knobs are unlocked.

Examples UNLOCK ALL unlocks all front panel buttons and knobs.

USBDevice:CONFigure

This command may be used to configure the rear USB port to be off or enabled as a USBTMC device. Users should be cautious using this command via the USBTMC interface as a change to the configuration of this interface from a USBTMC device will cause USBTMC communication to cease. It is intended to be used via the Ethernet interface to control the USB device interface.

Group Miscellaneous

Syntax
```
USBDevice:CONFigure {DISabled|USBTmc}
USBDevice:CONFigure?
```

Arguments

DISabled will disable the rear USB port.

USBTmc enables the rear USB port.

Examples

USBDEVICE:CONFIGURE DISABLED will disable the rear USB port.

USBDEVICE:CONFIGURE? might return :USBDEVICE:CONFIGURE USBTMC indicating the USB port is enabled.

VERBose

This command sets or queries the Verbose state that controls the length of keywords on query responses. Keywords can be both headers and arguments.

NOTE. *This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk).*

Group

Miscellaneous

Syntax

VERBose {<NR1>|OFF|ON}

Related Commands

HEADer

*LRN?

SET?

Arguments

<NR1> = 0 disables Verbose, any other value enables Verbose.

OFF sets the Verbose state to false, which returns minimum-length keywords for applicable setting queries.

ON sets the Verbose state to true, which returns full-length keywords for applicable setting queries.

A 0 returns minimum-length keywords for applicable setting queries; any other value returns full-length keywords.

Examples

VERBOSE ON sets the Verbose state to true and return the full length keyword for the applicable setting queries.

VERBOSE? might return :VERBOSE OFF, indicating that the Verbose state is set to false and return the minimum-length keywords for the applicable setting queries.

With :HEADer ON and :VERBose ON, the :ACQuire:MODe? query might return: :ACQUIRE:MODE SAMPLE

With :HEADer ON and :VERBose OFF, the :ACQuire:MODe? query might return: :ACQ:MOD SAM

With :HEADer OFF and :VERBose ON, the :ACQuire:MODe? query might return: SAMPLE

With :HEADer OFF and :VERBose OFF, the :ACQuire:MODe? query might return: SAM

VERTical:DESKew:FROM:CUSTOMPROPAgation

This command sets or queries a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.

Group Vertical

Syntax
```
VERTical:DESKew:FROM:CUSTOMPROPAgation <NR3>
VERTical:DESKew:FROM:CUSTOMPROPAgation?
```

Arguments <NR3> is a target (FROM) delay that you can specify when the propagation delay of the target (FROM) probe used for deskew cannot be detected automatically.

Examples VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION 0.25e-9 sets the custom delay to 250 ps.

VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION? might return :VERTICAL:DESKEW:FROM:CUSTOMPROPAGATION 0.0E+0 indicating the custom delay is 0.0 s.

VERTical:DESKEW:FROM:SOUrce

This command sets or queries the source channel for performing channel-to-channel deskew adjustment. Sources can be any of the analog channels.

Group Vertical

Syntax
```
VERTical:DESKEW:FROM:SOUrce CH<x>
VERTical:DESKEW:FROM:SOUrce?
```

Arguments The vertical deskew source.

Examples `VERTICAL:DESKEW:FROM:SOURCE CH1` sets channel 1 as the from source.

`VERTICAL:DESKEW:FROM:SOURCE?` might return `:VERTICAL:DESKEW:FROM:SOURCE CH2` indicating channel 2 is the from source.

VERTical:DESKEW:STATIC (No Query Form)

This command executes static deskew using the deskew settings.

Group Vertical

Syntax `VERTical:DESKEW:STATIC EXECute`

Arguments `EXECute` will execute static deskew using the deskew settings.

Examples `VERTICAL:DESKEW:STATIC EXECUTE` will execute static deskew using the deskew settings.

VERTical:DESKew:TO:CUSTOMPROPAgation

This command sets or queries a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.

Group Vertical

Syntax `VERTical:DESKew:TO:CUSTOMPROPAgation <NR3>`

Arguments `<NR3>` is a target (TO) delay that can be specified by the user when the propagation delay of the target (TO) probe used for deskew cannot be detected automatically.

Examples `VERTICAL:DESKEW:TO:CUSTOMPROPAGATION 0.25e-9` sets the custom propagation to 250 ps.

VERTICAL:DESKEW:TO:CUSTOMPROPAGATION? might return :VERTICAL:DESKEW:TO:CUSTOMPROPAGATION 0.0E+0 indicating the custom propagation is set to 0.0 ns.

VERTical:DESKEW:TO:SOUrce

This command sets or queries the target channel for performing channel-to-channel deskew adjustment. Target sources can be any of the live analog channels.

Group Vertical

Syntax

```
VERTical:DESKEW:TO:SOUrce CH<x>
VERTical:DESKEW:TO:SOUrce?
```

Arguments Arguments are the live analog channels.

Examples VERTICAL:DESKEW:TO:SOURCE CH4 sets the deskew to source is channel 4.

VERTICAL:DESKEW:TO:SOURCE? might return :VERTICAL:DESKEW:TO:SOURCE CH2 indicating the deskew to source is channel 2.

VISual:AREA<x>:ASPEctratio

Sets or queries whether the aspect ratio of the specified Visual Trigger area is locked.

Group Trigger

Syntax

```
VISual:AREA<x>:ASPEctratio {ON|OFF|<NR1>}
VISual:AREA<x>:ASPEctratio?
```

Arguments Area<x> specifies the integer number of a Visual Trigger area.

ON locks the aspect ratio of the specified Visual Trigger area. The aspect ratio is kept constant when the height or width of the area changes.

OFF unlocks the aspect ratio of the specified Visual Trigger area.

<NR1> is an integer number. 1 locks the aspect ratio of the specified Visual Trigger area; any other value unlocks the aspect ratio of the specified Visual Trigger area

Examples `VISual:AREA6:ASPEctratio ON` locks the aspect ratio of area 6.

`VISual:AREA2:ASPEctratio?` might return `VISual:AREA2:ASPEctratio OFF`,. indicating that the aspect ratio of area 2 is not locked.

VISual:AREA<x>:FLIP:HORizontal (No Query Form)

Flips the specified Visual Trigger area horizontally around its center point.

Group Trigger

Syntax `VISual:AREA<x>:FLIP:HORizontal`

Related Commands VISual:AREA<x>:FLIP:VERTical

Arguments `Area<x>` specifies the integer number of a Visual Trigger area.

Examples `VISual:AREA1:FLIP:HORizontal` horizontally flips area 1.

VISual:AREA<x>:FLIP:VERTical (No Query Form)

Flips the specified Visual Trigger area vertically around its center point.

Group Trigger

Syntax `VISual:AREA<x>:FLIP:VERTical`

Related Commands VISual:AREA<x>:FLIP:HORizontal

Arguments `Area<x>` specifies the integer number of a Visual Trigger area.

Examples `VISual:AREA8:FLIP:VERTical` vertically flips area 8.

VISual:AREA<x>:HEIGht

Sets or queries the height of the specified Visual Trigger area.

Group Trigger

Syntax
```
VISual:AREA<x>:HEIGht <NR3>
VISual:AREA<x>:HEIGht?
```

Arguments `Area<x>` specifies the integer number of a Visual Trigger area.

`<NR3>` specifies the height of the Visual Trigger area in amplitude.

Examples `VISual:AREA51:HEIGht 3.5` sets the height of area 51 to 3.5 (volts or Amps).

`VISual:AREA4:HEIGht?` might return `VISual:AREA4:HEIGht 15`, indicating that the height of area 4 is 15.

VISual:AREA<x>:HITType

Sets or queries the area hit logic true condition for the specified Visual Trigger area.

Group Trigger

Syntax
```
VISual:AREA<x>:HITType {IN|OUT|DONTcare}
VISual:AREA<x>:HITType?
```

Arguments `Area<x>` specifies the integer number of a Visual Trigger area.

`IN` specifies that the waveform must intrude anywhere into the specified area to be true.

`OUT` specifies that the waveform must not intrude anywhere into the specified area to be true.

`DONTcare` sets the area to a don't care state, causing the area to be ignored. This is useful when you are developing a Visual Trigger condition and need to change the hit logic type of an area while keeping the area.

Examples `VISual:AREA2:HITType OUT` sets the area 2 hit type to OUT.

`VISual:AREA5:HITType?` might return `VISual:AREA5:HITType IN`, indicating that the waveform must intrude into area 5 to be true.

VISual:AREA<x>:RESET (No Query Form)

Sets the specified Visual Trigger area shape to a default-sized triangle.

Group Trigger

Syntax
```
VISual:AREA<x>:RESET
```

Arguments Area<x> specifies the integer number of a Visual Trigger area.

Examples VISual:AREA2:RESET changes area 2 to a default triangle shape.

VISual:AREA<x>:ROTAtion

Sets or queries the rotation angle of the specified Visual Trigger area.

Group Trigger

Syntax
```
VISual:AREA<x>:ROTAtion <NR3>
VISual:AREA<x>:ROTAtion?
```

Arguments Area<x> specifies the integer number of a Visual Trigger area.

<NR3> specifies the rotation angle of the Visual Trigger area, in positive degrees from 0 to 360. Zero degrees is referenced from when the area was created.

Examples VISual:AREA2:ROTAtion 45 rotates Visual Trigger area 2 by 45 degrees.

VISual:AREA1:ROTAtion? might return VISual:AREA1:ROTAtion -60. indicating that area 4 rotation position is minus 60 degrees.

VISual:AREA<x>:SHAPE

Sets or queries the current shape of the area.

Group Trigger

Syntax
```
VISual:AREA<x>:SHAPE {TRIAngle|RECTangle|TRAPezoid|HEXAgon}
VISual:AREA<x>:SHAPE?
```

Arguments `Area<x>` specifies the integer number of a Visual Trigger area.

`TRIAngle` sets the specified area to a triangular shape (three vertices). If the area does not exist, the instrument creates a new triangular area with the specified area number.

`RECTangle` sets the specified area to a rectangular shape (four vertices, right angles at each corner). If the area does not exist, the instrument creates a new triangular area with the specified area number.

`TRAPezoid` sets the specified area to a trapezoidal shape (four vertices, two parallel sides). If the area does not exist, the instrument creates a new triangular area with the specified area number.

`HEXAgon` sets the specified area to a hexagonal shape (six vertices). If the area does not exist, the instrument creates a new hexagonal area with the specified area number.

Returns `CUSTom` indicates that the shape is a custom-created shape, or the positions of one or more vertices of a standard shape have been changed.

Examples `VISual:AREA1:SHAPE Hexagon` sets area 1 to be a hexagonal shape, or creates a new area 1 hexagonal shape if the specified area does not exist.

`VISual:AREA2:SHAPE?` might return `VISual:AREA2:SHAPE CUSTOM`, indicating that area 2 is a custom shape.

VISual:AREA<x>:SOUrce

Sets or queries the signal source for the specified Visual Trigger area. The source can only be an analog channel.

Group Trigger

Syntax
```
VISual:AREA<x>:SOUrce {CH1|CH2|CH3|CH4|CH5|CH6|CH7|CH8}
VISual:AREA<x>:SOUrce?
```

Arguments `Area<x>` specifies the integer number of a Visual Trigger area.

`CH1` through `CH8` sets the source channel number for the specified area.

Examples `VISual:AREA1:SOUrce CH3` sets the source channel number for area 1 to Channel 3.

VISual:AREA6:SOUrce? might return VISual:AREA6:SOUrce CH8, indicating that the source for area 6 is Channel 8.

VISual:AREA<x>:VERTICES

Sets or queries the X and Y vertex coordinate values for all vertices of the specified Visual Trigger area. Vertex values must be set in pairs.

Group Trigger

Syntax

```
VISual:AREA<x>:VERTICES “<NR3>, <NR3>, <NR3>, <NR3>, <NR3>,
<NR3> [,<NR3>, <NR3>, ...]”
VISual:AREA<x>:VERTICES?
```

Arguments Area<x> specifies the integer number of a Visual Trigger area.

<NR3>, <NR3> specifies the X, Y coordinate pair of each vertex in an area. The first value is X (time) and the second value is Y (amplitude). There must be a minimum of three vertex pairs to create an area. If the specified area exists, the area is changed to the shape specified in the arguments. If the specified area does not exist, a new area is created and assigned the specified vertices.

Examples VISual:AREA3:VERTICES "400E-9,1,400E-9,0.5,600E-9,0.5,600E-9,1" sets or creates area 3 as a rectangle that is 200 ns wide by 500 mV high.

VISual:AREA3:VERTICES? might return VISual:AREA3:VERTICES "171.1E-9,-141.7E-3,321.1E-9,-141.7E-3,283.6E-9,1.708, 208.6E-9,1.708", indicating that area 3 is defined as a trapezoid that is 150 ns wide by 1.85 V high.

VISual:AREA<x>:WIDTH

Sets or queries the width of the specified Visual Trigger area.

Group Trigger

Syntax

```
VISual:AREA<x>:WIDTH <NR3>
VISual:AREA<x>:WIDTH?
```

Arguments `Area<x>` specifies the integer number of a Visual Trigger area.

`<NR3>` specifies the width of the Visual Trigger area in seconds.

Examples `VISual:AREA5:WIDTH 0.0000045` sets the width of area 5 to 4.5 µs.

`VISual:AREA3:WIDTH?` might return `VISual:AREA3:WIDTH 0.000016`, indicating that the width of area 3 is 16 µs.

VISual:AREA<x>:XPOSition

Sets or queries the horizontal (X-axis) center position of the specified Visual Trigger area.

Group Trigger

Syntax
```
VISual:AREA<x>:XPOSition <NR3>
VISual:AREA<x>:XPOSition?
```

Arguments `Area<x>` specifies the integer number of a Visual Trigger area.

`<NR3>` specifies the horizontal position of the center of the Visual Trigger area, in seconds.

Examples `VISual:AREA1:XPOSition -6.9e-6` sets the horizontal center of area 1 to be –6.9 µs from the trigger point.

`VISual:AREA3:XPOSition?` might return `VISual:AREA3:XPOSition 8.0e-6`, indicating that the center of area 3 is 8 µs from the trigger point.

VISual:AREA<x>:YPOSition

Sets or queries the vertical (Y-axis) center position of the specified Visual Trigger area.

Group Trigger

Syntax
```
VISual:AREA<x>:YPOSition <NR3>
VISual:AREA<x>:YPOSition?
```

Arguments `Area<x>` specifies the integer number of a Visual Trigger area.

`<NR3>` specifies the vertical position of the center of the Visual Trigger area, in amplitude (volts, amps).

Examples `VISual:AREA1:YPOSition 0.5` sets the vertical center of area 1 to be 500 mV.

`VISual:AREA3:YPOSition?` might return `VISual:AREA3:YPOSition 2`, indicating that the vertical center of area 3 is 2 volts.

VISual:DELETEALL (No Query Form)

Deletes all Visual Trigger areas.

Group Trigger

Syntax `VISual:DELETEALL`

Examples `VISUAL:DELETEALL` deletes all Visual Trigger areas.

VISual:ENAble

Sets or queries the status (on or off) of the Visual Trigger mode.

Group Trigger

Syntax
```
VISual:ENAble {ON|OFF|<NR1>}
VISual:ENAble?
```

Arguments `ON` enables the Visual Trigger mode.

`OFF` disables the Visual Trigger mode.

`<NR1>` is an integer number. 0 turns off the Visual Trigger mode; any other value enables Visual Trigger mode.

Examples `VISual:ENAble ON` enables the Visual Trigger function.

`VISual:ENAble?` might return `VISual:ENAble OFF`, indicating that Visual Trigger mode is not enabled.

VISual:EQUation

Sets or queries the Visual Trigger area combination logic equation.

Group Trigger

Syntax

```
VISual:EQUation <Qstring>
VISual:EQUation?
```

Arguments <Qstring> defines the Visual Trigger area combination logic equation. The equation is made up of area names (A<x>) combined with logic operators AND, OR, or XOR (&, |, ^). It may also contain parentheses for grouping. The equation must be true to have a valid Visual Trigger event and display a waveform. Each area's true state depends on the area's condition setting (In, Out or Don't Care).

Examples VISual:EQUation "(A1 & A2) | A3" sets the combined area logic such that both areas 1 and 2 must be true, or area 3 must be true, to have a valid Visual Trigger event and display a waveform.

VISual:EQUation? might return VISual:EQUation "A1 & A2 & A3 & A4", indicating that the area combination logic equation requires that areas 1 through 4 must be true to have a valid Visual Trigger event.

VISual:SHOWAReas

Shows or hides all Visual Trigger areas.

Group Trigger

Syntax

```
VISual:SHOWAReas {ON|OFF|<NR1>}
VISual:SHOWAReas?
```

Arguments ON shows all Visual Trigger areas.

OFF hides all Visual Trigger areas.

<NR1> is an integer number. 0 hides all Visual Trigger areas; any other value shows all Visual Trigger areas.

Examples VISual:SHOWAReas OFF hides all Visual Trigger areas.

VISual:SHOWAReas? might return VISual:SHOWAReas ON, indicating that Visual Trigger areas are displayed on the screen.

VISual:SHOWCRiteria

Sets or queries display of the area names and hit criteria for all visual trigger areas.

Group Trigger

Syntax

```
VISual:SHOWCRiteria {ON|OFF|<NR1>}
VISual:SHOWCRiteria?
```

Arguments ON enables display of the area name and hit criteria labels (In, Out, Don't care icons) of all Visual Trigger areas.

OFF hides the area name and hit criteria labels (In, Out, Don't care icons) of all Visual Trigger areas.

<NR1> is an integer number. 0 hides the area name and hit criteria of all Visual Trigger areas; any other value enables displaying the area name and hit criteria of all Visual Trigger areas.

Examples VISual:SHOWCRiteria OFF hides the name and hit criteria labels of all Visual Trigger areas.

VISual:SHOWCRiteria? might return VISual:SHOWCRiteria ON, indicating that the name and hit criteria labels of all Visual Trigger areas are displayed.

VISual:SHOWEQuation

Shows or hides the Visual Trigger area combination logic equation.

Group Trigger

Syntax

```
VISual:SHOWEQuation {ON|OFF|<NR1>}
VISual:SHOWEQuation?
```

Arguments ON shows the Visual Trigger area combination logic equation.

OFF hides the Visual Trigger area combination logic equation.

<NR1> is an integer number. 0 hides the area combination logic equation; any other value displays the area combination logic equation.

Examples VISual:SHOWEQuation ON shows the Visual Trigger area combination logic equation.

`VISual:SHOWEQuation?` might return `VISual:SHOWEQuation OFF`, indicating that the Visual Trigger area combination logic equation is not displayed on the screen.

*WAI (No Query Form)

The *WAI (Wait) command (no query form) prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete. This command allows you to synchronize the operation of the instrument with your application program. For more information, refer to Synchronization Methods.

Group Status and Error

Syntax `*WAI`

Related Commands BUSY?

*OPC

Examples `*WAI` prevents the instrument from executing any further commands or queries until all pending commands that generate an OPC message are complete.

WAVFrm? (Query Only)

This query-only command provides the Tektronix standard waveform query which returns the waveform preamble followed by the waveform data for the source specified by :DATa:SOUrce using the :DATa settings for encoding, width, and so forth.

Group Waveform Transfer

Syntax `WAVFrm?`

Related Commands CURVe

DATa:SOUrce

WFMOutpre?

Examples WAVFRM? might return the waveform data as: WFMOUTPRE:BIT_NR 8;BN_FMT RI;BYT_NR 1; BYT_OR MSB;ENCDG ASC;NR_PT 500;PT_FMT Y; PT_ORDER LINEAR;PT_OFF 0;XINCR 400.0000E-12; XZERO 0.0000;XUNIT "s";YMULT 4.0000E-3; YOFF 0.0000;YZERO 0.0000;YUNIT "V"; WFID "Ch1,DC coupling, 100.0mV/div, 200.0ns/div, 5000 points,Sampl mode".

WFMOutpre? (Query Only)

This query-only command queries the waveform formatting data for the waveform specified by the DATa:SOUrce command. The preamble components are considered to be of two types; formatting and interpretation. The formatting components are: ENCdg, BN_Fmt, BYT_Or, BYT_Nr, BIT_Nr. The interpretation components are derived from the DATa:SOUrce specified waveform.

Group Waveform Transfer

Syntax WFMOutpre?

Examples WFMOUTPRE? might return the waveform formatting data as: WFMOUTPRE:BYT_NR 2;BIT_NR 16;ENCDG BINARY;BN_FMT RI;BYT_OR MSB;WFID "Ch1, DC coupling, 200.0mV/div, 10.00us/div, 1250 points, Sample mode";NR_PT 1000;PT_FMT Y;XUNIT "s";XINCR 80.0000E-9;XZERO 0.0000;PT_OFF 625;YUNIT "V";YMULT 31.2500E-6;YOFF 0.0000;YZERO 0.0000;NR_FR 3.

WFMOutpre:ASC_Fmt? (Query Only)

This query returns the format for ASCII data transferred from the instrument. No command form is provided as the format is determined by the data source type. Some waveforms are normalized vector data where the data points are 8-byte doubles in floating point format whereas other formats are 1-byte or 2-byte integers.

Group Waveform Transfer

Syntax WFMOutpre:ASC_Fmt?

Related Commands

DATa:SOUrce

WFMOutpre:BN_Fmt

WFMOutpre:ENCdg

Returns

`FP` represents floating point ASCII data. The waveforms are normalized vector data where the data points are 8-byte doubles in floating point format.

`INTEGER` represents signed integer ASCII data. The waveform data are 1-byte or 2-byte integers.

Examples

`WFMOutpre:ASC_Fmt?` might return `WFMOUTPRE:ASC_FMT INTEGER` indicating the ASCII format is integer.

WFMOutpre:BIT_Nr

This command sets and queries the number of bits per waveform point that outgoing waveforms contain, as specified by the DATa:SOUrce command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when WFMOutpre:ENCdg is set to BIN and WFMOutpre:BN_Fmt is set to either RI or RP.

Group

Waveform Transfer

Syntax

```
WFMOutpre:BIT_Nr <NR1>
WFMOutpre:BIT_Nr?
```

Related Commands

DATa:SOUrce

WFMOutpre:BN_Fmt

WFMOutpre:ENCdg

Arguments

`<NR1>` number of bits per data point can be 8 or 16.

Examples

`WFMOUTPRE:BIT_NR 16` sets the number of bits per waveform point to 16 for incoming RI and RP binary format data.

`WFMOUTPRE:BIT_NR?` might return `WFMOUTPRE:BIT_NR 8`, indicating that outgoing RI or RP binary format data uses 8 bits per waveform point.

WFMOutpre:BN_Fmt

This command sets or queries the format of binary data for outgoing waveforms specified by the DATa:SOUrce command.

Group Waveform Transfer

Syntax

```
WFMOutpre:BN_Fmt {RI|RP|FP}
WFMOutpre:BN_Fmt?
```

Related Commands DATa:SOUrce

Arguments

`RI` specifies signed integer data point representation.

`RP` specifies positive integer data point representation.

`FP` specifies floating point representation.

Examples

`WFMOUTPRE:BN_FMT FP` specifies that outgoing waveform data will be in single-precision binary floating point format.

`WFMOUTPRE:BN_FMT?` might return `WFMOUTPRE:BN_FMT RI`, indicating that the outgoing waveform data is currently in signed integer format.

WFMOutpre:BYT_Nr

This command sets or queries the binary field data width (bytes per point) for the waveform specified by the DATa:SOUrce command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when WFMOutpre:ENCdg is set to BIN, and WFMOutpre:BN_Fmt is set to either RI or RP.

Group Waveform Transfer

Syntax

```
WFMOutpre:BYT_Nr <NR1>
WFMOutpre:BYT_Nr?
```

Related Commands DATa:SOUrce

WFMOutpre:BN_Fmt

WFMOutpre:ENCdg

Arguments <NR1> is the number of bytes per data point and can be 1, 2 or 8. A value of 1 or 2 bytes per waveform point indicates channel data; 8 bytes per waveform point indicate pixel map (fast acquisition) data.

Examples WFMOUTPRE:BYT_NR 1 sets the number of bytes per outgoing waveform data point to 1, which is the default setting.

WFMOUTPRE:BYT_NR? might return WFMOUTPRE:BYT_NR 2, indicating that there are 2 bytes per outgoing waveform data point.

WFMOutpre:BYT_Or

This command sets or queries which byte of binary waveform data is transmitted first, during a waveform data transfer, when data points require more than one byte. This specification only has meaning when WFMOutpre:ENCdg is set to BIN.

Group Waveform Transfer

Syntax
```
WFMOutpre:BYT_Or {LSB|MSB}
WFMOutpre:BYT_Or?
```

Related Commands WFMOutpre:ENCdg

Arguments LSB specifies that the least significant byte will be transmitted first.

MSB specifies that the most significant byte will be transmitted first.

Examples WFMOUTPRE:BYT_OR MSB sets the most significant outgoing byte of waveform data to be transmitted first.

WFMOUTPRE:BYT_OR? might return WFMOUTPRE:BYT_OR LSB, indicating that the least significant data byte will be transmitted first.

WFMOutpre:CENTERFREQuency? (Query Only)

This query returns the center frequency of the outgoing waveform. The center frequency is a precise frequency at the center of the display of the frequency domain graticule. In many applications, it is a carrier frequency.

Group Waveform Transfer

Syntax `WFMOutpre:CENTERFREQuency?`

Related Commands DATa:SOUrce

Returns Returns the center frequency of the outgoing waveform, in Hertz.

Examples `WFMOutpre:CENTERFREQuency?` might return `WFMOutpre:CENTERFREQuency 1.0E+9`, indicating a center frequency of 1 GHz.

WFMOutpre:DOMain? (Query Only)

This query returns the domain of the outgoing waveform, either TIMe or FREQuency. If the domain is TIMe, it indicates that the data is to be treated as integer information. If the domain is FREQuency, it indicates that the data is to be treated as floating point information.

Group Waveform Transfer

Syntax `WFMOutpre:DOMain?`

Related Commands DATa:SOUrce

WFMOutpre:BN_Fmt

WFMOutpre:ENCdg

Returns Returns the domain of the outgoing waveform.

Examples `WFMOUTPRE:DOMAIN?` might return `WFMOUTPRE:DOMAIN TIME`, indicating that the outgoing waveform is a time domain trace.

`WFMOutpre:DOMain?` might return `WFMOutpre:DOMain FREQuency`, indicating that the outgoing waveform is one of the four Spectrum View frequency domain traces (CH<x>_SV_NORMal, CH<x>_SV_AVErage, CH<x>_SV_MINHOLD, or CH<x>_SV_MAXHOLD).

WFMOutpre:ENCdg

This command sets or queries the type of encoding for outgoing waveforms.

Group Waveform Transfer

Syntax
```
WFMOutpre:ENCdg {ASCii|BINary}
WFMOutpre:ENCdg?
```

Related Commands DATa:ENCdg

WFMOutpre:BYT_Nr

WFMOutpre:BYT_Or

WFMOutpre:BIT_Nr

WFMOutpre:BN_Fmt

Arguments `ASCii` specifies that the outgoing data is to be in `ASCII` format. Waveforms internally stored as integers will be sent as <NR1> numbers, while those stored as floating point will be sent as <NR3> numbers.

`BINary` specifies that outgoing data is to be in a binary format whose further specification is determined by WFMOutpre:BYT_Nr, WFMOutpre:BIT_Nr, WFMOutpre:BN_Fmt and WFMOutpre:BYT_Or.

Examples `WFMOUTPRE:ENCDG ASCii` specifies that the outgoing waveform data will be sent in ASCII format.

`WFMOUTPRE:ENCDG?` might return `WFMOUTPRE:ENCDG BINary`, indicating that outgoing waveform data will be sent in binary format.

WFMOutpre:NR_Pt? (Query Only)

This query-only command returns the number of points for the DATa:SOUrce waveform that will be transmitted in response to a CURVe? query.

Group Waveform Transfer

Syntax
```
WFMOutpre:NR_Pt?
```

Related Commands CURVe

DATa

DATa:STARt

DATa:STOP

SAVe:WAVEform

SAVEON:WAVEform:FILEFormat

Examples `WFMOUTPRE:NR_PT?` might return `WFMOUTPRE:NR_PT 5000`, indicating that there are 5000 data points to be sent.

WFMOutpre:PT_Fmt? (Query Only)

This query-only command returns the point format for the waveform specified by the DATa:SOUrce command. The format specifies a set of equations describing how the scale factors in the preamble are used to give meaning to the CURVe data points.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax `WFMOutpre:PT_Fmt?`

Related Commands CURVe

DATa:SOUrce

Examples `WFMOUTPRE:PT_FMT?` might return `WFMOutpre:PT_Fmt ENV`, indicating that the waveform data is a series of min-max pairs.

WFMOutpre:PT_Off? (Query Only)

This query-only command returns the trigger point relative to DATa:STARt for the waveform specified by the DATa:SOUrce command.

NOTE. *This returned value is the point immediately following the actual trigger.*

Group Waveform Transfer

Syntax `WFMOutpre:PT_Off?`

Related Commands DATa:SOUrce

DATa:STARt

WFMOutpre:XZEro?

Examples `WFMOUTPRE:PT_OFF?` might return `WFMOUTPRE:PT_OFF 251` specifying that the trigger actually occurred between points 250 and 251.

WFMOutpre:RESample

This command sets or queries the resampling of outgoing waveform data. This command is equivalent to setting `DATa:RESample`. Setting the `WFMOutpre:RESample` value causes the corresponding DATa value to be updated and vice versa.

Group Waveform Transfer

Syntax

```
WFMOutpre:RESample <NR1>
WFMOutpre:RESample?
```

Related Commands DATa:RESample

Arguments `<NR1>` is the resampling rate. The default value is 1, which means every sample is returned. A value of 2 returns every other sample, while a value of 3 returns every third sample, and so on.

Examples `WFMOUTPRE:RESAMPLE 1` sets the resampling to return all waveform data samples.

`WFMOUTPRE:RESAMPLE?` might return `WFMOUTPRE:RESAMPLE 4`, indicating that every forth sample of the waveform data is returned.

WFMOutpre:SPAN? (Query Only)

This query returns the frequency span of the outgoing waveform. The span is the range of frequencies you can observe around the center frequency.

Group Waveform Transfer

Syntax `WFMOutpre:SPAN?`

Related Commands DATa:SOUrce

Returns Returns the frequency span of the outgoing waveform, in Hertz.

Examples `RF:SPAN?` might return `RF:SPAN 10.0000E+6`, indicating the span is 10 MHz.

WFMOutpre:WFId? (Query Only)

This query-only command returns a string describing several aspects of the acquisition parameters for the waveform specified by the DATa:SOUrce command.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax `WFMOutpre:WFId?`

Related Commands DATa:SOUrce

Returns `<QString>` contains the following comma-separated fields documented in the following tables:

Table 2-52: Waveform Suffixes

Field	Description	Examples
Source	The source identification string as it appears in the front panel scale factor readouts.	"CH1–8" "Math<x>" "Ref<x>"
Coupling	A string describing the vertical coupling of the waveform (the Source1 waveform in the case of Dual Waveform Math).	"AC coupling" "DC coupling" "GND coupling"
Vert Scale	A string containing the vertical scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units.	"100.0 mV/div" "20.00 dB/div" "45.00 deg/div" "785.4 mrad/div" "500.0 µVs/div" "10.00 kV/s/div" "200.0 mV/div" "50.00 unk/div"

Table 2-52: Waveform Suffixes (cont.)

Field	Description	Examples
Horiz Scale	A string containing the horizontal scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units.	"100 ms/div" "10.00 kHz/div" "50.00 c/div"
Record Length	A string containing the number of waveform points available in the entire record. The numeric portion is given as an integer.	"500 points" "500000 points"
Acquisition Mode	A string describing the mode used to acquire the waveform.	"Sample mode" "Pk Detect mode" "Hi Res mode" "Envelope mode" "Average mode"
Center Frequency	A string containing the center frequency value of the waveform.	“2.400GHz”
Span	A string containing the frequency span value of the waveform.	“20.00MHz”
Reference Level	A string containing the reference level value of the waveform.	“1.000mW”

Examples `WFMOUTPRE:WFID?` might return `WFMOUTPRE:WFID "Ch1, DC coupling,100.0mVolts/div,500.0µs/div,500 points, Hi Res mode"`.

WFMOutpre:WFMTYPe? (Query Only)

This query returns the type of the outgoing waveform.

The type of waveform that is being transferred in turn determines in which window it will display on the instrument screen: (the Waveform View window or Spectrum View window). Both the Analog and RF_TD arguments specify the time domain window; the RF_RD argument specifies the frequency domain window.

Group Waveform Transfer

Syntax `WFMOutpre:WFMTYPe?`

Related Commands DATa:SOUrce

Returns Returns the type of the outgoing waveform:

ANALOG indicates a Channel or Math waveform.

SV_FD indicates a Spectrum View RF frequency domain waveform.

RF_TD indicates an RF time domain waveform.

Examples WFMOutpre:WFMTYPe? might return WFMOutpre:WFMTYPe CH<x>_SV_FD, which indicates that the outgoing waveform is an RF frequency domain waveform (CH<x>_SV_AVErage, CH<x>_SV_MAXHold, CH<x>_SV_MINHold, or CH<x>_SV_NORMal).

WFMOutpre:XINcr? (Query Only)

This query-only command returns the horizontal point spacing in units of WFMOutpre:XUNit for the waveform specified by the DATa:SOUrce command. This value corresponds to the sampling interval.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax WFMOutpre:XINcr?

Related Commands DATa:SOUrce

WFMOutpre:XUNit?

Examples WFMOUTPRE:XINCR? might return WFMOUTPRE:XINCR 10.0000E-6, indicating that the horizontal sampling interval is 10 µs/point (500 µs/div).

WFMOutpre:XUNit? (Query Only)

This query-only command returns the horizontal units for the waveform specified by the DATa:SOUrce command.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax WFMOutpre:XUNit?

Related Commands DATa:SOUrce

Examples `WFMOUTPRE:XUNIT?` might return `WFMOUTPRE:XUNIT "HZ"`, indicating that the horizontal units for the waveform are in Hertz.

WFMOutpre:XZEro? (Query Only)

This query-only command returns the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger for the waveform specified by the DATa:SOUrce command. This value is in units of WFMOutpre:XUNit.

An error is reported if the DATa:SOUrce waveform does not exist.

NOTE. *During steady state operation, when all control changes have settled and triggers are arriving on a regular basis, this is the only part of the preamble that changes on each acquisition.*

Group Waveform Transfer

Syntax `WFMOutpre:XZEro?`

Related Commands DATa:SOUrce

WFMOutpre:XUNit?

Examples `WFMOUTPRE:XZERO?` might return `WFMOUTPRE:XZERO 5.6300E-9`, indicating that the trigger actually occurred 5.63 ns before the trigger sample.

WFMOutpre:YMUlt? (Query Only)

This query-only command returns the vertical scale factor per digitizing level in units specified by WFMOutpre:YUNit for the waveform specified by the DATa:SOUrce command. For those formats in which WFMOutpre:BYT_Nr is important (all non-floating point formats), WFMOutpre:YMUlt? must take the location of the binary point implied by BYT_NR into consideration.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax `WFMOutpre:YMUlt?`

Related Commands DATa:SOUrce

Examples `WFMOUTPRE:YMULT?` might return `WFMOUTPRE:YMULT 4.0000E-3`, indicating that the vertical scale for the corresponding waveform is 100 mV/div.

WFMOutpre:YOFf? (Query Only)

This query-only command returns the vertical offset of the source specified by DATa:SOUrce. For this instrument family, the value returned is always 0.0 as the offset is combined with the :YZEro value.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax `WFMOutpre:YOFf?`

Related Commands DATa:SOUrce

WFMOutpre:BYT_Nr

Examples `WFMOUTPRE:YOFF?` might return `WFMOUTPRE:YOFF -50.0000E+0`, indicating that the position indicator for the waveform was 50 digitizing levels (2 divisions) below center screen.

WFMOutpre:YUNit? (Query Only)

This query-only command returns the vertical units for the waveform specified by the DATa:SOUrce command.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax `WFMOutpre:YUNit?`

Related Commands DATa:SOUrce

Examples `WFMOUTPRE:YUNIT?` might return `WFMOUTPRE:YUNIT "dB"`, indicating that the vertical units for the waveform are measured in decibels.

WFMOutpre:YZEro? (Query Only)

This query-only command returns the combined vertical position and offset for the source waveform specified by DATa:SOUrce. This represents a departure from previous instruments where the :YZEro value represented the vertical position in vertical units and the :YOFf value represented the vertical offset in digitizing levels. For this instrument family, the value of :YOFf is always 0.0.

An error is reported if the DATa:SOUrce waveform does not exist.

Group Waveform Transfer

Syntax `WFMOutpre:YZEro?`

Related Commands DATa:SOUrce

WFMOutpre:YUNit?

Examples `WFMOUTPRE:YZERO?` might return `WFMOUTPRE:YZERO -100.0000E-3`, indicating that vertical offset is set to -100 mV.

Status and Events

The instrument provides a status and event reporting system for the Ethernet and USB interfaces. This system informs you of certain significant events that occur within the instrument.

The instrument status handling system consists of five 8-bit registers and two queues for each interface. The remaining Status subtopics describe these registers and components. They also explain how the event handling system operates.

Registers

Overview

The registers in the event handling system fall into two functional groups:

- Status Registers contain information about the status of the instrument. They include the Standard Event Status Register (SESR).
- Enable Registers determine whether selected types of events are reported to the Status Registers and the Event Queue. They include the Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER).

Status Registers

The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the instrument is in use. IEEE Std 488.2-1987 defines these registers.

Each bit in a Status Register records a particular type of event, such as an execution error or message available. When an event of a given type occurs, the instrument sets the bit that represents that type of event to a value of one. (You can disable bits so that they ignore events and remain at zero. See Enable Registers). Reading the status registers tells you what types of events have occurred.

The Standard Event Status Register (SESR). The SESR records eight types of events that can occur within the instrument. Use the *ESR? query to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events.

NOTE. *TekVISA applications use SESR bit 6 to respond to any of several events, including some front panel actions.*

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 3-1: The Standard Event Status Register (SESR)

Table 3-1: SESR Bit Functions

Bit	Function	
7 (MSB)	PON	`Power On.` Shows that the instrument was powered on. On completion, the diagnostic self tests also set this bit.
6	URQ	`User Request.` Indicates that an application event has occurred. *See note.
5	CME	`Command Error.` Shows that an error occurred while the instrument was parsing a command or query.
4	EXE	`Execution Error.` Shows that an error executing a command or query.
3	DDE	`Device Error.` Shows that a device error occurred.
2	QYE	`Query Error.` Either an attempt was made to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost.
1	RQC	`Request Control.` This is not used.
0 (LSB)	OPC	`Operation Complete.` Shows that the operation is complete. This bit is set when all pending operations complete following an *OPC command.

The Status Byte Register (SBR). Records whether output is available in the Output Queue, whether the instrument requests service, and whether the SESR has recorded any events.

Use a Serial Poll or the *STB? query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the *STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits.

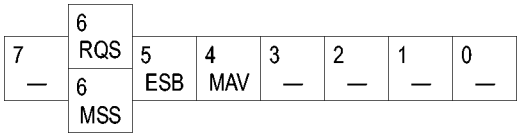


Figure 3-2: The Status Byte Register (SBR)

Table 3-2: SBR Bit Functions

Bit	Function	
7 (MSB)	———	Not used.
6	RQS	`Request Service.` Obtained from a serial poll. Shows that the instrument requests service from the controller.
6	MSS	`Master Status Summary.` Obtained from *STB? query. Summarizes the ESB and MAV bits in the SBR.
5	ESB	`Event Status Bit.` Shows that status is enabled and present in the SESR.

Table 3-2: SBR Bit Functions (cont.)

Bit	Function	
4	MAV	`Message Available.` Shows that output is available in the Output Queue.
3	————	Not used.
2	————	Not used.
1–0	————	Not used.

Enable Registers

DESER, ESER, and SRER allow you to select which events are reported to the Status Registers and the Event Queue. Each Enable Register acts as a filter to a Status Register (the DESER also acts as a filter to the Event Queue) and can prevent information from being recorded in the register or queue.

Each bit in an Enable Register corresponds to a bit in the Status Register it controls. In order for an event to be reported to a bit in the Status Register, the corresponding bit in the Enable Register must be set to one. If the bit in the Enable Register is set to zero, the event is not recorded.

Various commands set the bits in the Enable Registers. The Enable Registers and the commands used to set them are described below.

The Device Event Status Enable Register (DESER). This register controls which types of events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 3-3: The Device Event Status Enable Register (DESER)

The Event Status Enable Register (ESER). This register controls which types of events are summarized by the Event Status Bit (ESB) in the SBR. Use the *ESE command to set the bits in the ESER. Use the *ESE? query to read it.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 3-4: The Event Status Enable Register (ESER)

The Service Request Enable Register (SRER). This register controls which bits in the SBR generate a Service Request and are summarized by the Master Status Summary (MSS) bit.

Use the *SRE command to set the SRER. Use the *SRE? query to read the register. The RQS bit remains set to one until either the Status Byte Register is read with a Serial Poll or the MSS bit changes back to a zero.

7	6	5	4	3	2	1	0
—	—	ESB	MAV	—	—	—	—

Figure 3-5: The Service Request Enable Register (SRER)

*PSC Command

The *PSC command controls the Enable Registers contents at power-on. Sending *PSC 1 sets the Enable Registers at power on as follows:

- DESER 255 (equivalent to a DESe 255 command)
- ESER 0 (equivalent to an *ESE 0 command)
- SRER 0 (equivalent to an *SRE 0 command)

Sending *PSC 0 lets the Enable Registers maintain their values in nonvolatile memory through a power cycle.

NOTE. *To enable the PON (Power On) event to generate a Service Request, send *PSC 0, use the DESe and *ESE commands to enable PON in the DESER and ESER, and use the *SRE command to enable bit 5 in the SRER. Subsequent power-on cycles will generate a Service Request.*

Queues

The *PSC command controls the Enable Registers contents at power-on. Sending *PSC 1 sets the Enable Registers at power on as follows:

Output Queue

The instrument stores query responses in the Output Queue and empties this queue each time it receives a new command or query message after an <EOM>. The controller must read a query response before it sends the next command (or query) or it will lose responses to earlier queries.



CAUTION. *When a controller sends a query, an <EOM>, and a second query, the instrument normally clears the first response and outputs the second while reporting a Query Error (QYE bit in the ESER) to indicate the lost response. A fast controller, however, may receive a part or all of the first response as well. To avoid this situation, the controller should always read the response immediately after sending any terminated query message or send a DCL (Device Clear) before sending the second query.*

Event Queue

The Event Queue stores detailed information on up to 33 events. If than 32 events stack up in the Event Queue, the 32nd event is replaced by event code 350, "Queue Overflow."

Read the Event Queue with the EVENT? query (which returns only the event number), with the EVMSG? query (which returns the event number and a text description of the event), or with the ALLEV? query (which returns all the event numbers along with a description of the event). Reading an event removes it from the queue.

Before reading an event from the Event Queue, you must use the *ESR? query to read the summary of the event from the SESR. This makes the events summarized by the *ESR? read available to the EVENT? and EVMSG? queries, and empties the SESR.

Reading the SESR erases any events that were summarized by previous *ESR? reads but not read from the Event Queue. Events that follow an *ESR? read are put in the Event Queue but are not available until *ESR? is used again.

Event Handling Sequence

The following figure shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in the figure.

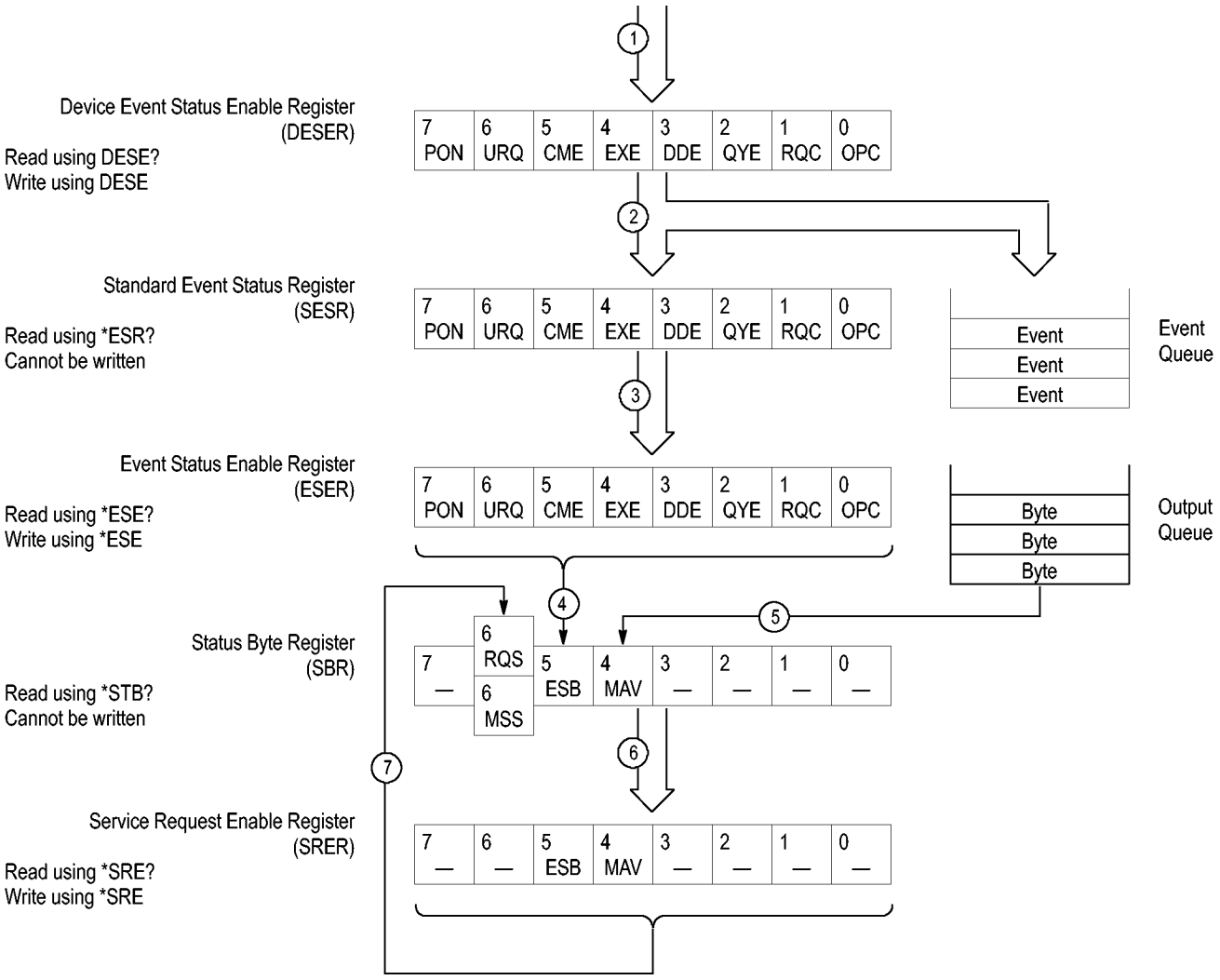


Figure 3-6: Status and Event Handling Process

When an event occurs, a signal is sent to the DESER (1). If that type of event is enabled in the DESER (that is, if the bit for that event type is set to 1), the appropriate bit in the SESR is set to one, and the event is recorded in the Event Queue (2). If the corresponding bit in the ESER is also enabled (3), then the ESB bit in the SBR is set to one (4).

When output is sent to the Output Queue, the MAV bit in the SBR is set to one (5).

When a bit in the SBR is set to one and the corresponding bit in the SRER is enabled (6), the MSS bit in the SBR is set to one and a service request is generated (7).

Synchronization Methods

Overview

Although most commands are completed almost immediately after being received by the instrument, some commands start a process that requires time. For example, once a single sequence acquisition command is executed, depending upon the applied signals and trigger settings, it may take an extended period of time before the acquisition is complete. Rather than remain idle while the operation is in process, the instrument will continue processing other commands. This means that some operations will not be completed in the order that they were sent. Furthermore, sometimes the result of an operation depends upon the result of an earlier operation. A first operation must complete before the next one is processed.

In order to handle these situations, the instrument status and event reporting system allows you to synchronize the operation of the instrument with your application program, using the Operation Complete function. Note, however, that only some operations are able to take advantage of this function; a table is provided below of commands that support this.

The following commands are used to synchronize the instrument functions using Operation Complete. See examples of how to use these commands later on in this section:

*OPC — sending the *OPC command will set bit 0 of the SESR (Standard Events Status Register). The bit will only be set high when all pending operations that generate an OPC message have finished execution. (The SESR is queried using *ESR?) The `*OPC?` query form returns 1 only when all operations have completed, or a device clear is received.

*WAI — prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete.

BUSY? — returns the status of operations: 1 means there are pending operations, 0 means that all pending operations are complete.

NOTE. *Some OPC operations may require an extended period of time to complete or may never complete. For example, a single sequence acquisition may never complete when no trigger event occurs. You should be aware of these conditions and tailor your program accordingly by:*

— setting the timeout sufficiently for the anticipated maximum time for the operation and

*— handle a timeout appropriately by querying the SESR (*ESR?) and event queue (EVMsg? or ALLEv?).*

NOTE. *The *OPC command form can also be used to cause an SRQ to be generated upon completion of all pending operations. This requires that the ESB (Event Status Bit, bit 5) of the SRER (Service Request Enable Regiser) is set, and the OPC bit (bit 0) of the DESR (Device Event Status Enable Register) and the ESER (Event Status Enable Register) are set. (The SRER is set/queried using *SRE. The DESR is set/queried using DESE. The ESER is set/queried using *ESE.)*

Only a subset of instrument operations support the Operation Complete function (OPC):

Table 3-3: instrument operations that can generate OPC

Command	Conditions
ACQuire:STATE <non-zero NR1>| ON|RUN	Only when in single sequence acquisition mode
:AUTOset < EXECute >	
CALibrate:INTERNal	
CALibrate:INTERNal:STARt	
CALibrate:FACtory STARt	
CALibrate:FACtory CONTinue	
CALibrate:FACtory PREVious	
CH<x>:PRObe:AUTOZero EXECute	
CH<x>:PRObe:DEGAUss EXECute	
DIAg:STATE EXECute	
FACtory	
MEASUrement:MEAS<x> :RESUlts	When used in single sequence acquisition mode or during waveform recall.
RECAll:SETUp (<file as quoted string>|FACtory)	
RECAll:WAVEform <.ISF or .CSV file >,< REF<x> >	
:RF:REFLevel AUTO	
*RST	
SAVe:IMAGe <file as quoted string>	
SAVe:SETUp <file as quoted string>	
SAVe:WAVEform < source wfm >, (< REF<x> >|< file >)	
TEKSecure	
TRIGger:A SETLevel	

Example of Acquiring and Measuring a Single-Sequence Waveform

For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

```
/** Set up conditional acquisition **/
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/** Acquire waveform data **/
ACQUIRE:STATE ON
/** Set up the measurement parameters **/
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/** Take amplitude measurement **/
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?
```

The acquisition of the waveform requires extended processing time. It may not finish before the instrument takes an amplitude measurement (see the following figure). This can result in an incorrect amplitude value.

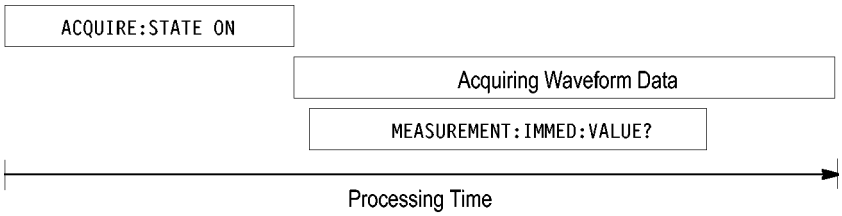


Figure 3-7: Command Processing Without Using Synchronization

To ensure the instrument completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program using *WAI, BUSY, *OPC, and *OPC?.

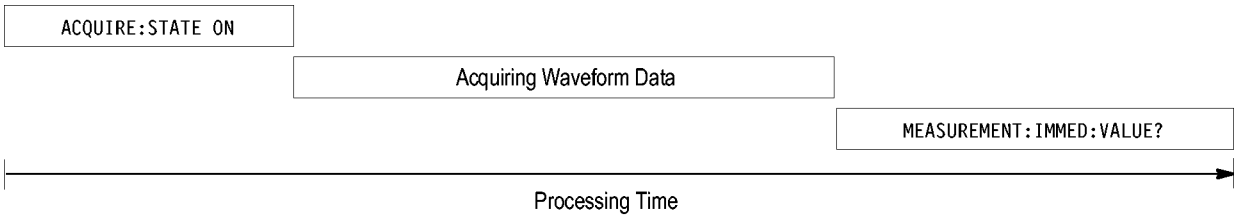


Figure 3-8: Processing Sequence With Synchronization

Example of Using the *OPC Command

If the corresponding status registers are enabled, the *OPC command sets the OPC bit in the Standard Event Status Register (SESR) when an operation is complete. You achieve synchronization by using this command with either a serial poll or service request handler.

Serial Poll Method: Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands.

When the operation is complete, the OPC bit in the Standard Event Status Register (SESR) will be enabled and the Event Status Bit (ESB) in the Status Byte Register will be enabled.

The same command sequence using the *OPC command for synchronization with serial polling looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 0
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters on the channel we're
about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Wait until the acquisition is complete before taking the
measurement.  */
*OPC
While serial poll = 0, keep looping
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?
```

This technique requires less bus traffic than did looping on BUSY.

Service Request Method: Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands.

You can also enable service requests by setting the ESB bit in the Service Request Enable Register (SRER) using the *SRE command. When the operation is complete, the instrument will generate a Service Request.

The same command sequence using the *OPC command for synchronization looks like this

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
```

```
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 32

/* Set up the measurement parameters on the channel we're
about to sequence */ :MEASUrement:MEAS1:TYPE AMPLITUDE
:MEASUrement:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Wait until the acquisition is complete before taking the
measurement*/
*OPC
```

The program can now do different tasks such as talk to other devices. The SRQ, when it comes, interrupts those tasks and returns control to this task.

```
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?
```

Example of Using the *OPC? Query

The *OPC? query places a 1 in the Output Queue once an operation that generates an OPC message is complete. The *OPC? query does not return until all pending OPC operations have completed. Therefore, your time-out must be set to a time at least as long as the longest expected time for the operations to complete.

The same command sequence using the *OPC? query for synchronization looks like this:

```
/* Set up single sequence acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Set up the measurement parameters on the channel we're
about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Wait until the acquisition is complete before taking the
measurement*/
*OPC?
```

Wait for read from Output Queue.

```
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?
```

This is the simplest approach. It requires no status handling or loops. However, you must set the controller time-out for longer than the acquisition operation.

Example of Using the *WAI Command

The *WAI command forces completion of previous commands that generate an OPC message. No commands after the *WAI are processed before the OPC message(s) are generated

The same command sequence using the *WAI command for synchronization looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Set up the measurement parameters on the channel we're
about to sequence */
MEASUREMENT:MEAS1:TYPE AMPLITUDE
MEASUREMENT:MEAS1:SOURCE CH1
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Wait until the acquisition is complete before taking
the measurement*/
*/
*WAI
/* Take amplitude measurement */
MEASUREMENT:MEAS1:RESUlts:CURRentacq:MEAN?
```

The controller can continue to write commands to the input buffer of the instrument, but the commands will not be processed by the instrument until all in-process OPC operations are complete. If the input buffer becomes full, the controller will be unable to write commands to the buffer. This can cause a time-out.

Example of Using the BUSY Query

The BUSY? query allows you to find out whether the instrument is busy processing a command that has an extended processing time such as single-sequence acquisition.

The same command sequence, using the BUSY? query for synchronization, looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
DISPLAY:WAVEVIEW1:CH1:STATE 1
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
```

```
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking
the measurement */
While BUSY? keep looping
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE?
```

This sequence lets you create your own wait loop rather than using the *WAI command. The BUSY? query helps you avoid time-outs caused by writing too many commands to the input buffer. The controller is still tied up though, and the repeated BUSY? query will result in bus traffic.

Reference waveforms

Measurements on references also support OPC when used in conjunction with a :RECALL:WAVEFORM command.

```
/* Add a reference slot on which to measure */

:REF:ADDNEW "REF1"

/* Set up the measurement parameters on the reference */

:MEASUrement:MEAS1:TYPE AMPLITUDE

:MEASUrement:MEAS1:SOURCE REF1

/* Load the new waveform file */

:RECALL:WAVEFORM "E:\waveform.wfm",REF1

*OPC?

/* Wait for read from Output Queue.  */

/* Take amplitude measurement */

:MEASUrement:MEAS1:RESUlts:CURRentacq:MEAN?
```

Messages

The information contained in the topics above covers all the programming interface messages the instrument generates in response to commands and queries.

For most messages, a secondary message from the instrument gives detail about the cause of the error or the meaning of the message. This message is part of the message string and is separated from the main message by a semicolon.

Each message is the result of an event. Each type of event sets a specific bit in the SESR and is controlled by the equivalent bit in the DESER. Thus, each message is associated with a specific SESR bit. In the message tables, the associated SESR bit is specified in the table title, with exceptions noted with the error message text.

No Event

The following table shows the messages when the system has no events or status to report. These have no associated SESR bit.

Table 3-4: No Event Messages

Code	Message
0	No events to report; queue empty
1	No events to report; new events pending *ESR?

Command Error

The following table shows the command error messages generated by improper syntax. Check that the command is properly formed and that it follows the rules in the section on command Syntax.

Table 3-5: Command Error Messages (CME Bit 5)

Code	Message
100	Command error
101	Invalid character
102	Syntax error
103	Invalid separator
104	Data type error
105	GET not allowed
108	Parameter not allowed
109	Missing parameter
110	Command header error
112	Program mnemonic too long
113	Undefined header
120	Numeric data error
121	Invalid character in numeric
123	Exponent too large
124	Too many digits
130	Suffix error
131	Invalid suffix
134	Suffix too long
140	Character data error
141	Invalid character data
144	Character data too long
150	String data error
151	Invalid string data
152	String data too long
160	Block data error

Table 3-5: Command Error Messages (CME Bit 5) (cont.)

Code	Message
161	Invalid block data
170	Command expression error
171	Invalid expression

Execution Error

The following table lists the execution errors that are detected during execution of a command.

Table 3-6: Execution Error Messages (EXE Bit 4)

Code	Message
200	Execution error
221	Settings conflict
222	Data out of range
224	Illegal parameter value
241	Hardware missing
250	Mass storage error
251	Missing mass storage
252	Missing media
253	Corrupt media
254	Media full
255	Directory full
256	File name not found
257	File name error
258	Media protected
259	File name too long
280	Program error
282	Insufficient network printer information
283	Network printer not responding
284	Network printer server not responding
286	Program runtime error
287	Future use
2200	Measurement error, Measurement system error
2201	Measurement error, Zero period
2202	Measurement error, No period, second waveform
2203	Measurement error, No period, second waveform
2204	Measurement error, Low amplitude, second waveform
2205	Measurement error, Low amplitude, second waveform

Table 3-6: Execution Error Messages (EXE Bit 4) (cont.)

Code	Message
2206	Measurement error, Invalid gate
2207	Measurement error, Measurement overflow
2208	Measurement error, No backwards Mid Ref crossing
2209	Measurement error, No second Mid Ref crossing
2210	Measurement error, No Mid Ref crossing, second waveform
2211	Measurement error, No backwards Mid Ref crossing
2212	Measurement error, No negative crossing
2213	Measurement error, No positive crossing
2214	Measurement error, No crossing, target waveform
2215	Measurement error, No crossing, second waveform
2216	Measurement error, No crossing, target waveform
2217	Measurement error, Constant waveform
2219	Measurement error, No valid edge - No arm sample
2220	Measurement error, No valid edge - No arm cross
2221	Measurement error, No valid edge - No trigger cross
2222	Measurement error, No valid edge - No second cross
2223	Measurement error, Waveform mismatch
2224	Measurement error, WAIT calculating
2225	Measurement error, No waveform to measure
2226	Measurement error, Null Waveform
2227	Measurement error, Positive and Negative Clipping
2228	Measurement error, Positive Clipping
2229	Measurement error, Negative Clipping
2230	Measurement error, High Ref < Low Ref
2231	Measurement error, No statistics available
2233	Requested waveform is temporarily unavailable
2235	Math error, invalid math description
2240	Invalid password
2241	Waveform requested is invalid
2244	Source waveform is not active
2245	Saveref error, selected channel is turned off
2250	Reference error, the reference waveform file is invalid
2253	Reference error, too many points received
2254	Reference error, too few points received
2259	File too big
2270	Alias error
2271	Alias syntax error

Table 3-6: Execution Error Messages (EXE Bit 4) (cont.)

Code	Message
2273	Illegal alias label
2276	Alias expansion error
2277	Alias redefinition not allowed
2278	Alias header not found
2285	TekSecure(R) Pass
2286	TekSecure(R) Fail
2500	Setup error, file does not look like a setup file
2501	Setup warning, could not recall all values from external setup
2620	Mask error, too few points received
2760	Mark limit reached
2761	No mark present
2762	Search copy failed

Device Error

The following table lists the device errors that can occur during instrument operation. These errors may indicate that the instrument needs repair.

Table 3-7: Device Error Messages (DDE Bit 3)

Code	Message
310	System error
311	Memory error
312	PUD memory lost
314	Save/recall memory lost

System Event

The following table lists the system event messages. These messages are generated whenever certain system conditions occur.

Table 3-8: System Event Messages

Code	Message
400	Query event
401	Power on (PON bit 7 set)
402	Operation complete (OPC bit 0 set)
403	User request (URQ bit 6 set)
404	Power fail (DDE bit 3 set)
405	Request control
410	Query INTERRUPTED (QYE bit 2 set)
420	Query UNTERMINATED (QYE bit 2 set)

Table 3-8: System Event Messages (cont.)

Code	Message
430	Query DEADLOCKED (QYE bit 2 set)
440	Query UNTERMINATED after indefinite response (QYE bit 2 set)
468	Knob/Keypad value changed
472	Application variable changed

Execution Warning

The following table lists warning messages that do not interrupt the flow of command execution. They also notify you of possible unexpected results.

Table 3-9: Execution Warning Messages (EXE Bit 4)

Code	Message
528	Parameter out of range
532	Curve data too long, Curve truncated
533	Curve error, Preamble values are inconsistent
540	Measurement warning, Uncertain edge
541	Measurement warning, Low signal amplitude
542	Measurement warning, Unstable histogram
543	Measurement warning, Low resolution
544	Measurement warning, Uncertain edge
545	Measurement warning, Invalid in minmax
546	Measurement warning, Need 3 edges
547	Measurement warning, Clipping positive/negative
548	Measurement warning, Clipping positive
549	Measurement warning, Clipping negative

Table 3-10: Execution Warning Messages (EXE Bit 4)

Code	Message
540	Measurement warning
541	Measurement warning, Low signal amplitude
542	Measurement warning, Unstable histogram
543	Measurement warning, Low resolution
544	Measurement warning, Uncertain edge
545	Measurement warning, Invalid min max
546	Measurement warning, Need 3 edges
547	Measurement warning, Clipping positive/negative
548	Measurement warning, Clipping positive
549	Measurement warning, Clipping negative

Internal Warning

The following table shows internal errors that indicate an internal fault in the instrument.

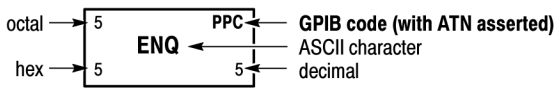
Table 3-11: Internal Warning Messages

Code	Message
630	Internal warning, 50Ω overload

Appendix A: Character Set

B7 B6 B5 BITS B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL 0 0	20 DLE 10 16	40 LA0 SP 20 32	60 LA16 0 30 48	100 TA0 @ 40 64	120 TA16 P 50 80	140 SA0 ` 60 96	160 SA16 p 70 112
0 0 0 1	1 GTL SOH 1 1	21 LL0 DC1 11 17	41 LA1 ! 21 33	61 LA17 1 31 49	101 TA1 A 41 65	121 TA17 Q 51 81	141 SA1 a 61 97	161 SA17 q 71 113
0 0 1 0	2 STX 2 2	22 DC2 12 18	42 LA2 " 22 34	62 LA18 2 32 50	102 TA2 B 42 66	122 TA18 R 52 82	142 SA2 b 62 98	162 SA18 r 72 114
0 0 1 1	3 ETX 3 3	23 DC3 13 19	43 LA3 # 23 35	63 LA19 3 33 51	103 TA3 C 43 67	123 TA19 S 53 83	143 SA3 c 63 99	163 SA19 s 73 115
0 1 0 0	4 SDC EOT 4 4	24 DCL DC4 14 20	44 LA4 $ 24 36	64 LA20 4 34 52	104 TA4 D 44 68	124 TA20 T 54 84	144 SA4 d 64 100	164 SA20 t 74 116
0 1 0 1	5 PPC ENQ 5 5	25 PPU NAK 15 21	45 LA5 % 25 37	65 LA21 5 35 53	105 TA5 E 45 69	125 TA21 U 55 85	145 SA5 e 65 101	165 SA21 u 75 117
0 1 1 0	6 ACK 6 6	26 SYN 16 22	46 LA6 & 26 38	66 LA22 6 36 54	106 TA6 F 46 70	126 TA22 V 56 86	146 SA6 f 66 102	166 SA22 v 76 118
0 1 1 1	7 BEL 7 7	27 ETB 17 23	47 LA7 ' 27 39	67 LA23 7 37 55	107 TA7 G 47 71	127 TA23 W 57 87	147 SA7 g 67 103	167 SA23 w 77 119
1 0 0 0	10 GET BS 8 8	30 SPE CAN 18 24	50 LA8 (28 40	70 LA24 8 38 56	110 TA8 H 48 72	130 TA24 X 58 88	150 SA8 h 68 104	170 SA24 x 78 120
1 0 0 1	11 TCT HT 9 9	31 SPD EM 19 25	51 LA9) 29 41	71 LA25 9 39 57	111 TA9 I 49 73	131 TA25 Y 59 89	151 SA9 i 69 105	171 SA25 y 79 121
1 0 1 0	12 LF A 10	32 SUB 1A 26	52 LA10 * 2A 42	72 LA26 : 3A 58	112 TA10 J 4A 74	132 TA26 Z 5A 90	152 SA10 j 6A 106	172 SA26 z 7A 122
1 0 1 1	13 VT B 11	33 ESC 1B 27	53 LA11 + 2B 43	73 LA27 ; 3B 59	113 TA11 K 4B 75	133 TA27 [5B 91	153 SA11 k 6B 107	173 SA27 { 7B 123
1 1 0 0	14 FF C 12	34 FS 1C 28	54 LA12 , 2C 44	74 LA28 < 3C 60	114 TA12 L 4C 76	134 TA28 \ 5C 92	154 SA12 l 6C 108	174 SA28 ¦ 7C 124
1 1 0 1	15 CR D 13	35 GS 1D 29	55 LA13 – 2D 45	75 LA29 = 3D 61	115 TA13 M 4D 77	135 TA29] 5D 93	155 SA13 m 6D 109	175 SA29 } 7D 125
1 1 1 0	16 SO E 14	36 RS 1E 30	56 LA14 . 2E 46	76 LA30 > 3E 62	116 TA14 N 4E 78	136 TA30 ^ 5E 94	156 SA14 n 6E 110	176 SA30 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 LA15 / 2F 47	77 UNL ? 3F 63	117 TA15 O 4F 79	137 UNT – 5F 95	157 SA15 o 6F 111	177 RUBOUT (DEL) 7F 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	

KEY



Tektronix
REF: ANSI STD X3.4-1977
IEEE STD 488.1-1987
ISO STD 646-2973

Appendix B: Reserved Words

This is a list of reserved words for your instrument. Capital letters identify the required minimum spelling.

*CAL
*CLS
*DDT
*ESE
*ESR
*IDN
*LRN
*OPC
*OPT
*PSC
*PUD
*RST
*SRE
*STB
*TRG
*TST
*WAI
A
ABCB
ABOrt
ABSOLUTE
ABSolute
AC
ACBC
ACCM
ACCOMMONMODE
ACDC
ACDCRMS
ACK
ACKMISS
ACKnum
ACQ
ACQDURATION
ACQuire
ACQuisition
ACRMS
ACTION
ACTONEVent
ACTive
ADD
ADDMEAS
ADDNew
ADDR10
ADDR7
ADDRANDDATA
ADDRess
ADVanced
AFG
AHPOSition
ALIas
ALL
ALLAcqs
ALLBits
ALLEv
ALLTHResholds
ALWAYS
AM14
AMPL
AMPLINear
AMPLitude
AMPMode
AMPlITUDE
ANALOg
ANALYZemode
AND
ANNOTate
ANOISEDensity
ANY
ANYERRor
AOFFSet
APOSition
APPID
APPPWR
APPly
ARBitrary
AREA
AREAofcrosssection
ARINC429
ARINC429A
ARROW
ASCIi
ASC_Fmt
ASCii
ASIC
ASOUrce
ASPEctratio
ATRIGger
AUDio
AUNITs
AUTHenticate
AUTO
AUTODim
AUTOETHERnet
AUTOINCrement
AUTORange
AUTOSAVEPITIMEOUT
AUTOSAVEUITIMEOUT
AUTOScale
AUTOSet
AUTOZero
AUTOmatic
AUTOset
DPMAutoset
AUTOset DPMPReset
AUXLevel
AUXiliary
AUXout
AVAILable
AVErage
AVG
AVPOSition
AXPOSition
AYPOSition
Auto
B
B0
B1
B10
B11
B12
B13
B14
B15
B16
B2
B3
B4
B5
B6
B7
B8
B9
BACA
BACKLight
BACKWard
BADDr
BADGE
BANDWidth
BASE
BASETop
BASic
BATHtub
BCR
BCRType
BDIFFBP
BER
BHPOSition
BIAS
BIN
BINary
BIT
BITAMPLITUDE
BITCfgmode
BITDelay
BITEnd
BITHIGH
BITLOW
BITOrder
BITPERiod
BITPcnt
BITRate
BITSTUFFing
BITSTart
BITType
BIT_Nr
BJT
BLACKMANHarris
BLACkmanharris
BLANKingtime
BLOCKDATa
BLOCkid
BM
BMP
BMSGEnd
BNOISE
BN_Fmt
BOFFSet
BOLD
BOOKMark
BOTH
BPACKets
BPOSition
BRGTID
BROADCASTPacket

BROadcast
BRSBit
BSOUrce
BULK
BUNIts
BURSTDETectmethod
BURSTEDGTYPe
BURSTLENGTH
BURSTLatency
BURSTWIDTH
BUS
BUSANDWAVEFORM
BUSTABle
BUSY
BUSx
BVPOSition
BXPOSition
BY
BYPOSition
BYT_Nr
BYT_Or
BYTe
Block
Blt;x>
Blt;x>PSIFIVe
Bpeak
Br
Bus
Bx
C
CALLOUTS
CALLOUTlt;x>
CALibrate
CAN
CAN2X
CANH
CANL
CAPACITANCE
CARDIac
CATalog
CCODe
CCRESUlts
CCYCles
CENTERFREQuency
CENTERFrequency
CGS
CH
CH1
CH2
CH3
CH4
CH5
CH6
CH7
CH8
CHAN1A
CHAN2B
CHANWidth
CHX
CHannel
CHecksum
CHlt;x>
CHlt;x>_DALL
CHlt;x>_Dlt;x>
CHx
CHx_DALL
CHx_Dx
CHx_Dy
CLASSA
CLASSB
CLASSC
CLASSD
CLASs
CLEAR
CLIPping
CLOCK
CLOCKBITSPERCHANNEL
CLOCKFrequency
CLOCKMultiplier
CLOCKRecovery
CLOCKSOUrce
CLOCk
CLOCkSOUrce
CLRESPONSE
CMEThod
COFFSet
COLOR
COLORy
COLor
COLors
COMM
COMMAND
COMMONMode
COMMents
COMPensate
COMPlete
COMPosition
CONDENRG
CONDLOSS
CONDition
CONFIGuration
CONFIGure
CONNECTSTATus
CONNECTion
CONSTAMPlitude
CONSTANTCLOCK
CONSTCLOCKMODe
CONSTant
CONTROLCHARTYPe
CONTROLCODETYPe
CONTRol
CONTinuous
CONTrol
COPy
COUNT
COUNTer
COUNt
COUPling
COUnt
CR
CRC
CRC16
CRC5
CRCERRor
CRCHeader
CRCTrailer
CREATor
CROSSIng
CSActive
CSLevel
CSMode
CSPLIT
CSSource
CSYMVALue
CURRENT
CURRent
CURRentacq
CURSOR1
CURSORlt;x>
CURSor
CURSors
CURVe
CURrent
CUSTOM
CUSTOMLIMITSFile
CUSTOMPROPAgation
CUSTOMTABle
CUSTom
CWD
CYCLE
CYCLEAmp
CYCLEBase
CYCLECount
CYCLEMAX
CYCLEMin
CYCLEMode
CYCLEPKPK
CYCLETop
CYCLEcount
CYCLemode
ChipSelect
Chx_Dx
ClockEdge
Current
D
DADDr
DAMPing
DATA
DATA0
DATA1
DATA2
DATAA
DATAB
DATABITs
DATABits
DATAFORMat
DATAFORmat
DATAFormat
DATAMINUSTHRESHOLD
DATAMINUSTHRESHold
DATAPLUSTHRESHold
DATAPacket
DATAPath
DATARATE
DATARate
DATASource
DATE
DATa
DBCA
DBM
DBMA
DBMV
DBUA
DBUV
DBUW
DC
DCD
DCDC
DCR
DCREJ
DCRType
DDJ
DDR
DDR3
DDRAOS
DDRAOSPERTCK
DDRAOSPERUI
DDRAUS
DDRAUSPERTCK
DDRAUSPERUI
DDRHOLDDIFF
DDRREADWRITE

DDRREADWrite
DDRRead
DDRSETUPDIFF
DDRTCHABS
DDRTCHAVERAGE
DDRTCKAVERAGE
DDRTCLABS
DDRTCLAVERAGE
DDRTDQSCK
DDRTERRMN
DDRTERRN
DDRTJITCC
DDRTJITDUTY
DDRTJITPER
DDRTPST
DDRTRPRE
DDRTWPRE
DDRVIXAC
DDRWrite
DDT
DECIMAL
DECode
DEFect
DEFine
DEFinedby
DEGAUSS
DEGrees
DELAY
DELETEALL
DELEte
DELIMiter
DELTATime
DELay
DELete
DELeteqstring
DELta
DESE
DESKew
DEST
DESTination
DESTinationaddr
DESTinationport
DEVICEDESCMASTERREAd
DEVICEDESCSLAVEREAd
DEVICEType
DHCPbootp
DIAg
DIBYDT
DIDT
DIFF
DIFFerential
DIGGRPlt;x>
DIR
DIRECTPacket
DIRacmodel
DIRect
DIRection
DISLave
DISPLAYUnits
DISParity
DISPlaystat
DISabled
DISplay
DISplaymode
DIVide
DIVisions
DJ
DJDD
DJDIRAC
DMINus
DMSGEnd
DNS
DOFFSet
DOMAINname
DOMain
DONTInclude
DONTcare
DOTsonly
DOUBLE
DPLUs
DPMOVERSHOOT
DPMRIPPLE
DPMTURNOFFTIME
DPMTURNONTIME
DPMUNDERSHOOT
DQDQS
DRIVE
DRIve
DUty
DVBYDT
DVDT
DVM
DWORd
DYNAMIC
DYNAMic
DYNRange
DeltaB
DeltaH
Dlt;x>
EACHCLOCKCYCLE
EBIT
EBITVALue
ECL
ECU
ECUDATa
ECUSENSor
ECUSOURce
EDECAy
EDGE
EDGEIncre
EDGEQUALONE
EDGEQUALTWO
EDGEQUALifier
EDGES
EDGESOURce
EDGESource
EDGEType
EDGElt;x>
EDGe
EFFICIENCY
EIGHTBIT
EIGHTHUNDREdhz
EITHER
EITher
ENABle
ENAbled
ENCdg
ENDPoint
ENDtime
ENET
ENET100
ENET1000
ENGineering
ENHANCED4
ENHANCED8
ENSLave
ENTERSWindow
ENTRDya
ENTRTSTMode
ENTasx
ENVelope
EOF
EOFTYPE
EOP
EOW
EOp
EQUal
EQUation
EQual
EQualNOTEQual
ERISe
ERRORTYPe
ERROr
ERRTYPE
ERRTYPe
ERRType
ERRor
ERRors
ESIBit
ET
ETHERNET
ETHERnet
ETHernet
EVEN
EVENT
EVENTBYTe
EVENTS
EVENTtable
EVERY
EVMsg
EVQty
EXCURsion
EXECute
EXITSWindow
EXPLICITCLOCK
EXPLICITCLOCKMODe
EXTAtten
EXTDBatten
EXTENDed
EXTENDuis
EXTREGREAd
EXTREGWRIte
EXTUnits
EXTended
EXTernal
EXTime
EYEDIAGRAM
EYEHIGH
EYELOW
EYEMASK
EYERENDER
EYERender
EYEWIDTHBER
EYEhistogram
Efficiency1
Efficiency2
Efficiency3
Execute
F1MAG
F2
F3MAG
F4
F8
FACtory
FAIL
FALL
FALLHigh
FALLING
FALLLow
FALLMid
FALLSLEWRATE
FALLTIME

FALSe
FALling
FAST
FASTAcq
FASTERthan
FASTer
FASTframe
FAst
FBD1
FBD2
FBD3
FC1063
FC133
FC2125
FC266
FC4250
FC531
FC8500
FCSError
FD
FDBITS
FDISO
FDNONISO
FFT
FIBREchannel
FIFTY
FIFTYHZ
FIFTy
FIFTyhz
FIFty
FILE
FILEDest
FILEFormat
FILEName
FILESYSTEM
FILESystem
FILTERRANGEFROM
FILTERRANGETO
FILTer
FILTers
FILepath
FIRST
FIVE
FIVEHundred
FIXed
FLATNESS
FLATtop2
FLAg
FLEXRAY
FLEXray
FLIP
FLYBACK
FOLDer
FONT
FORCEDRange
FORCe
FORMERRor
FORMat
FORWard
FOUR
FOURBIT
FOURHUNDREDHZ
FOURHUNDREdhz
FOURHundred
FOURTEENten
FP
FPAnel
FPBinary
FRAME
FRAMECONTrol
FRAMEID
FRAMELENgth
FRAMEType
FRAMEtype
FRAMING
FRAMe
FRAme
FREFerence
FREQ
FREQBYTe
FREQUENCY
FREQ[1-10]Val
FREQ[1-11]Val
FREQuency
FROM
FROMEDGESEARCHDIRect
FROMLevel
FROMSearch
FROMSymbol
FROMedge
FRame
FULL
FULLSPEED
FULLScreen
FULl
FUNCTIONCODETYPe
FUNCTIONCODe
FUNCtion
FUNDCURRent
FW1394BS1600B
FW1394BS400B
FW1394BS800B
GAIN
GAINCROSSOVERFREQ
GAP
GATESOurce
GATEWay
GATing
GAUSSian
GAUSsian
GENIPADDress
GENerator
GETBusch
GETDevch
GETMRdl
GETMWrl
GETPrid
GETSlave
GLOBal
GLOBalref
GLObal
GM
GMT
GPKNOB1
GPKNOB2
GRATicule
GRAticule
GRIDTYPE
GRIDlines
GRId
GROUPDelay
GSLSb
GSMSb
H
HAMMing
HANDSHAKEPacket
HANDSHAKEType
HANNing
HARMONICS
HARMONICSCLass
HARMONICSSOURce
HARMONICSStd
HARMONICSUNits
HAVERSINe
HBArs
HDRCapability
HDRExit
HDRRestart
HEADer
HEIGHTBER
HEIGht
HEX
HEXAgon
HFRej
HI
HID
HIGH
HIGHLEVel
HIGHLevel
HIGHLimit
HIGHPass
HIGHREFVoltage
HIGHTIME
HIGHZ
HIRes
HISTOGRAM
HIStory
HITDMVALue
HITS
HITType
HIVALue
HOLD
HOLDTime
HOLDoff
HOMEDir
HORDer
HORIZontalscale
HORZ
HORZPOS
HORZScale
HORizontal
HOTJoin
HSOURce
HUB
HUNDREDBASETX
HUNdred
HYSTeresis
HYSteresis
Hc
Hmax
I2C
I2S
I3C
IBA2500
IBA_GEN2
ICFACTOR
ID
IDANDDATA
IDFORmat
IDLE
IDLETime
IDLe
IDentifier
IEC
IEEE519
ILEVELAbs
ILEVELPct
IMAGe
IMAGinary
IMDAEFFICIENCY
IMDAHARMONICS
IMDAINPUTCURRENT

IMDAINPUTPOWER
IMDAINPUTVOLTAGE
IMDALINERIPPLE
IMDAPHASORDIAGRAM
IMDAPOWERQUALITY
IMDASWITCHRIPPLE
IMPEDANCE
IMPEDance
IN
INACCBYTe
INCLUDEREFs
INCLUde
INCLude
INDBits
INDEPENDENT
INDUCT
INDUCTANCE
INFILTers
INFInite
INFMAXimum
INFMINimum
INFPersist
INIT
INPUTCAP
INPUTLEVel
INPUTMode
INPUTPOwer
INPUTSOurce
INPUTType
INPut
INRUSH
INRUSHcurrent
INSIDEGreater
INSTR
INSTall
INTENSITy
INTERNal
INTERPRatio
INTERRUPT
INTERnal
INTERp
INTernal
INVERTColor
INVERTNIBble
INVERTed
INVErted
INVerted
INput
INrange
IO
IOUT1SOUrce
IOUT2SOUrce
IOUT3SOUrce
IPADDress
IPHeader
IPOWer
IPVFOUR
IRMS
IRipple
ISCLOCKED
ISOALL
ISOEND
ISOMID
ISOSTART
ISOURce
ISOUrce
ISOchronous
ITALIC
ITALic
ITEM
IVSINTEGRALV
IVSINTV
InputPwr
Inrange
J2
J9
JITTERMODE
JITTERSUMMARY
JITTERSummary
JITTermodel
JPG
JTFBandwidth
KAISERBessel
KAISerbessel
L
L2L
L2LTOL2N
L2N
LABELANDDATA
LABel
LAN
LATCh
LAYout
LDIR
LEFt
LENGth
LENgth
LESSEQual
LESSLimit
LESSThan
LESSthan
LEVELUNIts
LEVel
LF
LFREQuency
LFRej
LIC
LICense
LIMIT
LIN
LINE
LINEAR
LINEAr
LINEFREQUEncy
LINERIPPLE
LINESelected
LINear
LIST
LJ
LOAd
LOCKCenter
LOCKRJ
LOCKRJValue
LOCKSpectrum
LOCk
LOG
LOGARITHM
LOGIC1SOUrce
LOGIC2SOUrce
LOGIC3SOUrce
LOGIC4SOUrce
LOGIC5SOUrce
LOGICPattern
LOGICQUALification
LOGICSource
LOGIc
LOGic
LONGEXTREGREAd
LONGEXTREGWRIte
LOOP
LOOPBandwidth
LORENtz
LOW
LOWDATAMINus
LOWDATAPLUS
LOWERFREQuency
LOWLevel
LOWLimit
LOWPass
LOWREFVoltage
LOWSPEED
LOWTHRESHold
LOWTIME
LOWer
LOWerthreshold
LPDDR3
LRIPPKPK
LRIPRMS
LSB
LSLave
LXI
LogicState
MAC
MACADDRess
MACLENgth
MAG
MAGLOSS
MAGNETICLOSS
MAGNitude
MAGPROPERTY
MAINWindow
MAIn
MANual
MARKER
MARgin
MASK
MASKFail
MASKHit
MASKPass
MASKfile
MASKlt;x>
MASTERADDRess
MASTERREAd
MASTERWRIte
MASTer
MATH
MATHArbfltlt;x>
MATHFFTViewlt;n>
MATHFFTViewlt;x>
MATHFFTx
MATHlt;x>
MATHx
MATlab
MAX
MAXCAS
MAXCycle
MAXFRames
MAXHold
MAXPSRR
MAXPSRRFREQ
MAXREAD
MAXRTURN
MAXSamplerate
MAXTIMe
MAXVoltage
MAXWRITe
MAXimum
MDATA
MDATASpeed
ME
MEAN
MEANAUTOCalculate

MEANhistogram
MEAS1
MEASRange
MEASU
MEASUREAT
MEASUrement
MEASlt;x>
MEASured
MEASurement
MEASx
MEDian
MEDium
MEMORY
MESSaging
METHod
MID
MIDRef
MIL
MIL1553B
MIN
MINCas
MINCycle
MINHold
MINImum
MINMax
MINPSRR
MINPSRRFREQ
MINUI
MINimum
MIXEDASCII
MIXEDHEX
MIXed
MKDir
MODE
MODEhistogram
MODe
MODel
MOREEQua
MOREEQual
MOREEQualINrange
MORELimit
MOREThan
MOREthan
MOSFET
MOUNT
MOVEABLE
MSB
MULTiply
MULtipleframes
Manual
Mathlt;x>
Mathx
N
NACK
NAK
NAME
NAMe
NAND
NANd
NAVigate
NDUTY
NDUTYCYCLE
NDUtY
NDUty
NEGAtive
NEGative
NETWORKCONFig
NEWpass
NEXT
NIBBLECount
NIBBLe
NO
NOCARE
NOISEAdd
NOISErej
NOISe
NOMINALOFFset
NOMinal
NOMinalfreq
NONTRANsition
NONe
NOPARity
NOR
NORESPonse
NORMALColor
NORmal
NOTE
NOVERSHOOT
NOne
NPERIOD
NPJ
NPULSE
NPULSEWIDTH
NR1
NR2
NR3
NRZ
NR_Pt
NTIMES
NTIMes
NULL
NULLFRDynamic
NULLFRStatic
NULl
NUMACQs
NUMACq
NUMAVg
NUMBins
NUMCHANnel
NUMFRAMESACQuired
NUMOFOutputs
NUMOUTputs
NUMSEQuence
NWIDTH
NWIDTTH
NYET
None
OC1
OC12
OC3
OC48
OCCURS
ODD
ODDEVen
OFF
OFF1
OFFSet
ON
ONCE
ONE
ONEOVERDELTATVALUE
ONFAIL
OPPositeas
OPTIMIZation
OPTIMize
OPTion
OR
OUT
OUTFILTers
OUTPUT1
OUTPUT1SOURce
OUTPUT1Type
OUTPUT1VOLTage
OUTPUT2
OUTPUT2SOURce
OUTPUT2Type
OUTPUT2VOLTage
OUTPUT3
OUTPUT3SOURce
OUTPUT3Type
OUTPUT3VOLTage
OUTPUT4
OUTPUT4SOURce
OUTPUT4VOLTage
OUTPUT5
OUTPUT5SOURce
OUTPUT5VOLTage
OUTPUT6
OUTPUT6SOURce
OUTPUT6VOLTage
OUTPUT7
OUTPUT7SOURce
OUTPUT7VOLTage
OUTPUTSOurce
OUTPUTType
OUTPUTlt;n>VOLTage
OUTPut
OUTSIDEGreater
OUTput
OUTrange
OUTside
OVERLoad
OVERRide
OVERlay
OVErlay
Output1Pwr
Output2Pwr
Output3Pwr
P3W3V2I2
P3W3V3I3
P3W4
PACKET
PACKets
PALEtte
PANKNOB
PARItyerror
PARallel
PARity
PASSWord
PATTERNDETECTION
PATTERNLENgth
PATTERNTYPe
PATTern
PAUSE
PAUSEPULSe
PAUSe
PAYLENgth
PAYLength
PAYLoad
PAYload
PCIE_GEN1
PCIE_GEN2
PCIE_GEN3
PCIExpress
PDUTTY
PDUTY
PDUTYCYCLE
PEAK
PEAKCURRent
PEAKVOLTage
PEAKdetect
PERCent

PERFREQ
PERIod
PERSistence
PERSource
PFACtor
PFC
PHASE
PHASECROSSOVERFREQ
PHASENOISE
PHASOR
PHASe
PID
PING
PJ
PK2PK
PK2Pk
PKPK
PLL
PLOT
PLOTData
PLOTVIEWlt;x>
PLOTVIEWx
PLOTViewlt;n>
PLOTViewlt;x>
PLOTlt;x>
PLOTx
PM
PNG
PNJUNCTION
POHCL
POHCM
POHCS
POINT
POINTS
POLARITY
POLarity
POPUlation
PORT
POSITIVe
POST
POSTAMBLE
POSition
POSitive
POVERSHOOT
POWERFACtor
POWERQUALITY
POWERRating
POWERx
POWLINear
POWLOG
POWer
POWerlt;x>
PPD
PPULSE
PPULSEWIDTH
PREAMBLE
PRESET
PRESS
PREViewstate
PREVious
PREamble
PRIMARYTURNs
PRIOD
PROBECOntrol
PROBECal
PROBEFunc
PROFile
PROTOCol
PROTOcol
PRObe
PSIFIVe
PSRR
PT_Fmt
PT_Off
PULSEWIDTh
PULSEWidth
PULse
PWIDTH
PWRFACTOR
PWRFREQ
PWRUpstatus
Permeability
QFACTOR
QString
QStringNR1
QStringQString
QTAG
QTAGGING
QUALifier
Qstring
RADians
RAIL17
RAILNUM
RAMP
RAMPtime
RANDom
RATE100K
RATE10K
RATE10M
RATE115K
RATE11M
RATE125K
RATE12M
RATE13M
RATE14M
RATE153K
RATE15M
RATE16M
RATE19K
RATE1K
RATE1M
RATE20K
RATE250K
RATE25K
RATE2K
RATE2M
RATE300
RATE31K
RATE33K
RATE38K
RATE3M
RATE400K
RATE4K
RATE4M
RATE500K
RATE50K
RATE5M
RATE62K
RATE68K
RATE6M
RATE7M
RATE800K
RATE83K
RATE8M
RATE921K
RATE92K
RATE9K
RATE9M
RATEd
RBW
RBWMode
RCURRent
RDS
RDSON
RDSOn
READ
READFile
READout
READy
REAL
RECAll
RECAllmask
RECORD
RECORDLength
RECOrdlength
RECTANGLE
RECTANGular
RECTangle
RECTangular
REF
REFERence
REFFFTViewlt;n>
REFFFTViewlt;x>
REFFFTx
REFLEVELMode
REFLevel
REFLevels
REFLevelslt;x>
REFMode
REFOUT
REFVOLTAGElt;n>Val
REFVoltage
REFerence
REFlt;x>
REFlt;x>_DALL
REFlt;x>_Dlt;x>
REFx
REFx_Dx
REGOWRIte
REGISTERADDRess
REGREAd
REGWRIte
REGion
REGister
REM
REMote
REName
REPEATStart
REPEATstart
REPOrt
REPWR
REPeating
RERUN
RESAMPLE
RESAMPLErate
RESERVed
RESET
RESOlution
RESPONSECODe
RESPonse
RESPonsetime
RESUME
RESUlt
RESUlts
RESet
RESistance
RF_AVErage
RF_FREQuency
RF_FREQuencylt;x>
RF_MAGnitude
RF_MAGnitudelt;x>
RF_MAXHold

RF_MINHold
RF_NORMal
RF_PHASe
RF_PHASelt;x>
RFvsTime
RI
RIBinary
RIGht
RIO125
RIO250
RIO3125
RIPPLEFREQlt;n>Val
RISE
RISEHigh
RISELow
RISEMid
RISESLEWRATE
RISETIME
RISING
RISe
RISing
RJ
RJDD
RJDIRAC
RMDir
RMS
ROLL
ROLOCATION
ROSc
ROTAtion
RP
RPBinary
RRBDisplaystate
RS232C
RSTDya
RUN
RUNSTop
RUNT
RUNt
RWINADDR
RX
S8B10B
SADDress
SAME
SAMEas
SAMPLERate
SAMPLEpoint
SAMple
SAS12_NOSSC
SAS12_SSC
SAS15_NOSSC
SAS15_SSC
SAS3_NOSSC
SAS3_SSC
SAS6_NOSSC
SAS6_SSC
SATA_GEN1
SATA_GEN2
SATA_GEN3
SAVEIMAGe
SAVEON
SAVEONEVent
SAVEWAVEform
SAVe
SAVemask
SC
SCALERATio
SCALe
SCAle
SCIentific
SCLk
SCREEN
SDATa
SDI
SDIDATA
SDIDATASSM
SDR
SDRBroadcast
SDRDirect
SE
SEARCH
SEARCH1
SEARCHSource
SEARCHTABle
SEARCHlt;x>
SEARCHtotrigger
SEARCHx>
SEARch
SEC1SOURce
SEC1TURNs
SEC2SOURce
SEC2TURNs
SEC3SOURce
SEC3TURNs
SEC4SOURce
SEC4TURNs
SEC5SOURce
SEC5TURNs
SEC6SOURce
SEC6TURNs
SECOND
SECOnds
SECWINDings
SEConds
SEGlt;x>
SEGlt;x>COUNT
SEGlt;y>
SELECTED
SELECTIONtype
SELTrace
SELect
SELected
SENSORADDRess
SENSORECU
SENSORSTATus
SENSor
SENT
SEQSETup
SEQnum
SEQuence
SERVICE
SERVICEMODe
SERVICENAMe
SERnumber
SESsion
SET
SETBrt
SETDya
SETHOLDLogicval
SETHold
SETLevel
SETMrdl
SETMwrl
SETNdya
SETTime
SETUP
SETUp
SEText
SEVEN
SFD
SFPbinary
SHAPE
SHOWAReas
SHOWCRiteria
SHOWEQuation
SHOrt
SHUTdown
SI
SIGNALFREQUEncy
SIGNALTYpe
SIGNALType
SIGNal
SIGnal
SINC
SINE
SINGLE
SINX
SIX
SIXBIT
SIXFIFTyhz
SIXTEENeight
SIXTYHZ
SIXTyhz
SIXty
SIZE
SIZe
SKEW
SLAVEADDRess
SLEEP
SLEWRATEMethod
SLEep
SLOW
SLOWERthan
SLOWer
SLOpe
SMPS
SNAp
SOA
SOAHITSCNT
SOCKETServer
SOF
SOFFRAMENUMber
SOURCE
SOURCEEDGEType
SOURCELIst
SOURCEs
SOURce
SOUrce
SOUrceaddr
SOUrcelt;x>
SOUrceport
SOW
SPACEWIRe
SPAN
SPANABovebw
SPANBELowdc
SPANRBWRatio
SPEC
SPECIALPacket
SPECIALType
SPECTRAL
SPECTRALBUJ
SPECTRUM
SPECTral
SPECViewlt;x>
SPECtral
SPECtrum
SPI
SPLIT
SPLITMODE
SPLit
SPMI

SPREADSheet
SPace
SQUELCH
SQUare
SRIbinary
SRJ
SRPbinary
SRQ
SS
SSC
SSCFREQDEV
SSCMODRATE
SSCPROFILE
SSM
SSPLIT
STALL
STANDard
STANdard
START
STARTBIT
STARTFREQUEncy
STARTFREQuency
STARTFrequency
STARTofframe
STARTtime
STARTup
STARTupnosync
STARt
STATE
STATEBYTe
STATIC
STATIstics
STATUS
STATe
STATic
STATus
STAYSHigh
STAYSLow
STAcked
STAndard
STDDev
STEPRESPONSE
STOP
STOPACQ
STOPAcq
STOPAfter
STOPFREQuency
STOPFrequency
STROBE
STROBESource
STRObe
STRobe
STYle
STandard
SUBADdress
SUBGROUP
SUBNETMask
SUBSF
SUBtract
SUMFrame
SUPPORTBYTe
SUPPress
SUSPEND
SV
SWITCHING
SWITCHINGLOSS
SWITCHINGRIPPLE
SWL
SWLCONFIGType
SWRIPPKPK
SWRIPRMS
SYMB
SYMBol
SYMbol
SYMbols
SYMmetry
SYNC
SYNCFrame
SYNCMODe
SYNCTHRESHold
SYNCfield
SYSTEM
Start
Stop
String
TARGETBER
TBIT
TBITVALue
TCKAVG
TCPHeader
TDM
TDMVALue
TEKEXPonential
TEKSecure
TEMPerature
TENBASET
TENNinety
TERMinal
TERmination
TEST
TESTCONNection
TESTMODe
TESt
TEXT
TF
THDF
THDR
THIRD
THREE
THREEPHASEAUTOset
THREESIXTyhz
THRESHold
THReshold
TICKTIME
TICKTOLerance
TICKs
TIE
TIEHISTOGRAM
TIESPECTRUM
TIETIMETREND
TIMECode
TIMEOUTSIDELEVEL
TIMEOut
TIMEStamp
TIMETREND
TIMINGMode
TIMe
TJBER
TNTRATIO
TO
TOEDGESEARCHDIRect
TOEdge
TOFFENRG
TOFFLOSS
TOKENPacket
TOKENType
TOLERance
TOLerance
TOLevel
TONENRG
TONLOSS
TOOTHGAP
TOP
TOSYmbol
TOTAL
TOTaluptime
TOUCHSCReen
TRACK
TRANSferbusownership
TRANSition
TRANsition
TRAPezoid
TRBit
TRIAngle
TRIGLevel
TRIGger
TRIGgertosearch<!
TRPWR
TRUEPWR
TRUe
TTL
TTLENRG
TTLLOSS
TURN
TURNOFFtime
TURNONTIME
TURNONtime
TWELVEtwelve
TWENTYBIT
TWENtyeighty
TWENtyfive
TWO
TWOBIT
TWOCLOCKCYCLES
TWOFifty
TWOThousand
TX
TXRX
TXRXTHRESHold
TYPE
TYPE1
TYPE2
TYPe
TotalEfficiency
UNDERLine
UNDERline
UNDO
UNEQual
UNINSTALL
UNIQue
UNITINTERVAL
UNITINtervals
UNITIntervals
UNITs
UNIts
UNLock
UNLocked
UNMOUNT
UNWRap
UPDatenow
UPPERFREQuency
UPPer
UPPerthreshold
USB
USB3
USBDevice
USBTmc
USECLockedge
USEClockedge
UTCDELTa
V1X
V2X

VALUe
VALidate
VALue
VARpersist
VBArs
VCESat
VCFACTOR
VDIFFXOVR
VECtors
VERBose
VERTICES
VERTPOS
VERTSCALE
VERTical
VERsion
VGLevel
VIEW
VIEWStyle
VIEWTYpe
VISIBility
VISual
VLEVELAbs
VLEVELPct
VOLTAGE
VOLTage
VOUT1SOUrce
VOUT2SOUrce
VOUT3SOUrce
VRMS
VSOURce
VSOUrce
VTERm
VUNIT
WAITTime
WAKEup
WAKeup
WAVEFORM
WAVEVIEW1
WAVEView
WAVEViewlt;x>
WAVEViewlt;y>
WAVEform
WAVEforms
WAVFrm
WEIGht
WFId
WFMOutpre
WFMTYPe
WHEn
WIDTH
WIDTHBER
WIDth
WINDOW
WINDOWLENgth
WINSCALe
WINdow
WIRing
WIThin
WORD
WORDSel
WORDSize
WRAP
WRITE
WRITEFile
X
XAUI
XAUI_GEN2
XAXIS
XAXISUnits
XFF
XINcr
XPOS
XPOSition
XUNIT
XUNit
XY
XYZ
XYZY
XZEro
XZYZ
YAXIS
YES
YMUlt
YOFf
YPOS
YPOSition
YUNIT
YUNit
YXZX
YZEro
ZERo
ZONe
ZOOM
ZOOMNR1
ZOOMOVERride
ZOOMlt;x>
Zoom
_FREQ_VS_TIME
_MAG_VS_TIME
_PHASE_VS_TIME
_SV_AVErage
_SV_MAXHold
_SV_MINHold
_SV_NORMal
asccurve
block_data
boolean
directory_path
file_path
file_pathdata
harmonics
name
new_working_directory_path
old_file_pathnew_file_path
source_file_pathdestination
sourcefileMASKx
sourcefiledestination
v1
v2
wfmwfm

Appendix C: Factory Defaults

Default Setup

The following table lists the default values for each command.

NOTE. *Find the most up-to-date default values for your instrument and software by performing a TekSecure command, saving the instrument setup and looking at the instrument or setup file.*

Table C-1: Default Values

Item	Description
ACQUIRE:FASTACQ:PALETTE	TEMPERATURE
ACQUIRE:FASTACQ:STATE	0
ACQUIRE:MAGNIVU	0
ACQUIRE:MODE	SAMPLE
ACQUIRE:NUMAVG	16
ACQUIRE:NUMENV	INFINITE
ACQUIRE:STATE	1
ACQUIRE:STOPAFTER	RUNSTOP
ACTONEVENT:ACTION:AUXOUT:STATE	0
ACTONEVENT:ACTION:EMAIL:SETUP:TOADDRESS	"TestString"
ACTONEVENT:ACTION:EMAIL:STATE	0
ACTONEVENT:ACTION:PRINT:STATE	0
ACTONEVENT:ACTION:SAVEIMAGE:STATE	0
ACTONEVENT:ACTION:SAVEWFM:STATE	0
ACTONEVENT:ACTION:SRQ:STATE	0
ACTONEVENT:ACTION:STOPACQ:STATE	1
ACTONEVENT:ACTION:VISUAL:STATE	0
ACTONEVENT:EVENTTYPE	NONE
ACTONEVENT:NUMACQS	1
ACTONEVENT:REPEATCOUNT	1.0000
AFG:AMPLITUDE	500.0000E-3
AFG:ARBITRARY:EMEM:POINTS:ENCDG	ASCII
AFG:FREQUENCY	100.0E+3
AFG:FUNCTION	SINE
AFG:HIGHLEVEL	250.0000E-3
AFG:LEVELPRESET	USER
AFG:LOWLEVEL	-250.0000E-3

Table C-1: Default Values (cont.)

Item	Description
AFG:NOISEADD:PERCENT	0.0E+0
AFG:NOISEADD:STATE	0
AFG:OFFSET	0.0E+0
AFG:OUTPUT:LOAD:IMPEDANCE	HIGHZ
AFG:PERIOD	9.9999999999916E-6
AFG:PHASE	0.0E+0
AFG:PULSE:WIDTH	1.0E-6
AFG:RAMP:SYMMETRY	50.0000
AFG:SQUARE:DUTY	50.0000
ALIAS:STATE	0
APPLICATION:TYPE	POWER
AUXOUT:EDGE	RISING
AUXOUT:SOURCE	ATRIGGER
BUS:B1:AUDIO:BITDELAY	1
BUS:B1:AUDIO:BITORDER	MSB
BUS:B1:AUDIO:CHANNEL:SIZE	24
BUS:B1:AUDIO:CLOCK:POLARITY	RISE
BUS:B1:AUDIO:CLOCK:SOURCE	CH1
BUS:B1:AUDIO:DATA:POLARITY	NORMAL
BUS:B1:AUDIO:DATA:SIZE	24
BUS:B1:AUDIO:DISPLAY:FORMAT	SIGNEDDECIMAL
BUS:B1:AUDIO:FRAME:SIZE	8
BUS:B1:AUDIO:FRAMESYNC:POLARITY	RISE
BUS:B1:AUDIO:FRAMESYNC:SOURCE	CH2
BUS:B1:AUDIO:TYPE	I2S
BUS:B1:AUDIO:WORDSEL:POLARITY	NORMAL
BUS:B1:AUDIO:WORDSEL:SOURCE	CH2
BUS:B1:CAN:BITRATE	500000
BUS:B1:CAN:FD:BITRATE	4000000
BUS:B1:CAN:FD:STANDARD	ISO
BUS:B1:CAN:PROBE	CANH
BUS:B1:CAN:SAMPLEPOINT	50
BUS:B1:CAN:SOURCE	CH1
BUS:B1:CAN:STANDARD	CAN2X
BUS:B1:DISPLAY:FORMAT	HEXADECIMAL
BUS:B1:DISPLAY:TYPE	BUS
BUS:B1:FLEXRAY:BITRATE	10000000

Table C-1: Default Values (cont.)

Item	Description
BUS:B1:FLEXRAY:CHANNEL	A
BUS:B1:FLEXRAY:SIGNAL	BDIFFBP
BUS:B1:FLEXRAY:SOURCE	CH1
BUS:B1:I2C:ADDRESS:RWINCLUDE	0
BUS:B1:I2C:CLOCK:SOURCE	CH1
BUS:B1:I2C:DATA:SOURCE	CH2
BUS:B1:LABEL	"Parallel"
BUS:B1:LIN:BITRATE	19200
BUS:B1:LIN:IDFORMAT	NOPARITY
BUS:B1:LIN:POLARITY	NORMAL
BUS:B1:LIN:SAMPLEPOINT	50
BUS:B1:LIN:SOURCE	CH1
BUS:B1:LIN:STANDARD	V2X
BUS:B1:MIL1553B:POLARITY	NORMAL
BUS:B1:MIL1553B:RESPONSETIME:MAXIMUM	12.0000E-6
BUS:B1:MIL1553B:RESPONSETIME:MINIMUM	4.0000E-6
BUS:B1:MIL1553B:SOURCE	CH1
BUS:B1:PARALLEL:BIT0:SOURCE	D0
BUS:B1:PARALLEL:BIT10:SOURCE	D10
BUS:B1:PARALLEL:BIT11:SOURCE	D11
BUS:B1:PARALLEL:BIT12:SOURCE	D12
BUS:B1:PARALLEL:BIT13:SOURCE	D13
BUS:B1:PARALLEL:BIT14:SOURCE	D14
BUS:B1:PARALLEL:BIT15:SOURCE	D15
BUS:B1:PARALLEL:BIT16:SOURCE	CH1
BUS:B1:PARALLEL:BIT17:SOURCE	CH2
BUS:B1:PARALLEL:BIT18:SOURCE	CH3
BUS:B1:PARALLEL:BIT19:SOURCE	CH4
BUS:B1:PARALLEL:BIT1:SOURCE	D1
BUS:B1:PARALLEL:BIT2:SOURCE	D2
BUS:B1:PARALLEL:BIT3:SOURCE	D3
BUS:B1:PARALLEL:BIT4:SOURCE	D4
BUS:B1:PARALLEL:BIT5:SOURCE	D5
BUS:B1:PARALLEL:BIT6:SOURCE	D6
BUS:B1:PARALLEL:BIT7:SOURCE	D7
BUS:B1:PARALLEL:BIT8:SOURCE	D8
BUS:B1:PARALLEL:BIT9:SOURCE	D9

Table C-1: Default Values (cont.)

Item	Description
BUS:B1:PARALLEL:CLOCK:EDGE	RISING
BUS:B1:PARALLEL:CLOCK:ISCLOCKED	NO
BUS:B1:PARALLEL:CLOCK:SOURCE	CH1
BUS:B1:PARALLEL:WIDTH	16
BUS:B1:POSITION	0.0E+0
BUS:B1:RS232C:BITRATE	9600
BUS:B1:RS232C:DATABITS	8
BUS:B1:RS232C:DELIMITER	LF
BUS:B1:RS232C:DISPLAYMODE	FRAME
BUS:B1:RS232C:PARITY	NONE
BUS:B1:RS232C:POLARITY	NORMAL
BUS:B1:RS232C:RX:SOURCE	OFF
BUS:B1:RS232C:TX:SOURCE	CH1
BUS:B1:SPI:BITORDER	MSB
BUS:B1:SPI:CLOCK:POLARITY	RISE
BUS:B1:SPI:CLOCK:SOURCE	CH1
BUS:B1:SPI:DATA:IN:POLARITY	HIGH
BUS:B1:SPI:DATA:IN:SOURCE	OFF
BUS:B1:SPI:DATA:OUT:POLARITY	HIGH
BUS:B1:SPI:DATA:OUT:SOURCE	CH3
BUS:B1:SPI:DATA:SIZE	8
BUS:B1:SPI:FRAMING	SS
BUS:B1:SPI:IDLETIME	5.0000E-6
BUS:B1:SPI:SELECT:POLARITY	LOW
BUS:B1:SPI:SELECT:SOURCE	CH2
BUS:B1:STATE	0
BUS:B1:TYPE	PARALLEL
BUS:B1:USB:BITRATE	FULL
BUS:B1:USB:PROBE	DIFFERENTIAL
BUS:B1:USB:SOURCE:DIFFERENTIAL	CH1
BUS:B1:USB:SOURCE:DMINUS	CH2
BUS:B1:USB:SOURCE:DPLUS	CH1
BUS:B2:AUDIO:BITDELAY	1
BUS:B2:AUDIO:BITORDER	MSB
BUS:B2:AUDIO:CHANNEL:SIZE	24
BUS:B2:AUDIO:CLOCK:POLARITY	RISE
BUS:B2:AUDIO:CLOCK:SOURCE	CH1

Table C-1: Default Values (cont.)

Item	Description
BUS:B2:AUDIO:DATA:POLARITY	NORMAL
BUS:B2:AUDIO:DATA:SIZE	24
BUS:B2:AUDIO:DISPLAY:FORMAT	SIGNEDDECIMAL
BUS:B2:AUDIO:FRAME:SIZE	8
BUS:B2:AUDIO:FRAMESYNC:POLARITY	RISE
BUS:B2:AUDIO:FRAMESYNC:SOURCE	CH2
BUS:B2:AUDIO:TYPE	I2S
BUS:B2:AUDIO:WORDSEL:POLARITY	NORMAL
BUS:B2:AUDIO:WORDSEL:SOURCE	CH2
BUS:B2:CAN:BITRATE	500000
BUS:B2:CAN:FD:BITRATE	4000000
BUS:B2:CAN:FD:STANDARD	ISO
BUS:B2:CAN:PROBE	CANH
BUS:B2:CAN:SAMPLEPOINT	50
BUS:B2:CAN:SOURCE	CH1
BUS:B2:CAN:STANDARD	CAN2X
BUS:B2:DISPLAY:FORMAT	HEXADECIMAL
BUS:B2:DISPLAY:TYPE	BUS
BUS:B2:FLEXRAY:BITRATE	10000000
BUS:B2:FLEXRAY:CHANNEL	A
BUS:B2:FLEXRAY:SIGNAL	BDIFFBP
BUS:B2:FLEXRAY:SOURCE	CH1
BUS:B2:I2C:ADDRESS:RWINCLUDE	0
BUS:B2:I2C:CLOCK:SOURCE	CH1
BUS:B2:I2C:DATA:SOURCE	CH2
BUS:B2:LABEL	"Parallel"
BUS:B2:LIN:BITRATE	19200
BUS:B2:LIN:IDFORMAT	NOPARITY
BUS:B2:LIN:POLARITY	NORMAL
BUS:B2:LIN:SAMPLEPOINT	50
BUS:B2:LIN:SOURCE	CH1
BUS:B2:LIN:STANDARD	V2X
BUS:B2:MIL1553B:POLARITY	NORMAL
BUS:B2:MIL1553B:RESPONSETIME:MAXIMUM	12.0000E-6
BUS:B2:MIL1553B:RESPONSETIME:MINIMUM	4.0000E-6
BUS:B2:MIL1553B:SOURCE	CH1
BUS:B2:PARALLEL:BIT0:SOURCE	D0

Table C-1: Default Values (cont.)

Item	Description
BUS:B2:PARALLEL:BIT10:SOURCE	D10
BUS:B2:PARALLEL:BIT11:SOURCE	D11
BUS:B2:PARALLEL:BIT12:SOURCE	D12
BUS:B2:PARALLEL:BIT13:SOURCE	D13
BUS:B2:PARALLEL:BIT14:SOURCE	D14
BUS:B2:PARALLEL:BIT15:SOURCE	D15
BUS:B2:PARALLEL:BIT16:SOURCE	CH1
BUS:B2:PARALLEL:BIT17:SOURCE	CH2
BUS:B2:PARALLEL:BIT18:SOURCE	CH3
BUS:B2:PARALLEL:BIT19:SOURCE	CH4
BUS:B2:PARALLEL:BIT1:SOURCE	D1
BUS:B2:PARALLEL:BIT2:SOURCE	D2
BUS:B2:PARALLEL:BIT3:SOURCE	D3
BUS:B2:PARALLEL:BIT4:SOURCE	D4
BUS:B2:PARALLEL:BIT5:SOURCE	D5
BUS:B2:PARALLEL:BIT6:SOURCE	D6
BUS:B2:PARALLEL:BIT7:SOURCE	D7
BUS:B2:PARALLEL:BIT8:SOURCE	D8
BUS:B2:PARALLEL:BIT9:SOURCE	D9
BUS:B2:PARALLEL:CLOCK:EDGE	RISING
BUS:B2:PARALLEL:CLOCK:ISCLOCKED	NO
BUS:B2:PARALLEL:CLOCK:SOURCE	CH1
BUS:B2:PARALLEL:WIDTH	16
BUS:B2:POSITION	0.0E+0
BUS:B2:RS232C:BITRATE	9600
BUS:B2:RS232C:DATABITS	8
BUS:B2:RS232C:DELIMITER	LF
BUS:B2:RS232C:DISPLAYMODE	FRAME
BUS:B2:RS232C:PARITY	NONE
BUS:B2:RS232C:POLARITY	NORMAL
BUS:B2:RS232C:RX:SOURCE	OFF
BUS:B2:RS232C:TX:SOURCE	CH1
BUS:B2:SPI:BITORDER	MSB
BUS:B2:SPI:CLOCK:POLARITY	RISE
BUS:B2:SPI:CLOCK:SOURCE	CH1
BUS:B2:SPI:DATA:IN:POLARITY	HIGH
BUS:B2:SPI:DATA:IN:SOURCE	OFF

Table C-1: Default Values (cont.)

Item	Description
BUS:B2:SPI:DATA:OUT:POLARITY	HIGH
BUS:B2:SPI:DATA:OUT:SOURCE	CH3
BUS:B2:SPI:DATA:SIZE	8
BUS:B2:SPI:FRAMING	SS
BUS:B2:SPI:IDLETIME	5.0000E-6
BUS:B2:SPI:SELECT:POLARITY	LOW
BUS:B2:SPI:SELECT:SOURCE	CH2
BUS:B2:STATE	0
BUS:B2:TYPE	PARALLEL
BUS:B2:USB:BITRATE	FULL
BUS:B2:USB:PROBE	DIFFERENTIAL
BUS:B2:USB:SOURCE:DIFFERENTIAL	CH1
BUS:B2:USB:SOURCE:DMINUS	CH2
BUS:B2:USB:SOURCE:DPLUS	CH1
BUS:LOWERTHRESHOLD:CH1	0.0E+0
BUS:LOWERTHRESHOLD:CH2	0.0E+0
BUS:LOWERTHRESHOLD:CH3	0.0E+0
BUS:LOWERTHRESHOLD:CH4	0.0E+0
BUS:LOWERTHRESHOLD:MATH	0.0E+0
BUS:LOWERTHRESHOLD:REF1	0.0E+0
BUS:LOWERTHRESHOLD:REF2	0.0E+0
BUS:LOWERTHRESHOLD:REF3	0.0E+0
BUS:LOWERTHRESHOLD:REF4	0.0E+0
BUS:UPPERTHRESHOLD:CH1	800.0000E-3
BUS:UPPERTHRESHOLD:CH2	800.0000E-3
BUS:UPPERTHRESHOLD:CH3	800.0000E-3
BUS:UPPERTHRESHOLD:CH4	800.0000E-3
BUS:UPPERTHRESHOLD:MATH	0.0E+0
BUS:UPPERTHRESHOLD:REF1	0.0E+0
BUS:UPPERTHRESHOLD:REF2	0.0E+0
BUS:UPPERTHRESHOLD:REF3	0.0E+0
BUS:UPPERTHRESHOLD:REF4	0.0E+0
CH1:AMPSVIAVOLTS:ENABLE	0
CH1:AMPSVIAVOLTS:FACTOR	10.0000
CH1:BANDWIDTH	1.0000E+9
CH1:COUPLING	DC
CH1:DESKEW	0.0E+0

Table C-1: Default Values (cont.)

Item	Description
CH1:INVERT	0
CH1:LABEL	""
CH1:OFFSET	0.0E+0
CH1:POSITION	0.0E+0
CH1:PROBE:FORCEDRANGE	0.0E+0
CH1:PROBE:GAIN	1.0000
CH1:PROBE:PROPDELAY	0.0E+0
CH1:SCALE	100.0000E-3
CH1:TERMINATION	1.0000E+6
CH1:YUNITS	"V"
CH2:AMPSVIAVOLTS:ENABLE	0
CH2:AMPSVIAVOLTS:FACTOR	10.0000
CH2:BANDWIDTH	1.0000E+9
CH2:COUPLING	DC
CH2:DESKEW	0.0E+0
CH2:INVERT	0
CH2:LABEL	""
CH2:OFFSET	0.0E+0
CH2:POSITION	0.0E+0
CH2:PROBE:FORCEDRANGE	0.0E+0
CH2:PROBE:GAIN	1.0000
CH2:PROBE:PROPDELAY	0.0E+0
CH2:SCALE	100.0000E-3
CH2:TERMINATION	1.0000E+6
CH2:YUNITS	"V"
CH3:AMPSVIAVOLTS:ENABLE	0
CH3:AMPSVIAVOLTS:FACTOR	10.0000
CH3:BANDWIDTH	1.0000E+9
CH3:COUPLING	DC
CH3:DESKEW	0.0E+0
CH3:INVERT	0
CH3:LABEL	""
CH3:OFFSET	0.0E+0
CH3:POSITION	0.0E+0
CH3:PROBE:FORCEDRANGE	0.0E+0
CH3:PROBE:GAIN	1.0000
CH3:PROBE:PROPDELAY	0.0E+0

Table C-1: Default Values (cont.)

Item	Description
CH3:SCALE	100.0000E-3
CH3:TERMINATION	1.0000E+6
CH3:YUNITS	"V"
CH4:AMPSVIAVOLTS:ENABLE	0
CH4:AMPSVIAVOLTS:FACTOR	10.0000
CH4:BANDWIDTH	1.0000E+9
CH4:COUPLING	DC
CH4:DESKEW	0.0E+0
CH4:INVERT	0
CH4:LABEL	""
CH4:OFFSET	0.0E+0
CH4:POSITION	0.0E+0
CH4:PROBE:FORCEDRANGE	0.0E+0
CH4:PROBE:GAIN	1.0000
CH4:PROBE:PROPDELAY	0.0E+0
CH4:SCALE	100.0000E-3
CH4:TERMINATION	1.0000E+6
CH4:YUNITS	"V"
CURSOR:FUNCTION	OFF
CURSOR:HBARS:POSITION1	0.0E+0
CURSOR:HBARS:POSITION2	0.0E+0
CURSOR:HBARS:UNITS	BASE
CURSOR:MODE	INDEPENDENT
CURSOR:SOURCE	AUTO
CURSOR:VBARS:POSITION1	-12.00E-6
CURSOR:VBARS:POSITION2	12.00E-6
CURSOR:VBARS:UNITS	SECONDS
CURSOR:XY:READOUT	RECTANGULAR
CURSOR:XY:RECTANGULAR:X:POSITION1	0.0E+0
CURSOR:XY:RECTANGULAR:X:POSITION2	0.0E+0
CURSOR:XY:RECTANGULAR:Y:POSITION1	0.0E+0
CURSOR:XY:RECTANGULAR:Y:POSITION2	0.0E+0
D0:LABEL	""
D0:POSITION	80.0000E-3
D0:THRESHOLD	1.4000
D10:LABEL	""
D10:POSITION	80.0000E-3

Table C-1: Default Values (cont.)

Item	Description
D10:THRESHOLD	1.4000
D11:LABEL	""
D11:POSITION	80.0000E-3
D11:THRESHOLD	1.4000
D12:LABEL	""
D12:POSITION	80.0000E-3
D12:THRESHOLD	1.4000
D13:LABEL	""
D13:POSITION	80.0000E-3
D13:THRESHOLD	1.4000
D14:LABEL	""
D14:POSITION	80.0000E-3
D14:THRESHOLD	1.4000
D15:LABEL	""
D15:POSITION	80.0000E-3
D15:THRESHOLD	1.4000
D1:LABEL	""
D1:POSITION	80.0000E-3
D1:THRESHOLD	1.4000
D2:LABEL	""
D2:POSITION	80.0000E-3
D2:THRESHOLD	1.4000
D3:LABEL	""
D3:POSITION	80.0000E-3
D3:THRESHOLD	1.4000
D4:LABEL	""
D4:POSITION	80.0000E-3
D4:THRESHOLD	1.4000
D5:LABEL	""
D5:POSITION	80.0000E-3
D5:THRESHOLD	1.4000
D6:LABEL	""
D6:POSITION	80.0000E-3
D6:THRESHOLD	1.4000
D7:LABEL	""
D7:POSITION	80.0000E-3
D7:THRESHOLD	1.4000

Table C-1: Default Values (cont.)

Item	Description
D8:LABEL	""
D8:POSITION	80.0000E-3
D8:THRESHOLD	1.4000
D9:LABEL	""
D9:POSITION	80.0000E-3
D9:THRESHOLD	1.4000
DATA:DESTINATION	REF1
DATA:ENCDG	RIBINARY
DATA:SOURCE	CH1
DATA:START	1
DATA:STOP	10000
DESKEW:DISPLAY	1
DISPLAY:CLOCK	1
DISPLAY:DIGITAL:ACTIVITY	0
DISPLAY:DIGITAL:HEIGHT	MEDIUM
DISPLAY:GRATICULE	FULL
DISPLAY:INTENSITY:BACKLIGHT	HIGH
DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:ENABLE	1
DISPLAY:INTENSITY:BACKLIGHT:AUTODIM:TIME	60
DISPLAY:INTENSITY:GRATICULE	75
DISPLAY:INTENSITY:WAVEFORM	35
DISPLAY:PERSISTENCE	0.0E+0
DISPLAY:STYLE:DOTSONLY	0
DISPLAY:TRIGFREQUENCY	0
DISPLAY:XY	OFF
DISPLAY:XY:WITHYT	0
DVM:AUTORANGE	0
DVM:DISPLAYSTYLE	FULL
DVM:MODE	OFF
DVM:SOURCE	CH1
EMAIL:SETUP:FROMADDRESS	"TestString"
EMAIL:SETUP:HOSTALIASNAME	"TestString"
EMAIL:SETUP:SMTPLOGIN	"TestString"
EMAIL:SETUP:SMTPPORT	25
EMAIL:SETUP:SMTPSERVER	"TestString"
FASTACQ:STATE	0
HEADER	0

Table C-1: Default Values (cont.)

Item	Description
HISTOGRAM:BOXPCNT	20.0000,20.0000,80.0000,80.0000
HISTOGRAM:DISPLAY	LINEAR
HISTOGRAM:MODE	OFF
HISTOGRAM:SOURCE	CH1
HORIZONTAL:DELAY:MODE	1
HORIZONTAL:DELAY:TIME	0.0E+0
HORIZONTAL:POSITION	50.0000
HORIZONTAL:RECORDLENGTH	10000
	4.0000E-6
LOCK	NONE
MARKER:M1:FREQUENCY:ABSOLUTE	0.0E+0
MARKER:M2:FREQUENCY:ABSOLUTE	0.0E+0
MARKER:MANUAL	0
MARKER:PEAK:EXCURSION	30.0000
MARKER:PEAK:MAXIMUM	5
MARKER:PEAK:STATE	1
MARKER:PEAK:THRESHOLD	-50.0000
MARKER:TYPE	ABSOLUTE
MATH:AUTOSCALE	1
MATH:DEFINE	"CH1+CH2"
MATH:HORIZONTAL:POSITION	50.0000
MATH:	4.0000E-6
MATH:HORIZONTAL:UNITS	"s"
MATH:LABEL	""
MATH:SPECTRAL:MAG	DB
MATH:SPECTRAL:WINDOW	HANNING
MATH:TYPE	DUAL
MATH:VERTICAL:POSITION	0.0E+0
MATH:VERTICAL:SCALE	100.0000E-3
MATH:VERTICAL:UNITS	"V"
MATHVAR:VAR1	0.0E+0
MATHVAR:VAR2	0.0E+0
MEASUREMENT:GATING	SCREEN
MEASUREMENT:IMMED:DELAY:DIRECTION	FORWARDS
MEASUREMENT:IMMED:DELAY:EDGE1	RISE
MEASUREMENT:IMMED:DELAY:EDGE2	RISE
MEASUREMENT:IMMED:SOURCE1	CH1

Table C-1: Default Values (cont.)

Item	Description
MEASUREMENT:IMMED:SOURCE2	CH2
MEASUREMENT:IMMED:TYPE	PERIOD
MEASUREMENT:INDICATORS:STATE	OFF
MEASUREMENT:MEAS1:DELAY:DIRECTION	FORWARDS
MEASUREMENT:MEAS1:DELAY:EDGE1	RISE
MEASUREMENT:MEAS1:DELAY:EDGE2	RISE
MEASUREMENT:MEAS1:SOURCE1	CH1
MEASUREMENT:MEAS1:SOURCE2	CH2
MEASUREMENT:MEAS1:STATE	0
MEASUREMENT:MEAS1:TYPE	PERIOD
MEASUREMENT:MEAS2:DELAY:DIRECTION	FORWARDS
MEASUREMENT:MEAS2:DELAY:EDGE1	RISE
MEASUREMENT:MEAS2:DELAY:EDGE2	RISE
MEASUREMENT:MEAS2:SOURCE1	CH1
MEASUREMENT:MEAS2:SOURCE2	CH2
MEASUREMENT:MEAS2:STATE	0
MEASUREMENT:MEAS2:TYPE	PERIOD
MEASUREMENT:MEAS3:DELAY:DIRECTION	FORWARDS
MEASUREMENT:MEAS3:DELAY:EDGE1	RISE
MEASUREMENT:MEAS3:DELAY:EDGE2	RISE
MEASUREMENT:MEAS3:SOURCE1	CH1
MEASUREMENT:MEAS3:SOURCE2	CH2
MEASUREMENT:MEAS3:STATE	0
MEASUREMENT:MEAS3:TYPE	PERIOD
MEASUREMENT:MEAS4:DELAY:DIRECTION	FORWARDS
MEASUREMENT:MEAS4:DELAY:EDGE1	RISE
MEASUREMENT:MEAS4:DELAY:EDGE2	RISE
MEASUREMENT:MEAS4:SOURCE1	CH1
MEASUREMENT:MEAS4:SOURCE2	CH2
MEASUREMENT:MEAS4:STATE	0
MEASUREMENT:MEAS4:TYPE	PERIOD
MEASUREMENT:METHOD	AUTO
MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH	0.0E+0
MEASUREMENT:REFLEVEL:ABSOLUTE:LOW	0.0E+0
MEASUREMENT:REFLEVEL:ABSOLUTE:MID1	0.0E+0
MEASUREMENT:REFLEVEL:ABSOLUTE:MID2	0.0E+0
MEASUREMENT:REFLEVEL:METHOD	PERCENT

Table C-1: Default Values (cont.)

Item	Description
MEASUREMENT:REFLEVEL:PERCENT:HIGH	90.0000
MEASUREMENT:REFLEVEL:PERCENT:LOW	10.0000
MEASUREMENT:REFLEVEL:PERCENT:MID1	50.0000
MEASUREMENT:REFLEVEL:PERCENT:MID2	50.0000
MEASUREMENT:STATISTICS:MODE	ALL
MEASUREMENT:STATISTICS:WEIGHTING	32
MESSAGE:BOX	167,67,167,83
MESSAGE:SHOW	""
MESSAGE:STATE	0
PICTBRIDGE:DATEPRINT	DEFLT
PICTBRIDGE:IDPRINT	OFF
PICTBRIDGE:IMAGESIZE	DEFLT
PICTBRIDGE:PAPERSIZE	DEFLT
PICTBRIDGE:PAPERTYPE	DEFLT
PICTBRIDGE:PRINTQUAL	DEFLT
REF1:HORIZONTAL:DELAY:TIME	16.0000E-6
REF1:	4.0000E-6
REF1:VERTICAL:POSITION	0.0E+0
REF1:VERTICAL:SCALE	100.0000E-3
REF2:HORIZONTAL:DELAY:TIME	16.0000E-6
REF2:	4.0000E-6
REF2:VERTICAL:POSITION	0.0E+0
REF2:VERTICAL:SCALE	100.0000E-3
REF3:HORIZONTAL:DELAY:TIME	16.0000E-6
REF3:	4.0000E-6
REF3:VERTICAL:POSITION	0.0E+0
REF3:VERTICAL:SCALE	100.0000E-3
REF4:HORIZONTAL:DELAY:TIME	16.0000E-6
REF4:	4.0000E-6
REF4:VERTICAL:POSITION	0.0E+0
REF4:VERTICAL:SCALE	100.0000E-3
ROSC:SOURCE	INTERNAL
SAVE:ASSIGN:TYPE	WAVEFORM
SAVE:IMAGE:FILEFORMAT	PNG
SAVE:IMAGE:INKSAVER	1
SAVE:IMAGE:LAYOUT	PORTRAIT
SAVE:WAVEFORM	WFM

Table C-1: Default Values (cont.)

Item	Description
SAVE:WAVEFORM:GATING	NONE
SEARCH:SEARCH1:STATE	0
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:HIVALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET	0
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:VALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:AUDIO:DATA:WORD	EITHER
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION	NOCARE
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:OFFSET	0
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:FD:BRSBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:FD:ESIBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:FRAMETYPE	DATA
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE	STANDARD
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET	-1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE	STATIC
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE	CRCHEADER
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE	NORMAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID	"XXXXXXXXXXX"

Table C-1: Default Values (cont.)

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE	ADDR7
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:ADDRESS:TYPE	USER
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:CONDITION	START
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION	NOCARE
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:I2C:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:CONDITION	SYNCFIELD
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:ERRTYPE	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:HIVALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:ADDRESS:VALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:COUNT	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:SUBADDRESS	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:COMMAND:TRBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:CONDITION	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:DATA:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:DATA:VALUE	"XXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:ERRTYPE	PARITY
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:HIVALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:ADDRESS:VALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BCR	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:BUSY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:DBCA	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:INSTR	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:ME	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SRQ	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:SUBSF	X

Table C-1: Default Values (cont.)

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:BIT:TF	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:STATUS:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:TIME:LESSLIMIT	4.0000E-6
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:TIME:MORELIMIT	12.0000E-6
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:MIL1553B:TIME:QUALIFIER	MORETHAN
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:PARALLEL:VALUE	"XXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:CONDITION	TXSTART
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:RX:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:RX:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:TX:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:RS232C:TX:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:CONDITION	SS
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:DATA:MISO:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:DATA:MOSI:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:SPI:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:CONDITION	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:OFFSET	-1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:TYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE	"XXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:ERRTYPE	PID
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:SPECIALTYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B1:USB:TOKENTYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:HIVALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:OFFSET	0
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:VALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:AUDIO:DATA:WORD	EITHER
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:CONDITION	SOF

Table C-1: Default Values (cont.)

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:DIRECTION	NOCARE
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:OFFSET	0
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:FD:BRSBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:FD:ESIBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:FRAMETYPE	DATA
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:IDENTIFIER:MODE	STANDARD
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:CAN:IDENTIFIER:VALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:CONDITION	SOF
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:HIVALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:VALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:DATA:OFFSET	-1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:EOFTYPE	STATIC
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:ERRTYPE	CRCHEADER
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:HIVALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:VALUE	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:FRAMETYPE	NORMAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:HEADER:CRC	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:HEADER:CYCLECOUNT	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:HEADER:FRAMEID	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:HEADER:INDBITS	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:FLEXRAY:HEADER:PAYLENGTH	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:ADDRESS:MODE	ADDR7
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:ADDRESS:TYPE	USER
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:ADDRESS:VALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:CONDITION	START
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:DATA:DIRECTION	NOCARE
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:I2C:DATA:VALUE	"XXXXXXXX"

Table C-1: Default Values (cont.)

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:CONDITION	SYNCFIELD
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:ERRTYPE	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:LIN:IDENTIFIER:VALUE	"XXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:ADDRESS:HIVALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:ADDRESS:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:ADDRESS:VALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:COUNT	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:SUBADDRESS	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:COMMAND:TRBIT	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:CONDITION	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:DATA:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:DATA:VALUE	"XXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:ERRTYPE	PARITY
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:ADDRESS:HIVALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:ADDRESS:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:ADDRESS:VALUE	"XXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:BCR	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:BUSY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:DBCA	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:INSTR	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:ME	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:SRQ	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:SUBSF	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:BIT:TF	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:STATUS:PARITY	X
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:TIME:LESSLIMIT	4.0000E-6
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:TIME:MORELIMIT	12.0000E-6
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:MIL1553B:TIME:QUALIFIER	MORETHAN
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:PARALLEL:VALUE	"XXXXXXXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:CONDITION	TXSTART
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:RX:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:RX:DATA:VALUE	"XXXXXXXX"

Table C-1: Default Values (cont.)

Item	Description
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:TX:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:RS232C:TX:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:CONDITION	SS
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:DATA:MISO:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:DATA:MOSI:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:SPI:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:ADDRESS:HIVALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:ADDRESS:VALUE	"XXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:CONDITION	SYNC
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:DATA:HIVALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:DATA:OFFSET	-1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:DATA:SIZE	1
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:DATA:TYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:DATA:VALUE	"XXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:ENDPOINT:VALUE	"XXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:ERRTYPE	PID
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:HANDSHAKETYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:QUALIFIER	EQUAL
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:SOFFRAMENUMBER	"XXXXXXXXXXX"
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:SPECIALTYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:B2:USB:TOKENTYPE	ANY
SEARCH:SEARCH1:TRIGGER:A:BUS:SOURCE	B1
SEARCH:SEARCH1:TRIGGER:A:EDGE:SLOPE	RISE
SEARCH:SEARCH1:TRIGGER:A:EDGE:SOURCE	CH1
SEARCH:SEARCH1:TRIGGER:A:LEVEL	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:CH4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:MATH	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LEVEL:REF4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:FUNCTION	AND
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH1	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH2	X

Table C-1: Default Values (cont.)

Item	Description
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH3	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CH4	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CLOCK:EDGE	RISE
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE	NONE
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D0	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D1	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D10	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D11	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D12	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D13	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D14	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D15	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D2	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D3	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D4	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D5	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D6	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D7	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D8	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:D9	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:MATH	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF1	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF2	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF3	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:INPUT:REF4	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D0	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D1	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D10	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D11	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D12	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D13	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D14	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D15	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D2	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D3	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D4	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D5	X

Table C-1: Default Values (cont.)

Item	Description
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D6	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D7	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D8	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:INPUT:D9	X
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:WHEN	TRUE
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:CH4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:MATH	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOGIC:THRESHOLD:REF4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:CH4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:MATH	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:LOWERTHRESHOLD:REF4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:HIGHLIMIT	12.0000E-9
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:LOWLIMIT	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:POLARITY	POSITIVE
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WHEN	LESSTHAN
SEARCH:SEARCH1:TRIGGER:A:PULSEWIDTH:WIDTH	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:RUNT:POLARITY	POSITIVE
SEARCH:SEARCH1:TRIGGER:A:RUNT:WHEN	OCCURS
SEARCH:SEARCH1:TRIGGER:A:RUNT:WIDTH	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:EDGE	RISE
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:SOURCE	CH1
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:CLOCK:THRESHOLD	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:SOURCE	NONE

Table C-1: Default Values (cont.)

Item	Description
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:DATA:THRESHOLD	9.91E+37
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:HOLDTIME	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:SETTIME	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:CH4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:MATH	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF1	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF2	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF3	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:SETHOLD:THRESHOLD:REF4	0.0E+0
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:POLARITY	STAYSHIGH
SEARCH:SEARCH1:TRIGGER:A:TIMEOUT:TIME	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:TRANSITION:DELTATIME	8.0000E-9
SEARCH:SEARCH1:TRIGGER:A:TRANSITION:POLARITY	POSITIVE
SEARCH:SEARCH1:TRIGGER:A:TRANSITION:WHEN	SLOWER
SEARCH:SEARCH1:TRIGGER:A:TYPE	EDGE
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH1	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH2	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH3	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:CH4	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:MATH	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF1	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF2	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF3	800.0000E-3
SEARCH:SEARCH1:TRIGGER:A:UPPERTHRESHOLD:REF4	800.0000E-3
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:DATa:OFFSet	0
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:FD:BRSBIT	X
SEARCH:SEARCH<x>:TRIGGER:A:BUS:B<x>:CAN:FD:ESIBIT	X
TRIGGER:A:BUS:B1:AUDIO:CONDITION	SOF
TRIGGER:A:BUS:B1:AUDIO:DATA:HIVALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B1:AUDIO:DATA:OFFSET	0
TRIGGER:A:BUS:B1:AUDIO:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:AUDIO:DATA:VALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B1:AUDIO:DATA:WORD	EITHER
TRIGGER:A:BUS:B1:CAN:CONDITION	SOF

Table C-1: Default Values (cont.)

Item	Description
TRIGGER:A:BUS:B1:CAN:DATA:DIRECTION	NOCARE
TRIGGER:A:BUS:B1:CAN:DATA:OFFSET	0
TRIGGER:A:BUS:B1:CAN:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:CAN:DATA:SIZE	1
TRIGGER:A:BUS:B1:CAN:DATA:START	0.0E+0
TRIGGER:A:BUS:B1:CAN:FD:BRSBIT	X
TRIGGER:A:BUS:B1:CAN:FD:ESIBIT	X
TRIGGER:A:BUS:B1:CAN:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:CAN:FRAMETYPE	DATA
TRIGGER:A:BUS:B1:CAN:IDENTIFIER:MODE	STANDARD
TRIGGER:A:BUS:B1:CAN:IDENTIFIER:VALUE	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:CONDITION	SOF
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:HIVALUE	"XXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:FLEXRAY:CYCLECOUNT:VALUE	"XXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:DATA:OFFSET	-1
TRIGGER:A:BUS:B1:FLEXRAY:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:FLEXRAY:DATA:SIZE	1
TRIGGER:A:BUS:B1:FLEXRAY:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:EOFTYPE	STATIC
TRIGGER:A:BUS:B1:FLEXRAY:ERRTYPE	CRCHEADER
TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:HIVALUE	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:FLEXRAY:FRAMEID:VALUE	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:FRAMETYPE	NORMAL
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CRC	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:CYCLECOUNT	"XXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:FRAMEID	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:INDBITS	"XXXXX"
TRIGGER:A:BUS:B1:FLEXRAY:HEADER:PAYLENGTH	"XXXXXXX"
TRIGGER:A:BUS:B1:I2C:ADDRESS:MODE	ADDR7
TRIGGER:A:BUS:B1:I2C:ADDRESS:TYPE	USER
TRIGGER:A:BUS:B1:I2C:ADDRESS:VALUE	"XXXXXXX"
TRIGGER:A:BUS:B1:I2C:CONDITION	START
TRIGGER:A:BUS:B1:I2C:DATA:DIRECTION	NOCARE
TRIGGER:A:BUS:B1:I2C:DATA:SIZE	1

Table C-1: Default Values (cont.)

Item	Description
TRIGGER:A:BUS:B1:I2C:DATA:START	0.0E+0
TRIGGER:A:BUS:B1:I2C:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:LIN:CONDITION	SYNCFIELD
TRIGGER:A:BUS:B1:LIN:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:LIN:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:LIN:DATA:SIZE	1
TRIGGER:A:BUS:B1:LIN:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:LIN:ERRTYPE	SYNC
TRIGGER:A:BUS:B1:LIN:IDENTIFIER:VALUE	"XXXXXX"
TRIGGER:A:BUS:B1:PARALLEL:VALUE	"XXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B1:RS232C:CONDITION	TXSTART
TRIGGER:A:BUS:B1:RS232C:RX:DATA:SIZE	1
TRIGGER:A:BUS:B1:RS232C:RX:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:RS232C:TX:DATA:SIZE	1
TRIGGER:A:BUS:B1:RS232C:TX:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:SPI:CONDITION	SS
TRIGGER:A:BUS:B1:SPI:DATA:IN:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:SPI:DATA:OUT:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:SPI:DATA:SIZE	1
TRIGGER:A:BUS:B1:SPI:DATA:START	0.0E+0
TRIGGER:A:BUS:B1:USB:ADDRESS:HIVALUE	"XXXXXXX"
TRIGGER:A:BUS:B1:USB:ADDRESS:VALUE	"XXXXXXX"
TRIGGER:A:BUS:B1:USB:CONDITION	SYNC
TRIGGER:A:BUS:B1:USB:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:USB:DATA:OFFSET	-1
TRIGGER:A:BUS:B1:USB:DATA:SIZE	1
TRIGGER:A:BUS:B1:USB:DATA:TYPE	ANY
TRIGGER:A:BUS:B1:USB:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B1:USB:ENDPOINT:VALUE	"XXXX"
TRIGGER:A:BUS:B1:USB:ERRTYPE	PID
TRIGGER:A:BUS:B1:USB:HANDSHAKETYPE	ANY
TRIGGER:A:BUS:B1:USB:QUALIFIER	EQUAL
TRIGGER:A:BUS:B1:USB:SOFFRAMENUMBER	"XXXXXXXXXXX"
TRIGGER:A:BUS:B1:USB:SPECIALTYPE	ANY
TRIGGER:A:BUS:B1:USB:TOKENTYPE	ANY
TRIGGER:A:BUS:B2:AUDIO:CONDITION	SOF
TRIGGER:A:BUS:B2:AUDIO:DATA:HIVALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"

Table C-1: Default Values (cont.)

Item	Description
TRIGGER:A:BUS:B2:AUDIO:DATA:OFFSET	0
TRIGGER:A:BUS:B2:AUDIO:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:AUDIO:DATA:VALUE	"XXXXXXXXXXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B2:AUDIO:DATA:WORD	EITHER
TRIGGER:A:BUS:B2:CAN:CONDITION	SOF
TRIGGER:A:BUS:B2:CAN:DATA:DIRECTION	NOCARE
TRIGGER:A:BUS:B2:CAN:DATA:OFFSET	0
TRIGGER:A:BUS:B2:CAN:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:CAN:DATA:SIZE	1
TRIGGER:A:BUS:B2:CAN:DATA:START	0.0E+0
TRIGGER:A:BUS:B2:CAN:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:CAN:FD:BRSBUT	X
TRIGGER:A:BUS:B2:CAN:FD:ESIBIT	X
TRIGGER:A:BUS:B2:CAN:FRAMETYPE	DATA
TRIGGER:A:BUS:B2:CAN:IDENTIFIER:MODE	STANDARD
TRIGGER:A:BUS:B2:CAN:IDENTIFIER:VALUE	"XXXXXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:CONDITION	SOF
TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:HIVALUE	"XXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:FLEXRAY:CYCLECOUNT:VALUE	"XXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:DATA:OFFSET	-1
TRIGGER:A:BUS:B2:FLEXRAY:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:FLEXRAY:DATA:SIZE	1
TRIGGER:A:BUS:B2:FLEXRAY:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:EOFTYPE	STATIC
TRIGGER:A:BUS:B2:FLEXRAY:ERRTYPE	CRCHEADER
TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:HIVALUE	"XXXXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:FLEXRAY:FRAMEID:VALUE	"XXXXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:FRAMETYPE	NORMAL
TRIGGER:A:BUS:B2:FLEXRAY:HEADER:CRC	"XXXXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:HEADER:CYCLECOUNT	"XXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:HEADER:FRAMEID	"XXXXXXXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:HEADER:INDBITS	"XXXXX"
TRIGGER:A:BUS:B2:FLEXRAY:HEADER:PAYLENGTH	"XXXXXXX"
TRIGGER:A:BUS:B2:I2C:ADDRESS:MODE	ADDR7

Table C-1: Default Values (cont.)

Item	Description
TRIGGER:A:BUS:B2:I2C:ADDRESS:TYPE	USER
TRIGGER:A:BUS:B2:I2C:ADDRESS:VALUE	"XXXXXXX"
TRIGGER:A:BUS:B2:I2C:CONDITION	START
TRIGGER:A:BUS:B2:I2C:DATA:DIRECTION	NOCARE
TRIGGER:A:BUS:B2:I2C:DATA:SIZE	1
TRIGGER:A:BUS:B2:I2C:DATA:START	0.0E+0
TRIGGER:A:BUS:B2:I2C:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:LIN:CONDITION	SYNCFIELD
TRIGGER:A:BUS:B2:LIN:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:LIN:DATA:QUALIFIER	EQUAL
TRIGGER:A:BUS:B2:LIN:DATA:SIZE	1
TRIGGER:A:BUS:B2:LIN:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:LIN:ERRTYPE	SYNC
TRIGGER:A:BUS:B2:LIN:IDENTIFIER:VALUE	"XXXXXX"
TRIGGER:A:BUS:B2:PARALLEL:VALUE	"XXXXXXXXXXXXXXXX"
TRIGGER:A:BUS:B2:RS232C:CONDITION	TXSTART
TRIGGER:A:BUS:B2:RS232C:RX:DATA:SIZE	1
TRIGGER:A:BUS:B2:RS232C:RX:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:RS232C:TX:DATA:SIZE	1
TRIGGER:A:BUS:B2:RS232C:TX:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:SPI:CONDITION	SS
TRIGGER:A:BUS:B2:SPI:DATA:IN:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:SPI:DATA:OUT:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:SPI:DATA:SIZE	1
TRIGGER:A:BUS:B2:SPI:DATA:START	0.0E+0
TRIGGER:A:BUS:B2:USB:ADDRESS:HIVALUE	"XXXXXXX"
TRIGGER:A:BUS:B2:USB:ADDRESS:VALUE	"XXXXXXX"
TRIGGER:A:BUS:B2:USB:CONDITION	SYNC
TRIGGER:A:BUS:B2:USB:DATA:HIVALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:USB:DATA:OFFSET	-1
TRIGGER:A:BUS:B2:USB:DATA:SIZE	1
TRIGGER:A:BUS:B2:USB:DATA:TYPE	ANY
TRIGGER:A:BUS:B2:USB:DATA:VALUE	"XXXXXXXX"
TRIGGER:A:BUS:B2:USB:ENDPOINT:VALUE	"XXXX"
TRIGGER:A:BUS:B2:USB:ERRTYPE	PID
TRIGGER:A:BUS:B2:USB:HANDSHAKETYPE	ANY
TRIGGER:A:BUS:B2:USB:QUALIFIER	EQUAL

Table C-1: Default Values (cont.)

Item	Description
TRIGGER:A:BUS:B2:USB:SOFFRAMENUMBER	"XXXXXXXXXXX"
TRIGGER:A:BUS:B2:USB:SPECIALTYPE	ANY
TRIGGER:A:BUS:B2:USB:TOKENTYPE	ANY
TRIGGER:A:BUS:B<x>:CAN:DATa:OFFSet	0
TRIGGER:A:BUS:B<x>:CAN:FD:BRSBIT	X
TRIGGER:A:BUS:B<x>:CAN:FD:ESIBIT	X
TRIGGER:A:BUS:SOURCE	B1
TRIGGER:A:EDGE:COUPLING	DC
TRIGGER:A:EDGE:SLOPE	RISE
TRIGGER:A:EDGE:SOURCE	CH1
TRIGGER:A:HOLDOFF:TIME	20.0000E-9
TRIGGER:A:LEVEL	0.0E+0
TRIGGER:A:LEVEL:CH1	0.0E+0
TRIGGER:A:LEVEL:CH2	0.0E+0
TRIGGER:A:LEVEL:CH3	0.0E+0
TRIGGER:A:LEVEL:CH4	0.0E+0
TRIGGER:A:LEVEL:D0	1.4000
TRIGGER:A:LEVEL:D1	1.4000
TRIGGER:A:LEVEL:D10	1.4000
TRIGGER:A:LEVEL:D11	1.4000
TRIGGER:A:LEVEL:D12	1.4000
TRIGGER:A:LEVEL:D13	1.4000
TRIGGER:A:LEVEL:D14	1.4000
TRIGGER:A:LEVEL:D15	1.4000
TRIGGER:A:LEVEL:D2	1.4000
TRIGGER:A:LEVEL:D3	1.4000
TRIGGER:A:LEVEL:D4	1.4000
TRIGGER:A:LEVEL:D5	1.4000
TRIGGER:A:LEVEL:D6	1.4000
TRIGGER:A:LEVEL:D7	1.4000
TRIGGER:A:LEVEL:D8	1.4000
TRIGGER:A:LEVEL:D9	1.4000
TRIGGER:A:LOGIC:CLASS	LOGIC
TRIGGER:A:LOGIC:FUNCTION	AND
TRIGGER:A:LOGIC:INPUT:CH1	X
TRIGGER:A:LOGIC:INPUT:CH2	X
TRIGGER:A:LOGIC:INPUT:CH3	X

Table C-1: Default Values (cont.)

Item	Description
TRIGGER:A:LOGIC:INPUT:CH4	X
TRIGGER:A:LOGIC:INPUT:CLOCK:EDGE	RISE
TRIGGER:A:LOGIC:INPUT:CLOCK:SOURCE	NONE
TRIGGER:A:LOGIC:INPUT:D0	X
TRIGGER:A:LOGIC:INPUT:D1	X
TRIGGER:A:LOGIC:INPUT:D10	X
TRIGGER:A:LOGIC:INPUT:D11	X
TRIGGER:A:LOGIC:INPUT:D12	X
TRIGGER:A:LOGIC:INPUT:D13	X
TRIGGER:A:LOGIC:INPUT:D14	X
TRIGGER:A:LOGIC:INPUT:D15	X
TRIGGER:A:LOGIC:INPUT:D2	X
TRIGGER:A:LOGIC:INPUT:D3	X
TRIGGER:A:LOGIC:INPUT:D4	X
TRIGGER:A:LOGIC:INPUT:D5	X
TRIGGER:A:LOGIC:INPUT:D6	X
TRIGGER:A:LOGIC:INPUT:D7	X
TRIGGER:A:LOGIC:INPUT:D8	X
TRIGGER:A:LOGIC:INPUT:D9	X
TRIGGER:A:LOGIC:PATTERN:DELTATIME	8.0000E-9
TRIGGER:A:LOGIC:PATTERN:WHEN	TRUE
TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT	8.0000E-9
TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT	8.0000E-9
TRIGGER:A:LOGIC:THRESHOLD:CH1	0.0E+0
TRIGGER:A:LOGIC:THRESHOLD:CH2	0.0E+0
TRIGGER:A:LOGIC:THRESHOLD:CH3	0.0E+0
TRIGGER:A:LOGIC:THRESHOLD:CH4	0.0E+0
TRIGGER:A:LOGIC:THRESHOLD:D0	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D1	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D10	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D11	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D12	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D13	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D14	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D15	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D2	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D3	1.4000

Table C-1: Default Values (cont.)

Item	Description
TRIGGER:A:LOGIC:THRESHOLD:D4	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D5	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D6	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D7	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D8	1.4000
TRIGGER:A:LOGIC:THRESHOLD:D9	1.4000
TRIGGER:A:LOWERTHRESHOLD:CH1	0.0E+0
TRIGGER:A:LOWERTHRESHOLD:CH2	0.0E+0
TRIGGER:A:LOWERTHRESHOLD:CH3	0.0E+0
TRIGGER:A:LOWERTHRESHOLD:CH4	0.0E+0
TRIGGER:A:LOWERTHRESHOLD:D0	1.4000
TRIGGER:A:LOWERTHRESHOLD:D1	1.4000
TRIGGER:A:LOWERTHRESHOLD:D10	1.4000
TRIGGER:A:LOWERTHRESHOLD:D11	1.4000
TRIGGER:A:LOWERTHRESHOLD:D12	1.4000
TRIGGER:A:LOWERTHRESHOLD:D13	1.4000
TRIGGER:A:LOWERTHRESHOLD:D14	1.4000
TRIGGER:A:LOWERTHRESHOLD:D15	1.4000
TRIGGER:A:LOWERTHRESHOLD:D2	1.4000
TRIGGER:A:LOWERTHRESHOLD:D3	1.4000
TRIGGER:A:LOWERTHRESHOLD:D4	1.4000
TRIGGER:A:LOWERTHRESHOLD:D5	1.4000
TRIGGER:A:LOWERTHRESHOLD:D6	1.4000
TRIGGER:A:LOWERTHRESHOLD:D7	1.4000
TRIGGER:A:LOWERTHRESHOLD:D8	1.4000
TRIGGER:A:LOWERTHRESHOLD:D9	1.4000
TRIGGER:A:MODE	AUTO
TRIGGER:A:PULSE:CLASS	WIDTH
TRIGGER:A:PULSEWIDTH:HIGHLIMIT	12.0000E-9
TRIGGER:A:PULSEWIDTH:LOWLIMIT	8.0000E-9
TRIGGER:A:PULSEWIDTH:POLARITY	POSITIVE
TRIGGER:A:PULSEWIDTH:WHEN	LESSTHAN
TRIGGER:A:PULSEWIDTH:WIDTH	8.0000E-9
TRIGGER:A:RUNT:POLARITY	POSITIVE
TRIGGER:A:RUNT:WHEN	OCCURS
TRIGGER:A:RUNT:WIDTH	8.0000E-9
TRIGGER:A:SETHOLD:CLOCK:EDGE	RISE

Table C-1: Default Values (cont.)

Item	Description
TRIGGER:A:SETHOLD:CLOCK:SOURCE	CH1
TRIGGER:A:SETHOLD:CLOCK:THRESHOLD	0.0E+0
TRIGGER:A:SETHOLD:DATA:SOURCE	NONE
TRIGGER:A:SETHOLD:DATA:THRESHOLD	9.91E+37
TRIGGER:A:SETHOLD:HOLDTIME	8.0000E-9
TRIGGER:A:SETHOLD:SETTIME	8.0000E-9
TRIGGER:A:SETHOLD:THRESHOLD:CH1	0.0E+0
TRIGGER:A:SETHOLD:THRESHOLD:CH2	0.0E+0
TRIGGER:A:SETHOLD:THRESHOLD:CH3	0.0E+0
TRIGGER:A:SETHOLD:THRESHOLD:CH4	0.0E+0
TRIGGER:A:SETHOLD:THRESHOLD:D0	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D1	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D10	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D11	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D12	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D13	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D14	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D15	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D2	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D3	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D4	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D5	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D6	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D7	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D8	1.4000
TRIGGER:A:SETHOLD:THRESHOLD:D9	1.4000
TRIGGER:A:TIMEOUT:POLARITY	STAYSHIGH
TRIGGER:A:TIMEOUT:TIME	8.0000E-9
TRIGGER:A:TRANSITION:DELTATIME	8.0000E-9
TRIGGER:A:TRANSITION:POLARITY	POSITIVE
TRIGGER:A:TRANSITION:WHEN	SLOWER
TRIGGER:A:TYPE	EDGE
TRIGGER:A:UPPERTHRESHOLD:CH1	800.0000E-3
TRIGGER:A:UPPERTHRESHOLD:CH2	800.0000E-3
TRIGGER:A:UPPERTHRESHOLD:CH3	800.0000E-3
TRIGGER:A:UPPERTHRESHOLD:CH4	800.0000E-3
TRIGGER:B:BY	TIME

Table C-1: Default Values (cont.)

Item	Description
TRIGGER:B:EDGE:COUPLING	DC
TRIGGER:B:EDGE:SLOPE	RISE
TRIGGER:B:EDGE:SOURCE	CH1
TRIGGER:B:EVENTS:COUNT	1
TRIGGER:B:LEVEL	0.0E+0
TRIGGER:B:LEVEL:CH1	0.0E+0
TRIGGER:B:LEVEL:CH2	0.0E+0
TRIGGER:B:LEVEL:CH3	0.0E+0
TRIGGER:B:LEVEL:CH4	0.0E+0
TRIGGER:B:LEVEL:D0	1.4000
TRIGGER:B:LEVEL:D1	1.4000
TRIGGER:B:LEVEL:D10	1.4000
TRIGGER:B:LEVEL:D11	1.4000
TRIGGER:B:LEVEL:D12	1.4000
TRIGGER:B:LEVEL:D13	1.4000
TRIGGER:B:LEVEL:D14	1.4000
TRIGGER:B:LEVEL:D15	1.4000
TRIGGER:B:LEVEL:D2	1.4000
TRIGGER:B:LEVEL:D3	1.4000
TRIGGER:B:LEVEL:D4	1.4000
TRIGGER:B:LEVEL:D5	1.4000
TRIGGER:B:LEVEL:D6	1.4000
TRIGGER:B:LEVEL:D7	1.4000
TRIGGER:B:LEVEL:D8	1.4000
TRIGGER:B:LEVEL:D9	1.4000
TRIGGER:B:LOWERTHRESHOLD:CH1	0.0E+0
TRIGGER:B:LOWERTHRESHOLD:CH2	0.0E+0
TRIGGER:B:LOWERTHRESHOLD:CH3	0.0E+0
TRIGGER:B:LOWERTHRESHOLD:CH4	0.0E+0
TRIGGER:B:LOWERTHRESHOLD:D0	1.4000
TRIGGER:B:LOWERTHRESHOLD:D1	1.4000
TRIGGER:B:LOWERTHRESHOLD:D10	1.4000
TRIGGER:B:LOWERTHRESHOLD:D11	1.4000
TRIGGER:B:LOWERTHRESHOLD:D12	1.4000
TRIGGER:B:LOWERTHRESHOLD:D13	1.4000
TRIGGER:B:LOWERTHRESHOLD:D14	1.4000
TRIGGER:B:LOWERTHRESHOLD:D15	1.4000

Table C-1: Default Values (cont.)

Item	Description
TRIGGER:B:LOWERTHRESHOLD:D2	1.4000
TRIGGER:B:LOWERTHRESHOLD:D3	1.4000
TRIGGER:B:LOWERTHRESHOLD:D4	1.4000
TRIGGER:B:LOWERTHRESHOLD:D5	1.4000
TRIGGER:B:LOWERTHRESHOLD:D6	1.4000
TRIGGER:B:LOWERTHRESHOLD:D7	1.4000
TRIGGER:B:LOWERTHRESHOLD:D8	1.4000
TRIGGER:B:LOWERTHRESHOLD:D9	1.4000
TRIGGER:B:STATE	0
TRIGGER:B:TIME	8.0000E-9
TRIGGER:B:TYPE	EDGE
VERBOSE	1
VIDPIC:AUTOCONTRAST	1
VIDPIC:AUTOCONTRAST:UPDATERATE	20
ZOOM:MODE	0
ZOOM:ZOOM1:POSITION	50.0000
ZOOM:ZOOM1:SCALE	4.0000E-6
ZOOM:ZOOM1:STATE	0

Appendix D: Waveform Transfer (WFMOutpre and CURVe Query) Examples

Here are several examples of command sequences using the WFMOutpre? and CURVe? queries to transfer data from the instrument to a PC, with different data sources, including Analog, Digital waveforms. Each command sequence example is followed by an explanation of the returned WFMOutpre? results. The WFMOutpre? values act as settings that apply to the CURVe query data being transferred.

NOTE. *In order to guarantee that the waveform data returned from CURVE? queries of multiple waveforms are correlated to the same acquisition, you should use single sequence acquisition mode to acquire the waveform data from a single acquisition. Single sequence acquisition mode is enabled using SEQuence.*

Example 1: Analog Waveform (Channels 1–4)

Goal: Transfer 10,000 points of analog channel waveform data from the instrument to a PC.

Command	Comment
:DATa:SOUrce CH1	
:DATa:START 1	
:DATa:STOP 10000	
:WFMOutpre:ENCdg BINARY	
:WFMOutpre:BYT_Nr 1	
:HEADer 1	
:WFMOutpre?	Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations): :WFMOUTPRE:BYT_NR 1;BIT_NR 8;ENCDG BINARY;BN_FMT RI;BYT_OR MSB;WFID "Ch1, DC coupling, 100.0mV/div, 4.000us/div, 10000 points, Sample mode";NR_PT 10000;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 4.0000E-9;XZERO 139.9999E-12;PT_OFF 5000;YUNIT "V";YMULT 4.0000E-3;YOFF 0.0E+0;YZERO 0.0E+0
:CURVE?	Returns 10,000 data points: :CURVe #510000<10,000 binary bytes of waveform data.>

NOTE. *You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.*

WFMOutpre? Query results	Explanation
`BYT_NR 1`	This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOutpre:BYT_Nr command. Note that changing this value automatically changes the `BIT_NR` value accordingly. (This value can also be set using the command.)
`BIT_NR 8`	This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nr command. Note that changing this value automatically changes the `BYT_NR` value accordingly.
`ENCDG BINARY`	This value specifies the encoding of the waveform data. To change this value (the other possibility is ASCii), use the WFMOutpre:ENCdg command. (This value can also be set using the DATa:ENCdg command, which provides the ability to set the `WFMOutpre:ENCdg`, `WFMOutpre:BN_Fmt`, and `WFMOutpre:BYT_Or` values using a single command.)
`BN_FMT RI`	This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOutpre:BN_Fmt command. Note: this field is not applicable for ASCii encoding.
`BYT_OR MSB`	This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: this field is not applicable for ASCii encoding.
`WFID "Ch1, DC coupling, 100.0mV/div, 4.000us/div, 10000 points, Sample mode"`	This quoted string represents information about the source waveform that would be returned by a. `WFMOutpre:WFId?` query.
`NR_PT 10000`	This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOutpre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATa:STARt and DATa:STOP commands.
`PT_FMT Y`	This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOutpre:PT_Fmt? query.)
`PT_ORDER LINEAR`	This value is always `LINear`.
`XUNIT "s"`	This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is “s”, representing seconds. When using the math waveform as a source, the value can be “s” or “Hz”. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.)
`XINCR 4.0000E-9`	This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.)
`XZERO 139.9999E-12`	This value indicates the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger in the waveform record. The value is in units of WFMOutpre:XUNit. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.)
`PT_OFF 5000`	This value indicates the data point immediately following the trigger point relative to DATa:STARt in the waveform record. This is a query only. (If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)

WFMOutpre? Query results	Explanation
YUNIT "V"	This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, volts. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)
YMULT 4.0000E-3	This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)
YOFF 0.0E+0	This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFf? query.)
YZERO 0.0E+0	This value indicates the vertical offset of the source waveform in units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZEro? query.)

Example 2: Digital Waveform

Goal: Transfer 25 points of digital channel waveform data from the instrument to a PC.

Command	Comment
:DATa:SOUrce CH1_D5	
:DATa:START 1	
:DATa:STOP 25	
:WFMOutpre:ENCdg ASCii	
:WFMOutpre:BYT_Nr 1	
:HEADer 1	
:VERBose 1	
:WFMOutpre?	Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations): :WFMOUTPRE:BYT_NR 1;BIT_NR 8;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WFID "D5, unknown coupling, 100.0us/div, 10000 points, Digitalmode";NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 100.0000E-9;XZERO 155.000E-12;PT_OFF 5000;YUNIT "State";YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0
:CURVe?	Returns the following values. Each value represents a data point: :CURVe 0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0

NOTE. You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.

WFMOutpre? Query results	Explanation
`WFMOUTPRE:BYT_NR 1`	This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOutpre:BYT_Nr command. Note that changing this value automatically changes the `BIT_NR` value accordingly. (This value can also be set using the DATa:WIDth command.)
`BIT_NR 8`	This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nr command. Note that changing this value automatically changes the `BYT_NR` value accordingly.
`ENCDG ASCII`	This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOutpre:ENCdg command. (This value can also be set using the DATa:ENCdg command, which provides the ability to set the `WFMOutpre:ENCdg, WFMOutpre:BN_Fmt,`: and `WFMOutpre:BYT_Or` values using a single command.)
`BN_FMT RI`	This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOutpre:BN_Fmt command. Note: this field is not applicable for ASCii encoding.
`BYT_OR MSB`	This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: this field is not applicable for ASCii encoding.
`WFID "D5, unknown coupling, 100.0us/div, 10000 points, Digital mode"`	This quoted string represents information about the source waveform that would be returned by a. `WFMOutpre:WFId?` query. It cannot be changed.
`NR_PT 25`	This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOutpre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATa:STARt and DATa:STOP commands.
`PT_FMT Y`	This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only. (If you would like to determine only this value, use the WFMOutpre:PT_Fmt? query.)
`PT_ORDER LINEAR`	This value is always `LINear`.
`XUNIT "s"`	This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is “s”, representing seconds. When using the math waveform as a source, the value can be “s” or “Hz”. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.)
`XINCR 100.0000E-9`	This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.)
`XZERO 155.000E-12`	This value indicates the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger in the waveform record. The value is in units of WFMOutpre:XUNit. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.)
`PT_OFF 5000`	This value indicates the data point immediately following the trigger point relative to DATa:STARt in the waveform record.(If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)

WFMOutpre? Query results	Explanation
YUNIT "State"	This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)
YMULT 1.0000	This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)
YOFF 0.0E+0	This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFf? query.)
YZERO 0.0E+0	This value indicates the vertical offset of the source waveform in units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZEro? query.)

Example 3: Digital with 4 Bytes Per Point and Zoom Off

Goal: Transfer 25 points of Digital data from the instrument to a PC using 4 bytes per point and Zoom off.

Command	Comment
:DATa:SOUrce CH1_D1	
:ACQuire:MAGnivu 0	
:DATa:START 1	
:DATa:STOP 25	
:WFMOutpre:ENCdg ASCii	
:WFMOutpre:BYT_Nr 4	
:HEADer 1	
:VERBose 1	
:WFMOutpre?	Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations): :WFMOUTPRE:BYT_NR 4;BIT_NR 32;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode";NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 100.0000E-9;XZERO 155.000E-12;PT_OFF 5000;YUNIT "State";YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0
:CURVe?	Returns the following values. Each value represents a data point: :CURVe FB386,FB366,FB3E6,FB366,FB3E6,FB32E,FB3A2,FB32E,FB3AA,FB366, FB3EA,FB366,FB3E2,FB36,FB3E6,FB366,FB3E6,FB346,FB3C6,FB346,FB3C6, FB34E,FB3C2,FB34E,FB3CA

NOTE. *The returned hexadecimal data values are formatted without leading zeroes. For example, a 4-byte digital value of FB386 should be interpreted as 000FB386.*

NOTE. *You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.*

WFMOutpre? Query results	Explanation
WFMOUTPRE:BYT_NR 4	This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOutpre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATa:WIDth command.)
BIT_NR 32	This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly.
ENCDG ASCII	This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOutpre:ENCdg command. (This value can also be set using the DATa:ENCdg command, which provides the ability to set the WFMOutpre:ENCdg, WFMOutpre:BN_Fmt,: and WFMOutpre:BYT_Or values using a single command.)
BN_FMT RI	This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOutpre:BN_Fmt command. Note: this field is not applicable for ASCii encoding.
BYT_OR MSB	This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: this field is not applicable for ASCii encoding.
WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode"	This quoted string represents information about the source waveform that would be returned by a. WFMOutpre:WFId? query. It cannot be changed.
NR_PT 25	This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOutpre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATa:STARt and DATa:STOP commands.
PT_FMT Y	This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOutpre:PT_Fmt? query.)
PT_ORDER LINEAR	This value is always LINear.
XUNIT "s"	This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. For live channels, this value is “s”, representing seconds. When using the math waveform as a source, the value can be “s” or “Hz”. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.)

WFMOutpre? Query results	Explanation
XINCR 100.0000E-9	This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.)
XZERO 155.000E-12	This value indicates the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger in the waveform record. The value is in units of WFMOutpre:XUNit. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.)
PT_OFF 5000	This value indicates the data point immediately following the trigger point relative to DATa:STARt in the waveform record. (If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)
YUNIT "State"	This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)
YMULT 1.0000	This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)
YOFF 0.0E+0	This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFf? query.)
YZERO 0.0E+0	This value indicates the vertical offset of the source waveform in units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZEro? query.)

Example 4: Digital with 8 Bytes Per Point and Zoom Off

Goal: Transfer 25 points of Digital data from the instrument to a PC using 8 bytes per point and Zoom off.

Command	Comment
:DATa:SOUrce CH1_D1	
:ACQuire:MAGnivu 0	
:DATa:START 1	
:DATa:STOP 25	
:WFMOutpre:ENCdg ASCii	
:WFMOutpre:BYT_Nr 8	
:HEADer 1	
:VERBose 1	

Command	Comment
:WFMOutpre?	Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations): :WFMOUTPRE:BYT_NR 8;BIT_NR 64;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode";NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 100.0000E-9;XZERO 155.000E-12;PT_OFF 5000;YUNIT "State";YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0
:CURVe?	Returns the following values. Each value represents a data point: :CURVe 80000FB386,E0000FB386,80000FB3E6,80000FB3E6,80000FB3E6, C8000FB3A6,8C000FB3A6, 8C000FB3A6,84000FB3AE,CC000FB3A6, 8C000FB3E6,8C000FB3E6,84000FB3E6,80000FB3E6,80000FB3E6, 80000FB3E6,80000FB3E6,A0000FB3C6,80000FB3C6,80000FB3C6, 80000FB3C6,88000FB3C6,8C000FB3C6, 8C000FB3C6,84000FB3CE

NOTE. *The returned hexadecimal data values are formatted without leading zeroes.*

NOTE. *You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.*

WFMOutpre? Query results	Explanation
WFMOUTPRE:BYT_NR 8	This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOutpre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATa:WIDth command.)
BIT_NR 64	This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nr command. Note that changing this value automatically changes the BYT_NR value accordingly.
ENCDG ASCII	This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOutpre:ENCdg command. (This value can also be set using the DATa:ENCdg command, which provides the ability to set the WFMOutpre:ENCdg, WFMOutpre:BN_Fmt,: and WFMOutpre:BYT_Or values using a single command.)
BN_FMT RI	This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOutpre:BN_Fmt command. Note: this field is not applicable for ASCii encoding.
BYT_OR MSB	This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: this field is not applicable for ASCii encoding.

WFMOutpre? Query results	Explanation
WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode"	This quoted string represents information about the source waveform that would be returned by a. WFMOutpre:WFId? query. It cannot be changed.
NR_PT 25	This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOutpre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATa:STARt and DATa:STOP commands.
PT_FMT Y	This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOutpre:PT_Fmt? query.)
PT_ORDER LINEAR	This value is always LINear.
XUNIT "s"	This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is “s”, representing seconds. When using the math waveform as a source, the value can be “s” or “Hz”. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.)
XINCR 100.0000E-9	This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.)
XZERO 155.000E-12	This value indicates the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger in the waveform record. The value is in units of WFMOutpre:XUNit. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.)
PT_OFF 5000	This value indicates the data point immediately following the trigger point relative to DATa:STARt in the waveform record. (If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)
YUNIT "State"	This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)
YMULT 1.0000	This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)
YOFF 0.0E+0	This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFf? query.)
YZERO 0.0E+0	This value indicates the vertical offset of the source waveform in units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZEro? query.)

Example 5: Digital with 4 Bytes Per Point and Zoom On

Goal: Transfer 25 points of Digital data from the instrument to a PC using 4 bytes per point and Zoom on.

Command	Comment
:DATa:SOUrce CH1_D1	
:ACQuire:MAGnivu 1	
:DATa:START 1	
:DATa:STOP 25	
:WFMOutpre:ENCdg ASCii	
:WFMOutpre:BYT_Nr 4	
:HEADer 1	
:VERBose 1	
:WFMOutpre?	Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations): :WFMOUTPRE:BYT_NR 4;BIT_NR 32;ENCDG ASCII;BN_FMT RI;BYT_OR MSB;WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode";NR_PT 25;PT_FMT Y;PT_ORDER LINEAR;XUNIT "s";XINCR 1.2121E-9;XZERO 155.000E-12;PT_OFF 5000;YUNIT "State";YMULT 1.0000;YOFF 0.0E+0;YZERO 0.0E+0
:CURVe?	Returns the following values. Each value represents a data point: :CURVe FB6E6,FB666,FB6E6,FB666,FB666,FB6E6,FB666,FB6E6,FB666, FB666, FB6E6,FB666,FB6E6,FB6E6,FB666,FB6E6,FB666,FB666, FB6E6,FB666,FB6E6,FB6E6,FB666,FB6E6,FB666

NOTE. *The returned hexadecimal data values are formatted without leading zeroes.*

NOTE. *You can also use the WAVFrm? query, which concatenates the WFMOutpre? and CURVe? queries.*

WFMOutpre? Query results	Explanation
WFMOUTPRE:BYT_NR 4	This value specifies the number of bytes per data point in the waveform data. To change this value automatically, use the WFMOutpre:BYT_Nr command. Note that changing this value automatically changes the BIT_NR value accordingly. (This value can also be set using the DATa:WIDth command.)
BIT_NR 32	This value specifies the number of bits per data point in the waveform data. To change this value, use the WFMOutpre:BIT_Nrcommand. Note that changing this value automatically changes the BYT_NR value accordingly.

WFMOutpre? Query results	Explanation
ENCDG ASCII	This value specifies the encoding of the waveform data. To change this value (the other possibility is BINARY), use the WFMOutpre:ENCdg command. (This value can also be set using the DATa:ENCdg command, which provides the ability to set the WFMOutpre:ENCdg, WFMOutpre:BN_Fmt,: and WFMOutpre:BYT_Or values using a single command.)
BN_FMT RI	This value specifies the binary format, which in this case is RI (signed integer). To change this value (the other possibility is RP or positive integer), use the WFMOutpre:BN_Fmt command. Note: this field is not applicable for ASCii encoding.
BYT_OR MSB	This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: this field is not applicable for ASCii encoding.
WFID "Digital, unknown coupling, 100.0us/div, 10000 points, Digital mode"	This quoted string represents information about the source waveform that would be returned by a. WFMOutpre:WFId? query. It cannot be changed.
NR_PT 25	This value indicates the number of data points in the waveform record to be transferred using the CURVE? query. (If you would like to determine only this value, use the WFMOutpre:NR_Pt? query.) Note: this value is typically equal to the full record length, but you also have the option to transfer only a portion of the record length by using the DATa:STARt and DATa:STOP commands.
PT_FMT Y	This value indicates the format of the data points in the waveform record. In this case, the value represents YT format. This is query only – the returned values can be Y for YT format or ENV for envelope format (min/max pairs). (If you would like to determine only this value, use the WFMOutpre:PT_Fmt? query.)
PT_ORDER LINEAR	This value is always LINear.
XUNIT "s"	This value indicates the units of the x-axis of the waveform record. This is determined by the horizontal settings for the waveform source. Typically, this value is “s”, representing seconds. When using the math waveform as a source, the value can be “s” or “Hz”. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.)
XINCR 1.2121E-9	This value indicates the time, in seconds, or frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.)
XZERO 155.000E-12	This value indicates the sub-sample time between the trigger sample (designated by PT_OFF) and the occurrence of the actual trigger in the waveform record. The value is in units of WFMOutpre:XUNit. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.)
PT_OFF 5000	This value indicates the data point immediately following the trigger point relative to DATa:STARt in the waveform record. (If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)
YUNIT "State"	This value indicates the vertical units of data points in the waveform record. This can be any of several string values, depending upon the vertical units of the source waveform – in this case, State. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)
YMULT 1.0000	This value indicates the multiplying factor to convert the data point values from digitizing levels to the units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)

WFMOutpre? Query results	Explanation
YOFF 0.0E+0	This value indicates the vertical position of the source waveform in digitizing levels. There are 25 digitizing levels per vertical division. This is query only. (If you would like to determine only this value, use the WFMOutpre:YOFf? query.)
YZERO 0.0E+0	This value indicates the vertical offset of the source waveform in units specified by the WFMOUTpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YZEro? query.)

Example 6: RF Frequency Domain Waveform

Goal: Transfer 5 points of RF frequency domain data from the instrument to a PC using 8 bytes per point.

NOTE. *A frequency domain waveform used in data transfer from the instrument to a PC or other device may be one of the four RF frequency domain traces or the Spectrum Math waveform.*

Command	Comment
:DATa:SOUrce CH1_SV_NORMal	
:DATa:STARt 495	
:DATa:STOP 505	
:WFMOutpre:ENCdg ASCii	
:WFMOutpre:BYT_Nr 8	
:HEADer 1	
:VERBose 1	
:WFMOutpre?	Returns the following values. Each value represents the current settings that a CURVe? query will use to format the data that will be transferred from the instrument to a PC or other device (see next table for explanations): :WFMOUTPRE:BYT_NR 8;BIT_NR 64;ENCDG ASCII;BN_FMT FP;BYT_OR MSB;WFID "RF_NORMAL, unknown coupling, 10.00W/div, 200.0MHz/div, 1001 points, Sample mode, Center Freq: 2.400GHz, Span: 2.000GHz, Reference Level: 1.000mW";NR_PT 21;PT_FMT Y;PT_ORDER LINEAR;XUNIT "Hz";XINCR 2.0000E+6;XZERO 1.4000E+9;PT_OFF 0;YUNIT "W";YMULT 1.5625E-3;YOFF 0.0E+0;YZERO 0.0E+0;DOMAIN FREQUENCY;WFMTYPE RF_FD;CENTERFREQUENCY 2.4000E+9;SPAN 2.0000E+9;REFLEVEL 1.0000E-3
:CURVe?	Returns the following values. Each value represents a data point: :CURVE 2.43108E-10,1.62648E-10,2.78478E-10,4.15163E-10,7.66223E-10, 7.63905E-10,3.68375E-10,3.42668E-10,3.06422E-10,1.81561E-10, 2.0223E-10,4.24327E-10,1.95298E-10,2.1304E-10,9.41791E-11, 4.56513E-10,4.33067E-10,6.57215E-11,1.65388E-10,9.09116E-10, 5.39507E-10

WFMOutpre? Query results	Explanation
BYT_NR 8	This value specifies the number of bytes per data point in the waveform data. This value is fixed for RF frequency domain traces.
BIT_NR 64	This value specifies the number of bits per data point in the waveform data. This value is fixed for RF frequency domain traces.
ENCDG ASCII	This value specifies the encoding of the waveform data. To change this value (the other possibility for RF frequency domain traces is FPBINARY), use the WFMOutpre:ENCdg command. This value can also be set using the DATa:ENCdg command.
BN_FMT FP	This value specifies the binary format, which in this case is FP (floating point). This value is fixed for RF frequency domain traces. Note: This field is not applicable for ASCii encoding.
BYT_OR MSB	This value specifies the byte order for the BINARY encoding, which in this case is MSB (most significant byte first, also known as IBM format). To change this value to LSB, use the WFMOutpre:BYT_Or command. Note: This field is not applicable for ASCii encoding.
WFID "CH1_SV_NORMal, unknown coupling, 10.00W/div, 200.0MHz/div, 1001 points, Sample mode, Center Freq: 2.400GHz, Span: 2.000GHz, Reference Level: 1.000mW"	This quoted string represents information about the source waveform that would be returned by a WFMOutpre:WFId? Query. It cannot be changed.
NR_PT 21	This value indicates the number of data points in the waveform record. (If you would like to determine only this value, use the WFMOutpre:NR_Pt? query.) Note: This value is typically equal to the full record length of the RF frequency domain trace, but you also have the option to transfer only a portion of the record using the DATs:STARt and DATa:STOP commands.
PT_FMT Y	This value indicates the format of the data points in the waveform record. In this case, the value represents YF (Amplitude vs. Frequency) format. This is query only. (If you would like to determine only this value, use the WFMOutpre:PT_Fmt? query.
PT_ORDER LINEAR .	This value is always LINear
XUNIT "Hz"	This value indicates the units of the x-axis of the waveform record. This value is always “Hz” for RF frequency domain traces. This is query only. (If you would like to determine only this value, use the WFMOutpre:XUNit? query.)
XINCR 2.0000E+6	This value indicates the frequency, in hertz, between data points in the waveform record. This is query only. (If you would like to determine only this value, use the WFMOutpre:XINcr? query.)
XZERO 1.4000E+9	This value indicates the frequency, in hertz, of the first data point in the waveform record. This frequency is relative to the time of the trigger, which is always 0. This XZEro frequency can be negative. This is query only. (If you would like to determine only this value, use the WFMOutpre:XZEro? query.)
PT_OFF 0	This is a query provided only for compatibility with other Tektronix instruments. The returned value is always 0. (If you would like to determine only this value, use the WFMOutpre:PT_Off? query.)
YUNIT "W"	This value indicates the units of data points in the waveform record. This value depends on the vertical units of the source waveform – in this case, watts. This is query only. (If you would like to determine only this value, use the WFMOutpre:YUNit? query.)
YMULT 1.5625E-3	This value indicates the multiplying factor to convert the data point values to the units specified by the:WFMOutpre:YUNit command. This is query only. (If you would like to determine only this value, use the WFMOutpre:YMUlt? query.)

YOFF 0.0E+0	This value indicates the vertical position of the source waveform. This value is unused for RF frequency domain traces and is always 0.
YZERO 0.0E+0	This value indicates the vertical offset of the source waveform. This value is unused for RF frequency domain traces and is always 0.
DOMAIN FREQUENCY	This value indicates the domain in which the source waveform is displayed and stored. For RF time domain traces, the domain is Time and waveform transfer information is treated as integer information. For RF frequency domain traces, the domain is Frequency and waveform transfer information is treated as floating point information. This is query only. (If you would like to determine only this value, use the WFMOutpre:DOMain? query.)
WFMTYPE RF_FD	This value indicates the type of the source waveform. RF_FD indicates an RF frequency domain trace (frequency domain waveform). This is query only. (If you would like to determine only this value, use the WFMOutpre:WFMTYPe? query.)
CENTERFREQUENCY 2.4000E+9	This value indicates the center frequency, in hertz, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOutpre:CENTERFREQuency? query.)
SPAN 2.0000E+9	This value indicates the frequency span, in hertz, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOutpre:SPAN? query.)
REFLEVEL 1.0000E-3	This value indicates the reference level, in watts, of the source waveform. This is query only. (If you would like to determine only this value, use the WFMOutpre:REFLEvel? query.)

Example 7: Baseband IQ Waveform

Goal: Transfer Baseband I and Q data from the instrument to a PC.

NOTE. *Requires SV-RFVT option.*

Command	Comment
`:DATa:SOUrce CH1_BASEBAND_IQ`	
`:DATa:WIDth 2`	Set data width for every I,Q pair to 2.
`:WFMOutpre:ENCdg SRIbinary`	Request data LSB first.
`:WFMOutpre:YMUlt?`	Apply 2*YMUlt scaling to I and Q samples returned by CURVe?

Command	Comment
:WFMOutpre?	Returns the following values. Each value represents the current settings that aCURVe? query will use to format the data that will be transferred from the instrument to a PC or other device: :WFMOUTPRE:BYT_NR 1;BIT_NR 8;:WFMOUTPRE:ENCDG BINARY;BN_FMT RI;:WFMOUTPRE:ASC_FMT INTEGER;BYT_OR MSB;:WFMOUTPRE:WFID "CH1_SV_BASEBAND_IQ, 5062 points, Center Freq: 1GHz, Span: 312.5MHz";NR_PT 9;:WFMOUTPRE:PT_FMT ENV;PT_ORDER LINEAR;:WFMOUTPRE:XUNIT "s";XINCR 1.280E-9;:WFMOUTPRE:XZERO 815.000E-12;PT_OFF 0;:WFMOUTPRE:YUNIT "V";YMULT 8.0000E-3;:WFMOUTPRE:YOFF 0.0E+0;YZERO0.0E+0;:WFMOUTPRE:DOMAIN TIME;WFMTYPE F_TD;:WFMOUTPRE:CENTERFREQUENCY 1.0000E+9;SPAN 312.5000E+6
:CURVe?	CURVe? will transmit NR_Pt points which are interleaved I and Q pairs. The result can be read from the binary buffer as uint16 to get I and Q pair values in “digitizing levels”. Multiply the digitizing level value by 2*YMUlt to get absolute I and Q point values.

Appendix E: Search and Trigger Command Sequence Examples

The following are some example command sequences that show a variety of different searches and triggers. The commands in these sequences are not order-dependent.

To use these examples, connect channel 1, channel 2, channel 3 and channel 4 to the probe compensation signal located on the right hand side of the front panel.

The search and trigger command group sections contain more information on general search and trigger concepts. ,

Example 1: Single Threshold Edge Search

Goal: Search the channel 2 waveform and place a mark at each instance where it crosses below a threshold of 1.4 volts.

Command	Comment
*RST;:*OPC?	Resets the instrument and waits for that operation to complete (approximately 1 to 5 seconds depending on the complexity of the previous setup).
display:waveview1:ch2:state 1	Turns the CH2 waveform on.
:AUTOset EXECute>;*OPC?	Autosets the displayed waveform CH2 and waits for the auto setup to complete.
SEARCH:SEARCH1:TRIGger:A:TYPe EDGE	Specifies that this will be an edge search (a mark will be placed when the source waveform passes through a specified threshold level in the specified direction).
SEARCH:SEARCH1:TRIGger:A:EDGE:SOUrce CH2	Specifies the CH2 waveform as the source waveform.
SEARCH:SEARCH1:TRIGger:A:LOWerthreshold:CH2 1.4	Specifies 1.4 volts as the threshold level.
SEARCH:SEARCH1:TRIGger:A:EDGE:SLOpe FALL	Specifies the falling edge as the direction.
SEARCH:SEARCH1:STATE 1	Turns the search on.
SEARCH:SEARCH1:TOTal?	Returns 4, indicating that CH2 fell below the 1.4 volt threshold 4 times.

Example 2: Single Threshold Edge Trigger

Goal: Trigger on the channel 2 waveform when the waveform crosses below a threshold of 1.4 volts.

Command	Comment
*RST	Resets the instrument. Wait for the reset to complete (approximately 3 seconds).
:DISPLAY:WAVEVIEW1:CH2:STATE 1	Turns the CH2 waveform on.

Command	Comment
:TRIGger:A:TYPe EDGE	Specifies that this will be an edge trigger (trigger will occur when the source waveform passes through a specified threshold level in the specified direction).
:TRIGger:A:EDGE:SOUrce CH2	Specifies the CH2 waveform as the source waveform.
:TRIGger:A:LOWerthreshold:CH2 1.4	Specifies 1.4 volts as the threshold level.
:TRIGger:A:EDGE:SLOpe FALL	Specifies as falling edge as the direction.
:TRIGger:STATE?	Should return TRIGGER, not AUTO.

Example 3: Dual Threshold Runt Search

Goal: Search the channel 3 waveform for negative runt pulses and place a mark at each instance when the waveform drops below an upper threshold of 1.4 volts, but does not cross a lower threshold of -2 volts before re-crossing the upper threshold. The pulse width must be less than 600 microseconds.

Command	Comment
*RST	Resets the instrument. Wait for the reset to complete (approximately 3 seconds).
:DISPLAY:WAVEVIEW1:CH3:STATE 1	Turns the CH3 waveform on.
:AUTOset EXECute	Autosets the displayed waveform CH3. Wait for the autoset to complete (approximately 3 seconds).
SEARCH:SEARCH1:TRIGger:A:TYPe RUNT	Specifies that this will be a runt search (a mark will be placed on a pulse amplitude that crosses one threshold but fails to cross a second threshold before re-crossing the first).
SEARCH:SEARCH1:TRIGger:A:EDGE:SOUrce CH3	Specifies to use channel 3 as the source waveform.
SEARCH:SEARCH1:TRIGger:A:LOWerthreshold:CH3 -2	Specifies to use -2 volts as the lower threshold.
SEARCH:SEARCH1:TRIGger:A:UPPerthreshold: CH3 1.4	Specifies to use 1.4 volts as the upper threshold.
SEARCH:SEARCH1:TRIGger:A:RUNT:POLarity NEGative	Specifies to search for when the runt polarity is negative.
SEARCH:SEARCH1:TRIGger:A:RUNT:WIDth 600E-6	Specifies a pulse width of 600E-6 seconds.
SEARCH:SEARCH1:TRIGger:A:RUNT:WHEn LESSthan	Specifies to search for when the pulse width is less than 600E-6 seconds.
SEARCH:SEARCH1:STATE 1	Turns the search on.
SEARCH:SEARCH1:TOTal?	Returns 3, indicating that 3 total negative runt pulses were less than 600E-6 seconds wide

NOTE. *You could use a similar command sequence with a transition type search.*

Example 4: Single Threshold Logic Search on Three Waveforms

Goal: Search the channel 1, 2 and 3 waveforms and place a mark at each instance when either channel 1 is above 1.4 volts, channel 2 is above 1.5 volts, or channel 3 is above 1.3 volts.

Command	Comment
*RST	Resets the instrument. Wait for the reset to complete (approximately 3 seconds).
:DISPLAY:WAVEVIEW1:CH1:STATE 1	Turns the CH1 waveform on.
:DISPLAY:WAVEVIEW1:CH2:STATE 1	Turns the CH2 waveform on.
:DISPLAY:WAVEVIEW1:CH3:STATE 1	Turns the CH3 waveform on.
:AUTOset EXECute	Autosets the displayed waveforms CH1, CH2 and CH3. Wait for the autoset to complete (approximately 3 seconds).
SEARCH:SEARCH1:TRIGger:A:TYPe LOGIc	Specifies that this will be a logic search (a mark will be placed when all channels transition to the specified state).
SEARCH:SEARCH1:TRIGger:A:LOGIc:INPut:CH1 HIGH	Specifies the Boolean logic criteria for channel 1; in this case, high.
SEARCH:SEARCH1:TRIGger:A:LOGIc:INPut:CH2 HIGH	Specifies the Boolean logic criteria for channel 2; in this case, high.
SEARCH:SEARCH1:TRIGger:A:LOGIc:INPut:CH3 HIGH	Specifies the Boolean logic criteria for channel 3; in this case, high.
SEARCH:SEARCH1:TRIGger:A:LOGIc:THReshold:CH1 1.4	Specifies to use 1.4 volts as the threshold for CH1.
SEARCH:SEARCH1:TRIGger:A:LOGIc:THReshold:CH2 1.5	Specifies to use 1.5 volts as the threshold for CH2.
SEARCH:SEARCH1:TRIGger:A:LOGIc:THReshold:CH3 1.3	Specifies to use 1.3 volts as the threshold for CH3.
SEARCH:SEARCH1:TRIGger:A:LOGIc:PATtern:WHEn TRUE	Specifies the condition for generating a logic pattern search; in this case, true.
SEARCH:SEARCH1:TRIGger:A:LOGIc:FUNCtion OR	Specifies the logic operator for the logic search; in this case, OR.
SEARCH:SEARCH1:STATE 1	Turns the search on.
SEARCH:SEARCH1:TOTal?	Returns 3 or 4, indicating 3 or 4 times when one of channels 1, 2 or 3 became high. Note: Depending upon the number of transitions displayed, you may get 3 or 4 search marks for this example. If you get 3 search marks, try adjusting the horizontal position until you see 4 search marks.

Glossary

ASCII
Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the instrument using ASCII character encoding.

Address
A 7-bit code that identifies an instrument on the communication bus. The digitizing instrument must have a unique address for the controller to recognize and transmit commands to it.

Backus-Naur Form (BNF)
A standard notation system for command syntax diagrams. The syntax diagrams in this manual use BNF notation.

Controller
A computer or other device that sends commands to and accepts responses from the digitizing instrument.

EOI
A mnemonic referring to the control line End or Identify. One of the two possible end-of-message terminators.

EOM
A generic acronym referring to the end-of-message terminator. The end-of-message terminator can be either an EOI or the ASCII code for line feed (LF).

Equivalent-Time sampling (ET)
A sampling mode in which the instrument acquires signals over many repetitions of the event. This instrument uses a type of equivalent time sampling called random equivalent time sampling. It utilizes an internal clock that runs asynchronously with respect to the input signal and the signal trigger. The instrument takes samples continuously, independent of the trigger position, and displays them based on the time difference between the sample and the trigger. Although the samples are taken sequentially in time, they are random with respect to the trigger.

Real-Time sampling
A sampling mode where the instrument samples fast enough to completely fill a waveform record from a single trigger event. Use real-time sampling to capture single-shot or transient events.

IEEE
An acronym for the Institute for Electrical and Electronic Engineers.

Serial poll
A device on the bus can request service from the Controller by asserting the SRQ line. When a controller acknowledges the SRQ, it serial polls each device on the bus to determine which device on the bus requested service. Any device requesting service returns a status byte indicating it needs to be serviced and then

unasserts the SRQ line. Devices not requiring service return a status byte that indicates they do not need servicing.

TEKSecure
A Tektronix custom command that initializes both waveform and setup memories. This overwrites any previously stored data.

Index

A

ACQuire?, 2-135
ACQuire:FASTAcq:PALEtte, 2-135
ACQuire:FASTAcq:STATE, 2-136
ACQuire:MAXSamplerate?, 2-136
ACQuire:MODe, 2-136
ACQuire:NUMACq?, 2-138
ACQuire:NUMAVg, 2-138
ACQuire:NUMFRAMESACQuired?, 2-139
ACQuire:SEQuence:CURrent?, 2-139
ACQuire:SEQuence:MODe, 2-139
ACQuire:SEQuence:NUMSEQuence, 2-140
ACQuire:STATE, 2-140
ACQuire:STOPAfter, 2-141
ACTONEVent:ENable, 2-142
ACTONEVent:LIMit, 2-142
ACTONEVent:LIMITCount, 2-143
ACTONEVent:MASKFail:ACTION:SAVEIMAGe:STATE, 2-143
ACTONEVent:MASKFail:ACTION:SAVEWAVEform:STATE, 2-144
ACTONEVent:MASKFail:ACTION:SRQ:STATE, 2-144
ACTONEVent:MASKFail:ACTION:STOPACQ:STATE, 2-145
ACTONEVent:MASKHit:ACTION:SAVEIMAGe:STATE, 2-146
ACTONEVent:MASKHit:ACTION:SAVEWAVEform:STATE, 2-146
ACTONEVent:MASKHit:ACTION:SRQ:STATE, 2-147
ACTONEVent:MASKHit:ACTION:STOPACQ:STATE, 2-148
ACTONEVent:MASKPass:ACTION:SAVEIMAGe:STATE, 2-148
ACTONEVent:MASKPass:ACTION:SAVEWAVEform:STATE, 2-149
ACTONEVent:MASKPass:ACTION:SRQ:STATE, 2-150
ACTONEVent:MASKPass:ACTION:STOPACQ:STATE, 2-150
ACTONEVent:MEASUrement:ACTION:SAVEIMAGe:STATE, 2-151
ACTONEVent:MEASUrement:ACTION:SAVEWAVEform:STATE, 2-152
ACTONEVent:MEASUrement:ACTION:SRQ:STATE, 2-153
ACTONEVent:MEASUrement:ACTION:STOPACQ:STATE, 2-153
ACTONEVent:SEARCH:ACTION:SAVEIMAGe:STATE, 2-154
ACTONEVent:SEARCH:ACTION:SAVEWAVEform:STATE, 2-155
ACTONEVent:SEARCH:ACTION:SRQ:STATE, 2-155
ACTONEVent:SEARCH:ACTION:STOPACQ:STATE, 2-156
ACTONEVent:TRIGger:ACTION:SAVEIMAGe:STATE, 2-157
ACTONEVent:TRIGger:ACTION:SAVEWAVEform:STATE, 2-157
ACTONEVent:TRIGger:ACTION:SRQ:STATE, 2-158
ACTONEVent:TRIGger:ACTION:STOPACQ:STATE, 2-158
AFG:AMPLitude, 2-159
AFG:ARBitrary:SOUrce, 2-160
AFG:FREQuency, 2-160
AFG:FUNCtion, 2-161
AFG:HIGHLevel, 2-161
AFG:LOWLevel, 2-162
AFG:NOISEAdd:PERCent, 2-162
AFG:NOISEAdd:STATE, 2-163
AFG:OFFSet, 2-163
AFG:OUTPut:LOAd:IMPEDance, 2-164
AFG:OUTPut:STATE, 2-164
AFG:PERIod, 2-165
AFG:PULse:WIDth, 2-165
AFG:RAMP:SYMmetry, 2-166
AFG:SQUare:DUty, 2-166
ALIas, 2-167
ALIas:CATalog?, 2-167
ALIas:DEFine, 2-168
ALIas:DELEte, 2-169
ALIas:DELEte:ALL, 2-169
ALIas:DELEte:NAMe, 2-169
ALIas:STATE, 2-170

ALLEv?, 2-170
AUTOSAVEPITIMEOUT, 2-171
AUTOSAVEUITIMEOUT, 2-171
AUTOSet, 2-172
AUTOSet:ACQuisition:ENAble, 2-172
AUTOSet:ENAble, 2-173
AUTOSet:HORizontal:ENAble, 2-173
AUTOSet:TRIGger:ENAble, 2-174
AUTOSet:VERTical:ENAble, 2-174
AUTOSet:VERTical:OPTIMize, 2-175
AUXout:EDGE, 2-176
AUXout:SOUrce, 2-176

B

BUS:ADDNew, 2-177
BUS:B<x>:ARINC429A:BITRate, 2-177
BUS:B<x>:ARINC429A:BITRate:CUSTom, 2-178
BUS:B<x>:ARINC429A:DATAFORmat, 2-178
BUS:B<x>:ARINC429A:POLARITY, 2-179
BUS:B<x>:ARINC429A:SOUrce, 2-180
BUS:B<x>:ARINC429A:THRESHold, 2-180
BUS:B<x>:AUDio:BITDelay, 2-181
BUS:B<x>:AUDio:BITOrder, 2-182
BUS:B<x>:AUDio:CLOCk:POLarity, 2-182
BUS:B<x>:AUDio:CLOCk:SOUrce, 2-183
BUS:B<x>:AUDio:CLOCk:THReshold, 2-184
BUS:B<x>:AUDio:DATa:POLarity, 2-184
BUS:B<x>:AUDio:DATa:SIZe, 2-185
BUS:B<x>:AUDio:DATa:SOUrce, 2-185
BUS:B<x>:AUDio:DATa:THReshold, 2-186
BUS:B<x>:AUDio:DATa:WORDSize, 2-187
BUS:B<x>:AUDio:FRAME:
CLOCKBITSPERCHANNEL, 2-187
BUS:B<x>:AUDio:FRAME:SIZe, 2-188
BUS:B<x>:AUDio:TYPe, 2-188
BUS:B<x>:AUDio:WORDSel:POLarity, 2-189
BUS:B<x>:AUDio:WORDSel:SOUrce, 2-190
BUS:B<x>:AUDio:WORDSel:THReshold, 2-190
BUS:B<x>:AUTOETHERnet:
DATAMINUSTHRESHOLD, 2-191
BUS:B<x>:AUTOETHERnet:
DATAPLUSTHRESHold, 2-192
BUS:B<x>:AUTOETHERnet:
LOWDATAMINus, 2-192
BUS:B<x>:AUTOETHERnet:
LOWDATAPLUS, 2-193
BUS:B<x>:AUTOETHERnet:
LOWTHRESHold, 2-194
BUS:B<x>:AUTOETHERnet:SIGNALTYpe, 2-194
BUS:B<x>:AUTOETHERnet:SOUrce, 2-195
BUS:B<x>:AUTOETHERnet:SOUrce:
DMINus, 2-196
BUS:B<x>:AUTOETHERnet:SOUrce:DPLUs, 2-196
BUS:B<x>:AUTOETHERnet:THRESHold, 2-197
BUS:B<x>:AUTOETHERnet:TYPe, 2-198
BUS:B<x>:CAN:BITRate, 2-198
BUS:B<x>:CAN:BITRate:VALue, 2-199
BUS:B<x>:CAN:FD:BITRate, 2-200
BUS:B<x>:CAN:FD:BITRate:CUSTom, 2-200
BUS:B<x>:CAN:SAMPLEpoint, 2-201
BUS:B<x>:CAN:SIGNal, 2-201
BUS:B<x>:CAN:SOUrce, 2-202
BUS:B<x>:CAN:STANDard, 2-202
BUS:B<x>:CAN:THReshold, 2-203
BUS:B<x>:DISplay:FORMat, 2-204
BUS:B<x>:DISplay:LAYout, 2-204
BUS:B<x>:DPHY:CLOCk:SOUrce, 2-205
BUS:B<x>:DPHY:CLOCk:THRESHold, 2-206
BUS:B<x>:DPHY:DMINus:
DATATHRESHold, 2-206
BUS:B<x>:DPHY:DMINus:LPTHRESHold, 2-207
BUS:B<x>:DPHY:DMINus:SOUrce, 2-207
BUS:B<x>:DPHY:DPlus:DATATHRESHold, 2-208
BUS:B<x>:DPHY:DPlus:LPTHRESHold, 2-209
BUS:B<x>:DPHY:DPlus:SOUrce, 2-209
BUS:B<x>:DPHY:LP:DIRection, 2-210
BUS:B<x>:DPHY:PROTocol:TYPe, 2-210
BUS:B<x>:DPHY:SIGNal:ENCoding, 2-211
BUS:B<x>:ETHERnet:
DATAMINUSTHRESHold, 2-212
BUS:B<x>:ETHERnet:
DATAPLUSTHRESHold, 2-212
BUS:B<x>:ETHERnet:IPVFOUR, 2-213
BUS:B<x>:ETHERnet:LOWTHRESHold, 2-214
BUS:B<x>:ETHERnet:QTAGGING, 2-214
BUS:B<x>:ETHERnet:SIGNALTYpe, 2-215
BUS:B<x>:ETHERnet:SOUrce, 2-215
BUS:B<x>:ETHERNET:SOUrce:DMINus, 2-216
BUS:B<x>:ETHERNET:SOUrce:DPLUs, 2-217
BUS:B<x>:ETHERnet:THRESHold, 2-217
BUS:B<x>:ETHERnet:TYPe, 2-218
BUS:B<x>:EUSB:BITRate, 2-218

BUS:B<x>:EUSB:DATAMINUS:DATA:
THRESHold, 2-219
BUS:B<x>:EUSB:DATAMINUSTHRESHold, 2-219
BUS:B<x>:EUSB:DATAPLUS:DATA:
THRESHold, 2-220
BUS:B<x>:EUSB:DATAPLUSTHRESHold, 2-221
BUS:B<x>:EUSB:LOWTHRESHold, 2-221
BUS:B<x>:EUSB:OPERating:MODe, 2-222
BUS:B<x>:EUSB:SIGNALTYpe, 2-222
BUS:B<x>:EUSB:SOUrce:DIFF, 2-223
BUS:B<x>:EUSB:SOUrce:DMINus, 2-223
BUS:B<x>:EUSB:SOUrce:DPLUs, 2-224
BUS:B<x>:EUSB:THRESHold , 2-224
BUS:B<x>:FLEXray:BITRate, 2-225
BUS:B<x>:FLEXray:BITRate:CUSTom, 2-225
BUS:B<x>:FLEXray:CHannel, 2-226
BUS:B<x>:FLEXray:LOWTHRESHold, 2-227
BUS:B<x>:FLEXray:SIGnal, 2-227
BUS:B<x>:FLEXray:SOUrce, 2-228
BUS:B<x>:FLEXray:SOUrce:TXRX, 2-228
BUS:B<x>:FLEXray:THRESHold, 2-229
BUS:B<x>:FLEXray:TXRXTHRESHold, 2-229
BUS:B<x>:I2C:CLOCk:SOUrce, 2-230
BUS:B<x>:I2C:CLOCk:THReshold, 2-231
BUS:B<x>:I2C:DATa:SOUrce, 2-231
BUS:B<x>:I2C:DATa:THReshold, 2-232
BUS:B<x>:I2C:RWINADDR, 2-232
BUS:B<x>:I3C:CLOCk:SOUrce, 2-233
BUS:B<x>:I3C:CLOCk:THReshold, 2-234
BUS:B<x>:I3C:DATa:SOUrce, 2-234
BUS:B<x>:I3C:DATa:THReshold, 2-235
BUS:B<x>:LABel:COLor, 2-236
BUS:B<x>:LABel:FONT:BOLD, 2-236
BUS:B<x>:LABel:FONT:ITALic, 2-237
BUS:B<x>:LABel:FONT:SIZE, 2-237
BUS:B<x>:LABel:FONT:TYPE, 2-238
BUS:B<x>:LABel:FONT:UNDERline, 2-238
BUS:B<x>:LABel:name, 2-239
BUS:B<x>:LABel:XPOS, 2-240
BUS:B<x>:LABel:YPOS, 2-240
BUS:B<x>:LIN:BITRate, 2-241
BUS:B<x>:LIN:BITRate:CUSTom, 2-241
BUS:B<x>:LIN:IDFORmat, 2-242
BUS:B<x>:LIN:POLarity, 2-242
BUS:B<x>:LIN:SAMPLEpoint, 2-243
BUS:B<x>:LIN:SOUrce, 2-243
BUS:B<x>:LIN:SOUrce:THReshold, 2-244
BUS:B<x>:LIN:STANDard, 2-244
BUS:B<x>:MANChester:BITORDer, 2-245
BUS:B<x>:MANChester:BITRate, 2-246
BUS:B<x>:MANChester:DISplaymode, 2-246
BUS:B<x>:MANChester:HEADer:LENGth, 2-247
BUS:B<x>:MANChester:IDLE:BITS, 2-247
BUS:B<x>:MANChester:parity, 2-248
BUS:B<x>:MANChester:SOUrce, 2-249
BUS:B<x>:MANChester:START:INDex, 2-249
BUS:B<x>:MANChester:SYNC:SIZe, 2-250
BUS:B<x>:MANChester:THReshold, 2-250
BUS:B<x>:MANChester:TOLerance, 2-251
BUS:B<x>:MANChester:TRAiler:LENGth, 2-252
BUS:B<x>:MANChester:TRANstion:ZERo, 2-252
BUS:B<x>:MANChester:WORD:COUNt, 2-253
BUS:B<x>:MANChester:WORDSIZe, 2-253
BUS:B<x>:MDIO:CLOCk:SOUrce, 2-254
BUS:B<x>:MDIO:CLOCk:THReshold, 2-255
BUS:B<x>:MDIO:DATA:SOUrce, 2-255
BUS:B<x>:MDIO:DATA:THReshold, 2-256
BUS:B<x>:MIL1553B:LOWTHRESHold, 2-256
BUS:B<x>:MIL1553B:POLarity, 2-257
BUS:B<x>:MIL1553B:RESPonsetime:
MAXimum, 2-257
BUS:B<x>:MIL1553B:RESPonsetime:
MINimum, 2-258
BUS:B<x>:MIL1553B:SOUrce, 2-259
BUS:B<x>:MIL1553B:THRESHold, 2-259
BUS:B<x>:NRZ:BITOrder, 2-260
BUS:B<x>:NRZ:BITRate, 2-261
BUS:B<x>:NRZ:POLarity, 2-261
BUS:B<x>:NRZ:SOUrce, 2-262
BUS:B<x>:NRZ:SPMI:VERsion, 2-262
BUS:B<x>:NRZ:THReshold, 2-263
BUS:B<x>:PARallel:ALLTHResholds, 2-264
BUS:B<x>:PARallel:ALLTHResholds:APPly, 2-264
BUS:B<x>:PARallel:BIT<n>SOUrce, 2-265
BUS:B<x>:PARallel:BIT<n>SOUrce:
THReshold, 2-265
BUS:B<x>:PARallel:CLOCk:EDGE, 2-266
BUS:B<x>:PARallel:CLOCk:ISCLOCKED, 2-267
BUS:B<x>:PARallel:CLOCkSOUrce, 2-267
BUS:B<x>:PARallel:CLOCKSOUrce:
THReshold, 2-268
BUS:B<x>:PSIFIVe:BITRate, 2-269
BUS:B<x>:PSIFIVe:DATAA, 2-270
BUS:B<x>:PSIFIVe:DATAB, 2-271

BUS:B<x>:PSIFIVe:DATAFORMat, 2-271
BUS:B<x>:PSIFIVe:ECUSOURce, 2-272
BUS:B<x>:PSIFIVe:FRAMECONTrol, 2-272
BUS:B<x>:PSIFIVe:MESSaging, 2-273
BUS:B<x>:PSIFIVe:SOUrce, 2-273
BUS:B<x>:PSIFIVe:STATus, 2-274
BUS:B<x>:PSIFIVe:SYNCMODe, 2-274
BUS:B<x>:PSIFIVe:SYNCTHRESHold, 2-275
BUS:B<x>:PSIFIVe:THRESHold, 2-276
BUS:B<x>:RS232C:BITRate, 2-276
BUS:B<x>:RS232C:BITRate:CUSTom, 2-277
BUS:B<x>:RS232C:DATABits, 2-277
BUS:B<x>:RS232C:DELIMiter, 2-278
BUS:B<x>:RS232C:DISplaymode, 2-279
BUS:B<x>:RS232C:PARity, 2-279
BUS:B<x>:RS232C:POLarity, 2-280
BUS:B<x>:RS232C:SOUrce, 2-280
BUS:B<x>:RS232C:SOUrce:THReshold, 2-281
BUS:B<x>:S8B10B:BITRate, 2-282
BUS:B<x>:S8B10B:SOUrce, 2-282
BUS:B<x>:S8B10B:THReshold, 2-283
BUS:B<x>:SDLC:BITRate, 2-283
BUS:B<x>:SDLC:DATA:SOUrce, 2-284
BUS:B<x>:SDLC:DATA:THReshold, 2-285
BUS:B<x>:SDLC:ENCoding, 2-285
BUS:B<x>:SDLC:MODulo, 2-286
BUS:B<x>:SENT:CHANWidth, 2-286
BUS:B<x>:SENT:NIBBLECount, 2-287
BUS:B<x>:SENT:NUMCHANnel, 2-287
BUS:B<x>:SENT:PAUSEPULSe, 2-288
BUS:B<x>:SENT:POLARITY, 2-288
BUS:B<x>:SENT:SLOW, 2-289
BUS:B<x>:SENT:SOUrce, 2-290
BUS:B<x>:SENT:THRESHold, 2-290
BUS:B<x>:SENT:TICKTIME, 2-291
BUS:B<x>:SENT:TICKTOLerance, 2-291
BUS:B<x>:SPACEWIRe:BITRate, 2-292
BUS:B<x>:SPACEWIRe:DATa:SOUrce, 2-293
BUS:B<x>:SPACEWIRe:DATa:THReshold, 2-293
BUS:B<x>:SPACEWIRe:DECode:TYPe, 2-294
BUS:B<x>:SPACEWIRe:STRobe:SOUrce, 2-295
BUS:B<x>:SPACEWIRe:STRobe:THReshold, 2-295
BUS:B<x>:SPACEWIRe:SYNC, 2-296
BUS:B<x>:SPACEWIRe:SYNC:COUnt, 2-297
BUS:B<x>:SPACEWIRe:SYNC:PATTern, 2-297
BUS:B<x>:SPACEWIRe:SYNC:VALUe, 2-298
BUS:B<x>:SPI:BITOrder, 2-298
BUS:B<x>:SPI:CLOCk:POLarity, 2-299
BUS:B<x>:SPI:CLOCk:SOUrce, 2-300
BUS:B<x>:SPI:CLOCk:THReshold, 2-300
BUS:B<x>:SPI:DATa:POLarity, 2-301
BUS:B<x>:SPI:DATa:SIZe, 2-301
BUS:B<x>:SPI:DATa:SOUrce, 2-302
BUS:B<x>:SPI:DATa:THReshold, 2-303
BUS:B<x>:SPI:FRAMING, 2-303
BUS:B<x>:SPI:IDLETime, 2-304
BUS:B<x>:SPI:MISo:DATa:POLarity, 2-304
BUS:B<x>:SPI:MISo:INPut, 2-305
BUS:B<x>:SPI:MISo:THReshold, 2-306
BUS:B<x>:SPI:MOSi:DATa:POLarity, 2-306
BUS:B<x>:SPI:MOSi:INPut, 2-307
BUS:B<x>:SPI:MOSi:THReshold, 2-307
BUS:B<x>:SPI:NUMBer:INputs, 2-308
BUS:B<x>:SPI:SELect:POLarity, 2-308
BUS:B<x>:SPI:SELect:SOUrce, 2-309
BUS:B<x>:SPI:SELect:THReshold, 2-310
BUS:B<x>:SPMI:SCLk:SOUrce, 2-310
BUS:B<x>:SPMI:SCLk:THReshold, 2-311
BUS:B<x>:SPMI:SDATa:SOUrce, 2-312
BUS:B<x>:SPMI:SDATa:THReshold, 2-313
BUS:B<x>:SVID:ALERT:SOUrce, 2-313
BUS:B<x>:SVID:ALERT:THReshold, 2-314
BUS:B<x>:SVID:CLOCk:SOUrce, 2-314
BUS:B<x>:SVID:CLOCk:THReshold, 2-315
BUS:B<x>:SVID:DATA:SOUrce, 2-316
BUS:B<x>:SVID:DATA:THReshold, 2-316
BUS:B<x>:TYPe, 2-317
BUS:B<x>:USB:BITRate, 2-318
BUS:B<x>:USB:DATAMINUSTHRESHold, 2-318
BUS:B<x>:USB:DATAPLUSTHRESHold, 2-319
BUS:B<x>:USB:LOWTHRESHold, 2-320
BUS:B<x>:USB:SIGNALTYpe, 2-320
BUS:B<x>:USB:SOUrce, 2-321
BUS:B<x>:USB:SOUrce:DMINus, 2-321
BUS:B<x>:USB:SOUrce:DPLUs, 2-322
BUS:B<x>:USB:THRESHold, 2-323
BUS:B<x>PSIFIVe:BITPERiod, 2-268
BUS:B<x>PSIFIVe:COMM:DIRection, 2-269
BUS:DELete, 2-324
BUS:LIST?, 2-324
BUSTABle:ADDNew, 2-324
BUSTABle:DELete, 2-325
BUSTABle:LIST?, 2-325
BUSY?, 2-325

C

*CAL?, 2-326
CALibrate?, 2-326
CALibrate:INTERNal, 2-327
CALibrate:INTERNal:STARt, 2-327
CALibrate:INTERNal:STATus?, 2-328
CALibrate:PWRUpstatus?, 2-329
CALLOUTS:CALLOUT<x>, 2-329
CALLOUTS:CALLOUT<x>:BOOKMark:SOURCE, 2-329
CALLOUTS:CALLOUT<x>:BOOKMark:XPOS, 2-330
CALLOUTS:CALLOUT<x>:COLOR, 2-330
CALLOUTS:CALLOUT<x>:FONT:BOLD, 2-331
CALLOUTS:CALLOUT<x>:FONT:ITALIC, 2-331
CALLOUTS:CALLOUT<x>:FONT:SIZE, 2-332
CALLOUTS:CALLOUT<x>:FONT:TYPE, 2-332
CALLOUTS:CALLOUT<x>:FONT:UNDERLine, 2-333
CALLOUTS:CALLOUT<x>:TEXT, 2-333
CALLOUTS:CALLOUT<x>:TYPE, 2-333
CH<x>?, 2-334
CH<x>:BANdwidth, 2-334
CH<x>:BANdwidth:FILTer:OPTIMIZation, 2-335
CH<x>:CLIPping?, 2-336
CH<x>:COUPling, 2-336
CH<x>_D<x>:LABel:COLor, 2-364
CH<x>_D<x>:LABel:FONT:BOLD, 2-365
CH<x>_D<x>:LABel:FONT:ITALic, 2-365
CH<x>_D<x>:LABel:FONT:SIZE, 2-366
CH<x>_D<x>:LABel:FONT:TYPE, 2-366
CH<x>_D<x>:LABel:FONT:UNDERline, 2-367
CH<x>_D<x>:LABel:NAMe, 2-367
CH<x>_DALL:LABel:COLor, 2-361
CH<x>_DALL:LABel:FONT:BOLD, 2-361
CH<x>_DALL:LABel:FONT:ITALic, 2-362
CH<x>_DALL:LABel:FONT:SIZE, 2-362
CH<x>_DALL:LABel:FONT:TYPE, 2-363
CH<x>_DALL:LABel:FONT:UNDERline, 2-363
CH<x>_DALL:LABel:NAMe, 2-364
CH<x>:DESKew, 2-337
CH<x>:LABel:COLor, 2-337
CH<x>:LABel:FONT:BOLD, 2-338
CH<x>:LABel:FONT:ITALic, 2-338
CH<x>:LABel:FONT:SIZE, 2-339
CH<x>:LABel:FONT:TYPE, 2-339
CH<x>:LABel:FONT:UNDERline, 2-340
CH<x>:LABel:NAMe, 2-340
CH<x>:LABel:XPOS, 2-341
CH<x>:LABel:YPOS, 2-341
CH<x>:OFFSet, 2-342
CH<x>:POSition, 2-342
CH<x>:PRObe?, 2-343
CH<x>:PRObe:AUTOZero, 2-343
CH<x>:PRObe:COMPensate, 2-344
CH<x>:PRObe:DEGAUSS, 2-344
CH<x>:PRObe:DEGAUSS:STATE?, 2-344
CH<x>:PRObe:FORCEDRange, 2-345
CH<x>:PRObe:GAIN?, 2-345
CH<x>:PRObe:ID?, 2-346
CH<x>:PRObe:ID:SERnumber?, 2-346
CH<x>:PRObe:ID:TYPe?, 2-346
CH<x>:PRObe:INPUTMode, 2-347
CH<x>:PRObe:INPUTMode:AOFFSet, 2-347
CH<x>:PRObe:INPUTMode:BOFFSet, 2-348
CH<x>:PRObe:INPUTMode:COFFSet, 2-349
CH<x>:PRObe:INPUTMode:DOFFSet, 2-349
CH<x>:PRObe:RESistance?, 2-350
CH<x>:PRObe:SET, 2-350
CH<x>:PRObe:STATus?, 2-351
CH<x>:PRObe:UNIts?, 2-351
CH<x>:PROBECal?, 2-352
CH<x>:PROBECOntrol, 2-352
CH<x>:PROBEFunc:EXTAtten, 2-353
CH<x>:PROBEFunc:EXTDBatten, 2-354
CH<x>:PROBEFunc:EXTUnits, 2-354
CH<x>:PROBEFunc:EXTUnits:STATE, 2-355
CH<x>:SCAle, 2-355
CH<x>:SCALERATio, 2-356
CH<x>:SV:CENTERFrequency, 2-356
CH<x>:SV:POSition, 2-357
CH<x>:SV:SPANABovebw?, 2-357
CH<x>:SV:SPANBELowdc?, 2-358
CH<x>:SV:STARTFrequency?, 2-358
CH<x>:SV:STATE, 2-359
CH<x>:SV:STOPFrequency?, 2-359
CH<x>:TERmination, 2-360
CH<x>:VTERm:BIAS, 2-360
CLEAR, 2-367
*CLS, 2-368
Command Groups, 2-11
CONFIGuration:ANALOg:BANDWidth?, 2-369
CONNected:REQUested:STATus , 2-369
CONNected:STATus?, 2-369

CONNected:USAGe:TRack:REQUested:
STATus, 2-370
CONNected:USAGe:TRack:STATus?, 2-370
CURVe, 2-371
CUSTOMTABle:ADDNew, 2-373
CUSTOMTABle:DELete, 2-374
CUSTOMTABle:LIST?, 2-374

D

DATa, 2-374
DATa:ENCdg, 2-375
DATa:RESample, 2-378
DATa:SOUrce, 2-378
DATa:SOUrce:AVAILable?, 2-379
DATa:STARt, 2-380
DATa:STOP, 2-380
DATa:WIDth, 2-381
DATE?, 2-382
*DDT, 2-383
DESE, 2-383
DIAg:LOOP:OPTion, 2-384
DIAg:LOOP:OPTion:NTIMes, 2-385
DIAg:LOOP:STOP, 2-385
DIAg:MODe, 2-385
DIAg:RESUlt?, 2-386
DIAg:RESUlt:FLAg?, 2-386
DIAg:RESUlt:LOG?, 2-387
DIAg:SELect, 2-387
DIAg:STATE, 2-387
DIGGRP<x>:D<x>:THReshold, 2-388
DISplay?, 2-388
DISplay:{CH<x>|Math<x>|REF<x>}:
INVERTColor, 2-389
DISplay:{CH<x>|Math<x>|REF<x>}:
NORMALColor, 2-390
DISplay:COLors, 2-390
DISplay:GLOBal:B<x>:STATE, 2-391
DISplay:GLOBal:CH<x>:STATE, 2-391
DISplay:GLOBal:MATH<x>:STATE, 2-392
DISplay:GLOBal:PLOT<x>:STATE, 2-393
DISplay:GLOBal:REF<x>:STATE, 2-393
DISplay:INTENSITy?, 2-394
DISplay:INTENSITy:BACKLight, 2-394
DISplay:INTENSITy:BACKLight:AUTODim:
ENAble, 2-395
DISplay:INTENSITy:BACKLight:AUTODim:
TIMe, 2-395
DISplay:MATHFFTView<n>:CURSor:
ROLOCATION, 2-396
DISplay:MATHFFTView<x>:AUTOScale, 2-396
DISplay:MATHFFTView<x>:CURSor:
ASOUrce?, 2-397
DISplay:MATHFFTView<x>:CURSor:
BSOUrce?, 2-397
DISplay:MATHFFTView<x>:CURSor:DDT?, 2-398
DISplay:MATHFFTView<x>:CURSor:
FUNCtion, 2-398
DISplay:MATHFFTView<x>:CURSor:HBArs:
APOSition, 2-399
DISplay:MATHFFTView<x>:CURSor:HBArs:
AUNIts?, 2-399
DISplay:MATHFFTView<x>:CURSor:HBArs:
BPOSition, 2-400
DISplay:MATHFFTView<x>:CURSor:HBArs:
BUNIts?, 2-400
DISplay:MATHFFTView<x>:CURSor:HBArs:
DELTa?, 2-401
DISplay:MATHFFTView<x>:CURSor:MODe, 2-401
DISplay:MATHFFTView<x>:CURSor:
ONEOVERDELTATVALUE?, 2-402
DISplay:MATHFFTView<x>:CURSor:SCREEN:
AXPOSition, 2-402
DISplay:MATHFFTView<x>:CURSor:SCREEN:
AYPOSition, 2-403
DISplay:MATHFFTView<x>:CURSor:SCREEN:
BXPOSition, 2-403
DISplay:MATHFFTView<x>:CURSor:SCREEN:
BYPOSition, 2-404
DISplay:MATHFFTView<x>:CURSor:STATE, 2-404
DISplay:MATHFFTView<x>:CURSor:VBArs:
APOSition, 2-405
DISplay:MATHFFTView<x>:CURSor:VBArs:
AUNIts?, 2-405
DISplay:MATHFFTView<x>:CURSor:VBArs:
BPOSition, 2-406
DISplay:MATHFFTView<x>:CURSor:VBArs:
BUNIts?, 2-406
DISplay:MATHFFTView<x>:CURSor:VBArs:
DELTa?, 2-407
DISplay:MATHFFTView<x>:CURSor:WAVEform:
APOSition, 2-407
DISplay:MATHFFTView<x>:CURSor:WAVEform:
BPOSition, 2-407
DISplay:MATHFFTView<x>:GRIDlines, 2-408

DISplay:MATHFFTView<x>:MATH:MATH<x>:
STATE, 2-409
DISplay:MATHFFTView<x>:XAXIS:SCALE, 2-409
DISplay:MATHFFTView<x>:YAXIS:SCALE, 2-410
DISplay:MATHFFTView<x>:ZOOM:XAXIS:
FROM, 2-410
DISplay:MATHFFTView<x>:ZOOM:XAXIS:
TO, 2-411
DISplay:MATHFFTView<x>:ZOOM:YAXIS:
FROM, 2-411
DISplay:MATHFFTView<x>:ZOOM:YAXIS:
TO, 2-412
DISplay:PERSistence, 2-412
DISplay:PERSistence:RESET, 2-413
DISplay:PLOTView<n>:CURSor:
ROLOCATION, 2-413
DISplay:PLOTView<x>:AUTOScale, 2-414
DISplay:PLOTView<x>:CURSor:ASOUrce?, 2-415
DISplay:PLOTView<x>:CURSor:BSOUrce?, 2-415
DISplay:PLOTView<x>:CURSor:DDT?, 2-415
DISplay:PLOTView<x>:CURSor:FUNCtion, 2-416
DISplay:PLOTView<x>:CURSor:HBArs:
APOSition, 2-416
DISplay:PLOTView<x>:CURSor:HBArs:
AUNIts?, 2-417
DISplay:PLOTView<x>:CURSor:HBArs:
BPOSition, 2-417
DISplay:PLOTView<x>:CURSor:HBArs:
BUNIts?, 2-418
DISplay:PLOTView<x>:CURSor:HBArs:
DELTa?, 2-418
DISplay:PLOTView<x>:CURSor:MODe, 2-419
DISplay:PLOTView<x>:CURSor:
ONEOVERDELTATVALUE?, 2-419
DISplay:PLOTView<x>:CURSor:SCREEN:
AXPOSition, 2-420
DISplay:PLOTView<x>:CURSor:SCREEN:
AYPOSition, 2-420
DISplay:PLOTView<x>:CURSor:SCREEN:
BXPOSition, 2-421
DISplay:PLOTView<x>:CURSor:SCREEN:
BYPOSition, 2-421
DISplay:PLOTView<x>:CURSor:
SPLITMODE, 2-422
DISplay:PLOTView<x>:CURSor:STATE, 2-422
DISplay:PLOTView<x>:CURSor:VBArs:
APOSition, 2-423
DISplay:PLOTView<x>:CURSor:VBArs:
BPOSition, 2-423
DISplay:PLOTView<x>:CURSor:VBArs:
DELTa?, 2-424
DISplay:PLOTView<x>:CURSor:VBArs:
UNIts?, 2-424
DISplay:PLOTView<x>:CURSor:WAVEform:
APOSition, 2-425
DISplay:PLOTView<x>:CURSor:WAVEform:
BPOSition, 2-425
DISplay:PLOTView<x>:GRIDlines, 2-426
DISPLAY:PLOTVIEW<x>:XAXIS:SCALE, 2-427
DISPLAY:PLOTVIEW<x>:YAXIS:SCALE, 2-427
DISplay:PLOTView<x>:ZOOM:XAXIS:
FROM, 2-428
DISplay:PLOTView<x>:ZOOM:XAXIS:TO, 2-428
DISplay:PLOTView<x>:ZOOM:YAXIS:
FROM, 2-429
DISplay:PLOTView<x>:ZOOM:YAXIS:TO, 2-429
DISplay:REFFFTView<n>:CURSor:
ROLOCATION, 2-430
DISplay:REFFFTView<x>:AUTOScale, 2-430
DISplay:REFFFTView<x>:CURSor:
ASOUrce?, 2-431
DISplay:REFFFTView<x>:CURSor:
BSOUrce?, 2-431
DISplay:REFFFTView<x>:CURSor:DDT?, 2-432
DISplay:REFFFTView<x>:CURSor:
FUNCtion, 2-432
DISplay:REFFFTView<x>:CURSor:HBArs:
APOSition, 2-433
DISplay:REFFFTView<x>:CURSor:HBArs:
AUNIts?, 2-433
DISplay:REFFFTView<x>:CURSor:HBArs:
BPOSition, 2-434
DISplay:REFFFTView<x>:CURSor:HBArs:
BUNIts?, 2-434
DISplay:REFFFTView<x>:CURSor:HBArs:
DELTa?, 2-434
DISplay:REFFFTView<x>:CURSor:MODe, 2-435
DISplay:REFFFTView<x>:CURSor:
ONEOVERDELTATVALUE?, 2-435
DISplay:REFFFTView<x>:CURSor:SCREEN:
AXPOSition, 2-436
DISplay:REFFFTView<x>:CURSor:SCREEN:
AYPOSition, 2-436

DISplay:REFFFTView<x>:CURSor:SCREEN:
BXPOSition, 2-437
DISplay:REFFFTView<x>:CURSor:SCREEN:
BYPOSition, 2-437
DISplay:REFFFTView<x>:CURSor:
SPLITMODE, 2-438
DISplay:REFFFTView<x>:CURSor:STATE, 2-438
DISplay:REFFFTView<x>:CURSor:VBArs:
APOSition, 2-439
DISplay:REFFFTView<x>:CURSor:VBArs:
BPOSition, 2-439
DISplay:REFFFTView<x>:CURSor:VBArs:
DELTa?, 2-440
DISplay:REFFFTView<x>:CURSor:VBArs:
UNIts?, 2-440
DISplay:REFFFTView<x>:CURSor:WAVEform:
AHPOSition?, 2-441
DISplay:REFFFTView<x>:CURSor:WAVEform:
APOSition, 2-441
DISplay:REFFFTView<x>:CURSor:WAVEform:
AVPOSition?, 2-441
DISplay:REFFFTView<x>:CURSor:WAVEform:
BHPOSition?, 2-442
DISplay:REFFFTView<x>:CURSor:WAVEform:
BPOSition, 2-442
DISplay:REFFFTView<x>:CURSor:WAVEform:
BVPOSition?, 2-443
DISplay:REFFFTView<x>:GRIDlines, 2-443
DISplay:REFFFTView<x>:REF:REF<x>:
STATE, 2-444
DISplay:REFFFTView<x>:XAXIS:SCALE, 2-444
DISplay:REFFFTView<x>:ZOOM:XAXIS:
FROM, 2-445
DISplay:REFFFTView<x>:ZOOM:XAXIS:TO, 2-445
DISplay:REFFFTView<x>:ZOOM:YAXIS:
FROM, 2-446
DISplay:REFFFTView<x>:ZOOM:YAXIS:TO, 2-446
DISplay:SELect:BUS, 2-447
DISplay:SELect:MATH, 2-447
DISplay:SELect:REFerence, 2-448
DISplay:SELect:SOUrce, 2-448
DISplay:SELect:SPECView<x>:SOUrce, 2-448
DISplay:SELect:VIEW, 2-449
DISplay:SELect:WAVEView<x>:SOUrce, 2-449
DISplay:SPECView<x>:CURSor:CURSOR:
ANOISEDensity?, 2-450
DISplay:SPECView<x>:CURSor:CURSOR:
ASOUrce, 2-450
DISplay:SPECView<x>:CURSor:CURSOR:
BNOISE?, 2-451
DISplay:SPECView<x>:CURSor:CURSOR:
BSOUrce, 2-451
DISplay:SPECView<x>:CURSor:CURSOR:HBArs:
APOSition, 2-452
DISplay:SPECView<x>:CURSor:CURSOR:HBArs:
AUNIts?, 2-452
DISplay:SPECView<x>:CURSor:CURSOR:HBArs:
BPOSition, 2-453
DISplay:SPECView<x>:CURSor:CURSOR:HBArs:
BUNIts?, 2-453
DISplay:SPECView<x>:CURSor:CURSOR:
READout, 2-454
DISplay:SPECView<x>:CURSor:CURSOR:
SPLITMODE, 2-455
DISplay:SPECView<x>:CURSor:CURSOR:
STATE, 2-455
DISplay:SPECView<x>:CURSor:CURSOR:VBArs:
APOSition, 2-456
DISplay:SPECView<x>:CURSor:CURSOR:VBArs:
BPOSition, 2-456
DISplay:SPECView<x>:CURSor:CURSOR:VBArs:
UNIts?, 2-457
DISplay:SPECView<x>:CURSor:CURSOR:
WAVEform:APOSition, 2-457
DISplay:SPECView<x>:CURSor:CURSOR:
WAVEform:BPOSition, 2-458
DISplay:SPECView<x>:GRAticule, 2-459
DISplay:SPECView<x>:HORZ, 2-458
DISplay:SPECView<x>:INTENSITy:
GRATicule, 2-459
DISplay:SPECView<x>:INTENSITy:
WAVEform, 2-460
DISplay:SPECView<x>:VIEWStyle, 2-461
DISplay:VARpersist, 2-461
DISplay:WAVEform, 2-462
DISplay:WAVEView:CURSor:CURSOR1:
ROLOCATION, 2-462
DISplay:WAVEView:GRIDTYPE, 2-463
DISplay:WAVEView<x>:BUS:B<x>:STATE, 2-463
DISplay:WAVEView<x>:BUS:B<x>:VERTical:
POSition, 2-464
DISplay:WAVEView<x>:CH<x>_D<x>:
STATE, 2-467

DISplay:WAVEView<x>:CH<x>_DALL:
STATE, 2-466
DISplay:WAVEView<x>:CH<x>_DALL:VERTical:
POSition, 2-467
DISplay:WAVEView<x>:CH<x>:STATE, 2-464
DISplay:WAVEView<x>:CH<x>:VERTical:
POSition, 2-465
DISplay:WAVEView<x>:CH<x>:VERTical:
SCAle, 2-465
DISplay:WAVEView<x>:CURSor?, 2-468
DISplay:WAVEView<x>:CURSor:CURSOR:
WAVEform:AVPOSition?, 2-482
DISplay:WAVEView<x>:CURSor:CURSOR:
WAVEform:BVPOSition?, 2-483
DISplay:WAVEView<x>:CURSor:
CURSOR<x>?, 2-468
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
ASOUrce, 2-469
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
BSOUrce, 2-470
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
DDT?, 2-470
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
FUNCtion, 2-471
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArs:APOSition, 2-471
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArs:AUNIts?, 2-472
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArs:BPOSition, 2-473
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArs:BUNIts?, 2-473
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
HBArs:DELTa?, 2-474
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
MODe, 2-474
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
ONEOVERDELTATVALUE?, 2-475
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:AXPOSition, 2-475
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:AYPOSition, 2-476
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:BXPOSition, 2-477
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SCREEN:BYPOSition, 2-477
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
SPLITMODE, 2-478
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
STATE, 2-478
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBArs:APOSition, 2-479
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBArs:BPOSition, 2-480
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBArs:DELTa?, 2-480
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
VBArs:UNIts?, 2-481
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
WAVEform:APOSition, 2-481
DISplay:WAVEView<x>:CURSor:CURSOR<x>:
WAVEform:BPOSition, 2-482
DISplay:WAVEView<x>:FILTer, 2-483
DISplay:WAVEView<x>:GRAticule, 2-484
DISplay:WAVEView<x>:INTENSITy:
GRATicule, 2-484
DISplay:WAVEView<x>:INTENSITy:
WAVEform, 2-485
DISplay:WAVEView<x>:MATH:MATH<x>:
AUTOScale, 2-485
DISplay:WAVEView<x>:MATH:MATH<x>:
STATE, 2-486
DISplay:WAVEView<x>:MATH:MATH<x>:
VERTical:POSition, 2-487
DISplay:WAVEView<x>:MATH:MATH<x>:
VERTical:SCAle, 2-487
DISplay:WAVEView<x>:PLOT:PLOT<x>:
AUTOScale, 2-488
DISplay:WAVEView<x>:PLOT:PLOT<x>:
STATE, 2-488
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:
POSition, 2-489
DISplay:WAVEView<x>:PLOT:PLOT<x>:VERTical:
SCAle, 2-489
DISplay:WAVEView<x>:REF:REF<x>:
STATE, 2-490
DISplay:WAVEView<x>:REF:REF<x>:VERTical:
POSition, 2-490
DISplay:WAVEView<x>:REF:REF<x>:VERTical:
SCAle, 2-491
DISplay:WAVEView<x>:RF_FREQuency<x>:
VERTical:POSition, 2-491

DISplay:WAVEView<x>:RF_FREQuency<x>:
VERTical:SCAle, 2-492
DISplay:WAVEView<x>:RF_MAGnitude<x>:
VERTical:POSition, 2-493
DISplay:WAVEView<x>:RF_MAGnitude<x>:
VERTical:SCAle, 2-493
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:
POSition, 2-494
DISplay:WAVEView<x>:RF_PHASe<x>:VERTical:
SCAle, 2-495
DISplay:WAVEView<x>:STYle, 2-495
DISplay:WAVEView<x>:VIEWStyle, 2-496
DISplay:WAVEView<x>:Zoom?, 2-496
DISplay:WAVEView<x>:ZOOM:ZOOM<x>?, 2-497
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
HORizontal:POSition, 2-497
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
HORizontal:SCALe, 2-498
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
HORizontal:WINSCALe, 2-499
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
STATe, 2-499
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
VERTical:POSition, 2-500
DISplay:WAVEView<x>:ZOOM:ZOOM<x>:
VERTical:SCALe, 2-501
DISplay:WAVEView<y>:REF:REF<x>:
FRAMe, 2-501
DISplay:WAVEView<y>:REF<x>_DALL:
FRAMe, 2-502
DVM, 2-502
DVM:AUTORange, 2-503
DVM:MEASUrement:FREQuency?, 2-503
DVM:MEASUrement:HIStory:AVErage?, 2-504
DVM:MEASUrement:HIStory:MAXimum?, 2-504
DVM:MEASUrement:HIStory:MINImum?, 2-504
DVM:MEASUrement:INFMAXimum?, 2-505
DVM:MEASUrement:INFMINimum?, 2-505
DVM:MEASUrement:VALue?, 2-506
DVM:MODe, 2-506
DVM:SOUrce, 2-507
DVM:TRIGger:FREQuency:COUNTer, 2-507

E

*ESE, 2-508
*ESR?, 2-509
ETHERnet:DHCPbootp, 2-509
ETHERnet:DNS:IPADDress, 2-510
ETHERnet:DOMAINname, 2-510
ETHERnet:ENET:ADDress?, 2-511
ETHERnet:GATEWay:IPADDress, 2-511
ETHERnet:IPADDress, 2-511
ETHERnet:LXI:LAN:RESET, 2-512
ETHERnet:LXI:LAN:SERVICENAMe, 2-512
ETHERnet:LXI:LAN:STATus?, 2-513
ETHERnet:NAME, 2-513
ETHERnet:NETWORKCONFig, 2-513
ETHERnet:PING, 2-514
ETHERnet:PING:STATus?, 2-514
ETHERnet:SUBNETMask, 2-515
EVENT?, 2-515
EVMsg?, 2-516
EVQty?, 2-517
EYEMASK:MASK<x>:COUNt:HITS?, 2-517
EYEMASK:MASK<x>:COUNt:SEG<y>:
HITS?, 2-518
EYEMASK:MASK<x>:CREATor?, 2-518
EYEMASK:MASK<x>:ENAbled, 2-519
EYEMASK:MASK<x>:MASKfile, 2-520
EYEMASK:MASK<x>:TESt:SAMple:
THReshold, 2-520
EYEMASK:MASK<x>:TESt:STATUS?, 2-521

F

FACtory, 2-522
FILESystem?, 2-522
FILESystem:COPy, 2-523
FILESystem:CWD, 2-524
FILESystem:DELEte, 2-524
FILESystem:DIR?, 2-525
FILESystem:HOMEDir?, 2-526
FILESystem:LDIR?, 2-526
FILESystem:MKDir, 2-526
FILESYSTEM:MOUNT:DRIVE, 2-527
FILESystem:READFile, 2-528
FILESystem:REName, 2-528
FILESystem:RMDir, 2-529
FILESystem:UNMOUNT:DRIve, 2-529
FILESystem:WRITEFile, 2-530
FPAnel:PRESS, 2-530
FPAnel:TURN, 2-531

H

HEADer, 2-531
HORizontal?, 2-532
HORizontal:ACQDURATION?, 2-533
HORizontal:DELay:MODe, 2-533
HORizontal:DELay:TIMe, 2-534
HORizontal:DIVisions?, 2-534
HORizontal:FASTframe?, 2-534
HORizontal:FASTframe:COUNt, 2-535
HORizontal:FASTframe:MAXFRames?, 2-535
HORizontal:FASTframe:MULtipleframes:MODe, 2-535
HORizontal:FASTframe:REF:FRAme, 2-536
HORizontal:FASTframe:REF:INCLUde, 2-536
HORizontal:FASTframe:SELECTED, 2-537
HORizontal:FASTframe:STATE, 2-537
HORizontal:FASTframe:SUMFrame?, 2-538
HORizontal:FASTframe:SUMFrame:STATE, 2-538
HORizontal:FASTframe:TIMEStamp:ALL?, 2-539
HORizontal:FASTframe:TIMEStamp:DELTa?, 2-539
HORizontal:FASTframe:TIMEStamp:REFerence?, 2-540
HORizontal:FASTframe:TIMEStamp:SELECTED?, 2-540
HORizontal:FASTframe:XZEro:ALL?, 2-540
HORizontal:FASTframe:XZEro:REF?, 2-541
HORizontal:FASTframe:XZEro:SELECTED?, 2-541
HORizontal:MAIn:INTERPRatio?, 2-541
HORizontal:MODE, 2-542
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:VALue, 2-542
HORizontal:MODe:AUTOmatic:FASTAcq:RECOrdlength:MAXimum:ZOOMOVERride, 2-543
HORizontal:MODe:MANual:CONFIGure, 2-544
HORizontal:MODE:RECOrdlength, 2-544
HORizontal:MODE:SAMPLERate, 2-545
HORizontal:MODE:SCAle, 2-545
HORizontal:POSition, 2-545
HORizontal:PREViewstate?, 2-546
HORizontal:RECOrdlength, 2-546
HORizontal:ROLL?, 2-547
HORizontal:SAMPLERate, 2-547
HORizontal:SAMPLERate:ANALYZemode:MINimum:OVERRide, 2-548
HORizontal:SAMPLERate:ANALYZemode:MINimum:VALue, 2-548
HORizontal:SCAle, 2-549

I

ID?, 2-549
*IDN?, 2-550

L

LIC:UNINSTALL?, 2-550
LICense?, 2-551
LICENSE:APPID?, 2-551
LICense:COUNt?, 2-551
LICense:ERRor?, 2-552
LICense:GMT?, 2-552
LICense:HID?, 2-553
LICense:INSTall, 2-553
LICense:ITEM?, 2-553
LICense:LIST?, 2-554
LICense:VALidate?, 2-554
LOCk, 2-555
*LRN?, 2-555

M

MAINWindow:BADGe:BRINgtoview, 2-556
MAINWindow:FONTSize, 2-557
MAINWindow:RRBDisplaystate, 2-557
MASK:DELete, 2-558
MASK:MASK<x>:COUNT?, 2-558
MASK:MASK<x>:COUNT:HITS?, 2-559
MASK:MASK<x>:DEFinedby, 2-559
MASK:MASK<x>:DISplay, 2-560
MASK:MASK<x>:LIST?, 2-560
MASK:MASK<x>:SEG<x>:POINTS, 2-561
MASK:MASK<x>:SEG<x>COUNT:HITS?, 2-561
MASK:MASK<x>:SOUrce, 2-562
MASK:MASK<x>:TESt:STATE, 2-563
MASK:MASK<x>:TESt:STATUS?, 2-563
MASK:MASK<x>:TEST:THReshold, 2-564
MASK:MASK<x>:TOLerance:HORizontal, 2-564
MASK:MASK<x>:TOLerance:UPDatenow, 2-565
MASK:MASK<x>:TOLerance:VERTical, 2-565
MASK:TESt:WAVEforms, 2-566
MATH:ADDNew, 2-567
MATH:DELete, 2-567
MATH:LIST?, 2-567
MATH:MATH<x>:AVG:MODE, 2-568
MATH:MATH<x>:AVG:WEIGht, 2-568

MATH:MATH<x>:DEFine, 2-569
MATH:MATH<x>:FUNCtion, 2-570
MATH:MATH<x>:GATing, 2-570
MATH:MATH<x>:LABel:COLor, 2-571
MATH:MATH<x>:LABel:FONT:BOLD, 2-571
MATH:MATH<x>:LABel:FONT:ITALic, 2-572
MATH:MATH<x>:LABel:FONT:SIZE, 2-572
MATH:MATH<x>:LABel:FONT:TYPE, 2-573
MATH:MATH<x>:LABel:FONT:UNDERline, 2-573
MATH:MATH<x>:LABel:NAMe, 2-574
MATH:MATH<x>:LABel:XPOS, 2-574
MATH:MATH<x>:LABel:YPOS, 2-575
MATH:MATH<x>:SOUrce<x>, 2-575
MATH:MATH<x>:SPECTral:HORZ, 2-576
MATH:MATH<x>:SPECTral:MAG, 2-576
MATH:MATH<x>:SPECTral:PHASE, 2-577
MATH:MATH<x>:SPECTral:SOUrce, 2-577
MATH:MATH<x>:SPECTral:SUPPress, 2-578
MATH:MATH<x>:SPECTral:SUPPress:
VALue, 2-579
MATH:MATH<x>:SPECTral:TYPE, 2-579
MATH:MATH<x>:SPECTral:UNWRap, 2-580
MATH:MATH<x>:SPECTral:UNWRap:
DEGrees, 2-580
MATH:MATH<x>:SPECTral:WINdow, 2-581
MATH:MATH<x>:TYPe, 2-582
MATH:MATH<x>:VUNIT, 2-582
MATHArbflt<x>:FILepath, 2-583
MEASTABle:ADDNew, 2-583
MEASTABle:DELETE, 2-584
MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:
MAXimum?, 2-584
MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:
MEAN?, 2-585
MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:
MINimum?, 2-586
MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:
PK2PK?, 2-586
MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:
POPUlation?, 2-587
MEASU:MEAS1:SUBGROUP:RESUlts:ALLAcqs:
STDDev?, 2-588
MEASU:MEAS1:SUBGROUP:RESUlts:
CURRentacq:MAXimum?, 2-588
MEASU:MEAS1:SUBGROUP:RESUlts:
CURRentacq:MEAN?, 2-589
MEASU:MEAS1:SUBGROUP:RESUlts:
CURRentacq:MINimum?, 2-590
MEASU:MEAS1:SUBGROUP:RESUlts:
CURRentacq:PK2PK?, 2-591
MEASU:MEAS1:SUBGROUP:RESUlts:
CURRentacq:POPUlation?, 2-592
MEASU:MEAS1:SUBGROUP:RESUlts:
CURRentacq:STDDev?, 2-593
MEASUrement?, 2-593
MEASUrement:ADDMEAS , 2-594
MEASUrement:ADDNew, 2-603
MEASUrement:ANNOTate, 2-603
MEASUrement:AUTOset DPMAutoset, 2-604
MEASUrement:AUTOset DPMPReset, 2-605
MEASUrement:AUTOset, 2-604
MEASUrement:CH<x>:REFLevels:ABSolute:
FALLHigh, 2-605
MEASUrement:CH<x>:REFLevels:ABSolute:
FALLLow, 2-605
MEASUrement:CH<x>:REFLevels:ABSolute:
FALLMid, 2-606
MEASUrement:CH<x>:REFLevels:ABSolute:
HYSTeresis, 2-606
MEASUrement:CH<x>:REFLevels:ABSolute:
RISEHigh, 2-607
MEASUrement:CH<x>:REFLevels:ABSolute:
RISELow, 2-607
MEASUrement:CH<x>:REFLevels:ABSolute:
RISEMid, 2-608
MEASUrement:CH<x>:REFLevels:ABSolute:
TYPE, 2-608
MEASUrement:CH<x>:REFLevels:BASETop, 2-609
MEASUrement:CH<x>:REFLevels:METHod, 2-610
MEASUrement:CH<x>:REFLevels:PERCent:
FALLHigh, 2-610
MEASUrement:CH<x>:REFLevels:PERCent:
FALLLow, 2-611
MEASUrement:CH<x>:REFLevels:PERCent:
FALLMid, 2-611
MEASUrement:CH<x>:REFLevels:PERCent:
HYSTeresis, 2-612
MEASUrement:CH<x>:REFLevels:PERCent:
RISEHigh, 2-612
MEASUrement:CH<x>:REFLevels:PERCent:
RISELow, 2-613
MEASUrement:CH<x>:REFLevels:PERCent:
RISEMid, 2-613

MEASUrement:CH<x>:REFLevels:PERCent:
TYPE, 2-614
MEASUrement:CLOCKRecovery:ADVanced:
METHod, 2-615
MEASUrement:CLOCKRecovery:
CLOCKFrequency, 2-615
MEASUrement:CLOCKRecovery:
CLOCKMultiplier, 2-616
MEASUrement:CLOCKRecovery:
CONSTCLOCKMODe, 2-616
MEASUrement:CLOCKRecovery:DAMPing, 2-617
MEASUrement:CLOCKRecovery:DATAPath, 2-617
MEASUrement:CLOCKRecovery:DATARate, 2-618
MEASUrement:CLOCKRecovery:
EXPLICITCLOCKMODe, 2-618
MEASUrement:CLOCKRecovery:
JTFBandwidth, 2-619
MEASUrement:CLOCKRecovery:
LOOPBandwidth, 2-619
MEASUrement:CLOCKRecovery:
MEANAUTOCalculate, 2-620
MEASUrement:CLOCKRecovery:METHod, 2-620
MEASUrement:CLOCKRecovery:MODel, 2-621
MEASUrement:CLOCKRecovery:
NOMINALOFFset, 2-621
MEASUrement:CLOCKRecovery:
NOMINALOFFset:SELECTIONtype, 2-622
MEASUrement:CLOCKRecovery:STAndard, 2-622
MEASUrement:DELete , 2-623
MEASUrement:DELETEALL , 2-623
MEASUrement:DIRacmodel, 2-624
MEASUrement:DISPLAYUnits, 2-624
MEASUrement:EDGE<x>, 2-625
MEASUrement:EYERENDER, 2-625
MEASUrement:FILTers:BLANKingtime, 2-626
MEASUrement:FILTers:HIGHPass:FREQ, 2-626
MEASUrement:FILTers:HIGHPass:SPEC, 2-627
MEASUrement:FILTers:LOWPass:FREQ, 2-627
MEASUrement:FILTers:LOWPass:SPEC, 2-628
MEASUrement:FILTers:RAMPtime, 2-628
MEASUrement:GATing, 2-629
MEASUrement:GATing:ACTive, 2-629
MEASUrement:GATing:ENDtime, 2-630
MEASUrement:GATing:HYSTeresis, 2-630
MEASUrement:GATing:LOGICSource, 2-631
MEASUrement:GATing:MIDRef, 2-631
MEASUrement:GATing:SEARCHSource, 2-632
MEASUrement:GATing:STARTtime, 2-632
MEASUrement:HIGHLEVel:CONNECTion, 2-633
MEASUrement:HIGHLEVel:L2LTOL2N, 2-633
MEASUrement:HIGHLEVel:WIRing, 2-633
MEASUrement:INTERp, 2-634
MEASUrement:JITTermodel, 2-634
MEASUrement:LIST?, 2-635
MEASUrement:LOCKRJ, 2-635
MEASUrement:LOCKRJValue, 2-636
MEASUrement:MATH<x>:REFLevels:ABSolute:
FALLHigh, 2-636
MEASUrement:MATH<x>:REFLevels:ABSolute:
FALLLow, 2-637
MEASUrement:MATH<x>:REFLevels:ABSolute:
FALLMid, 2-637
MEASUrement:MATH<x>:REFLevels:ABSolute:
HYSTeresis, 2-638
MEASUrement:MATH<x>:REFLevels:ABSolute:
RISEHigh, 2-638
MEASUrement:MATH<x>:REFLevels:ABSolute:
RISELow, 2-639
MEASUrement:MATH<x>:REFLevels:ABSolute:
RISEMid, 2-640
MEASUrement:MATH<x>:REFLevels:ABSolute:
TYPE, 2-640
MEASUrement:MATH<x>:REFLevels:
BASETop, 2-641
MEASUrement:MATH<x>:REFLevels:
METHod, 2-641
MEASUrement:MATH<x>:REFLevels:PERCent:
FALLHigh, 2-642
MEASUrement:MATH<x>:REFLevels:PERCent:
FALLLow, 2-643
MEASUrement:MATH<x>:REFLevels:PERCent:
FALLMid, 2-643
MEASUrement:MATH<x>:REFLevels:PERCent:
HYSTeresis, 2-644
MEASUrement:MATH<x>:REFLevels:PERCent:
RISEHigh, 2-644
MEASUrement:MATH<x>:REFLevels:PERCent:
RISELow, 2-645
MEASUrement:MATH<x>:REFLevels:PERCent:
RISEMid, 2-645
MEASUrement:MATH<x>:REFLevels:PERCent:
TYPE, 2-646
MEASUrement:MEAS<x>:BER, 2-648

MEASUrement:MEAS<x>:BER:
TARGETBER, 2-649
MEASUrement:MEAS<x>:BIN, 2-649
MEASUrement:MEAS<x>:BITCfgmode, 2-650
MEASUrement:MEAS<x>:BITEnd, 2-650
MEASUrement:MEAS<x>:BITPcnt, 2-651
MEASUrement:MEAS<x>:BITSTart, 2-651
MEASUrement:MEAS<x>:BITType, 2-652
MEASUrement:MEAS<x>:BURSTEDGTYPe, 2-652
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
MAXimum?, 2-653
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
MEAN?, 2-653
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
MINimum?, 2-654
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
PK2PK?, 2-654
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
POPUlation?, 2-654
MEASUrement:MEAS<x>:CCRESUlts:ALLAcqs:
STDDev?, 2-655
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:MAXimum?, 2-655
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:MEAN?, 2-656
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:MINimum?, 2-656
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:PK2PK?, 2-656
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:POPUlation?, 2-657
MEASUrement:MEAS<x>:CCRESUlts:
CURRentacq:STDDev?, 2-657
MEASUrement:MEAS<x>:CLOCKRecovery:
ADVanced:METHod, 2-658
MEASUrement:MEAS<x>:CLOCKRecovery:
CLOCKFrequency, 2-658
MEASUrement:MEAS<x>:CLOCKRecovery:
CLOCKMultiplier, 2-659
MEASUrement:MEAS<x>:CLOCKRecovery:
CONSTCLOCKMODe, 2-659
MEASUrement:MEAS<x>:CLOCKRecovery:
DAMPing, 2-660
MEASUrement:MEAS<x>:CLOCKRecovery:
DATAPath, 2-660
MEASUrement:MEAS<x>:CLOCKRecovery:
DATARate, 2-661
MEASUrement:MEAS<x>:CLOCKRecovery:
EXPLICITCLOCKMODe, 2-661
MEASUrement:MEAS<x>:CLOCKRecovery:
GLOBal, 2-662
MEASUrement:MEAS<x>:CLOCKRecovery:
JTFBandwidth, 2-663
MEASUrement:MEAS<x>:CLOCKRecovery:
LOOPBandwidth, 2-663
MEASUrement:MEAS<x>:CLOCKRecovery:
MEANAUTOCalculate, 2-664
MEASUrement:MEAS<x>:CLOCKRecovery:
METHod, 2-664
MEASUrement:MEAS<x>:CLOCKRecovery:
MODel, 2-665
MEASUrement:MEAS<x>:CLOCKRecovery:
NOMINALOFFset, 2-665
MEASUrement:MEAS<x>:CLOCKRecovery:
NOMINALOFFset:SELECTIONtype, 2-666
MEASUrement:MEAS<x>:CLOCKRecovery:
STAndard, 2-666
MEASUrement:MEAS<x>:COMMONMode:
FILTers:STATE, 2-667
MEASUrement:MEAS<x>:COMMONMode:
SOURCEs, 2-668
MEASUrement:MEAS<x>:CONFIGuration, 2-668
MEASUrement:MEAS<x>:
CUSTOMLIMITSFile, 2-669
MEASUrement:MEAS<x>:CYCLemode, 2-669
MEASUrement:MEAS<x>:DELay:EDGE<x>, 2-670
MEASUrement:MEAS<x>:DISPlaystat:
ENABle, 2-671
MEASUrement:MEAS<x>:EDGE<x>, 2-677
MEASUrement:MEAS<x>:EDGEIncre, 2-671
MEASUrement:MEAS<x>:EDGEQUALifier, 2-672
MEASUrement:MEAS<x>:EDGEQUALONE, 2-672
MEASUrement:MEAS<x>:EDGEQUALTWO, 2-673
MEASUrement:MEAS<x>:EDGES:
FROMLevel, 2-673
MEASUrement:MEAS<x>:EDGES:LEVel, 2-674
MEASUrement:MEAS<x>:EDGES:
LOWERFREQuency, 2-674
MEASUrement:MEAS<x>:EDGES:N, 2-675
MEASUrement:MEAS<x>:EDGES:
SLEWRATEMethod, 2-675
MEASUrement:MEAS<x>:EDGES:TOLevel, 2-676
MEASUrement:MEAS<x>:EDGES:
UPPERFREQuency, 2-677

MEASUrement:MEAS<x>:FAILCount?, 2-678
MEASUrement:MEAS<x>:
FILTERRANGEFROM, 2-678
MEASUrement:MEAS<x>:
FILTERRANGETO, 2-679
MEASUrement:MEAS<x>:FILTers:
BLANKingtime, 2-679
MEASUrement:MEAS<x>:FILTers:GLOBal, 2-680
MEASUrement:MEAS<x>:FILTers:HIGHPass:
FREQ, 2-680
MEASUrement:MEAS<x>:FILTers:HIGHPass:
SPEC, 2-681
MEASUrement:MEAS<x>:FILTers:LOWPass:
FREQ, 2-681
MEASUrement:MEAS<x>:FILTers:LOWPass:
SPEC, 2-682
MEASUrement:MEAS<x>:FILTers:
RAMPtime, 2-683
MEASUrement:MEAS<x>:FREQ, 2-683
MEASUrement:MEAS<x>:FROMedge, 2-684
MEASUrement:MEAS<x>:
FROMEDGESEARCHDIRect, 2-684
MEASUrement:MEAS<x>:FROMSymbol:
LOGIC2SOUrce, 2-685
MEASUrement:MEAS<x>:FROMSymbol:
LOGIC3SOUrce, 2-685
MEASUrement:MEAS<x>:FROMSymbol:
LOGIC4SOUrce, 2-686
MEASUrement:MEAS<x>:FROMSymbol:
LOGIC5SOUrce, 2-687
MEASUrement:MEAS<x>:FROMSymbol:
MEASUREAT, 2-688
MEASUrement:MEAS<x>:FUNDCURRent, 2-689
MEASUrement:MEAS<x>:GATing, 2-689
MEASUrement:MEAS<x>:GATing:ACTive, 2-690
MEASUrement:MEAS<x>:GATing:ENDtime, 2-690
MEASUrement:MEAS<x>:GATing:GLOBal, 2-691
MEASUrement:MEAS<x>:GATing:
HYSTeresis, 2-692
MEASUrement:MEAS<x>:GATing:
LOGICSource, 2-692
MEASUrement:MEAS<x>:GATing:MIDRef, 2-693
MEASUrement:MEAS<x>:GATing:
SEARCHSource, 2-693
MEASUrement:MEAS<x>:GATing:
STARTtime, 2-694
MEASUrement:MEAS<x>:GLOBalref, 2-694
MEASUrement:MEAS<x>:
HARMONICSCLass, 2-695
MEASUrement:MEAS<x>:
HARMONICSSOURce, 2-695
MEASUrement:MEAS<x>:HARMONICSStd, 2-696
MEASUrement:MEAS<x>:
HARMONICSUNits, 2-696
MEASUrement:MEAS<x>:HIGHREFVoltage, 2-697
MEASUrement:MEAS<x>:IDLETime, 2-697
MEASUrement:MEAS<x>:INFILTers:LOWPass:
FREQ, 2-698
MEASUrement:MEAS<x>:INFILTers:LOWPass:
SPEC, 2-698
MEASUrement:MEAS<x>:INPUTLEVel, 2-699
MEASUrement:MEAS<x>:INPUTPOwer, 2-700
MEASUrement:MEAS<x>:JITTERSummary:
DCD, 2-700
MEASUrement:MEAS<x>:JITTERSummary:
DDJ, 2-701
MEASUrement:MEAS<x>:JITTERSummary:
DJDD, 2-701
MEASUrement:MEAS<x>:JITTERSummary:
EYEWIDTHBER, 2-702
MEASUrement:MEAS<x>:JITTERSummary:
NPJ, 2-702
MEASUrement:MEAS<x>:JITTERSummary:
PJ, 2-703
MEASUrement:MEAS<x>:JITTERSummary:
RJDD, 2-703
MEASUrement:MEAS<x>:JITTERSummary:
TIE, 2-704
MEASUrement:MEAS<x>:JITTERSummary:
TJBER, 2-704
MEASUrement:MEAS<x>:LABel, 2-705
MEASUrement:MEAS<x>:LINESelected, 2-705
MEASUrement:MEAS<x>:LOWREFVoltage, 2-706
MEASurement:MEAS<x>:MAXCycle, 2-706
MEASUrement:MEAS<x>:MAXVoltage, 2-707
MEASUrement:MEAS<x>:MEASRange:
GLOBal, 2-707
MEASUrement:MEAS<x>:MEASRange:
MAX, 2-708
MEASUrement:MEAS<x>:MEASRange:MIN, 2-709
MEASUrement:MEAS<x>:MEASRange:
STATE, 2-709
MEASurement:MEAS<x>:MINCycle, 2-710
MEASUrement:MEAS<x>:ODDEVen, 2-710

MEASUrement:MEAS<x>:OUTFILTers:LOWPass:
FREQ, 2-711
MEASUrement:MEAS<x>:OUTFILTers:LOWPass:
SPEC, 2-711
MEASUrement:MEAS<x>:
OUTPUT<n>VOLTage, 2-712
MEASUrement:MEAS<x>:
PASSFAILENabled, 2-712
MEASUrement:MEAS<x>:
PASSFAILHIGHlimit, 2-713
MEASUrement:MEAS<x>:PASSFAILLIMit, 2-714
MEASUrement:MEAS<x>:
PASSFAILLOWlimit, 2-715
MEASUrement:MEAS<x>:
PASSFAILMARgin, 2-715
MEASUrement:MEAS<x>:PASSFAILWHEN, 2-716
MEASUrement:MEAS<x>:
PATTERNDETECTION, 2-717
MEASUrement:MEAS<x>:PATTERNLENgth, 2-717
MEASUrement:MEAS<x>:PATTERNTYPe, 2-718
MEASUrement:MEAS<x>:PERFREQ:EDGE, 2-718
MEASUrement:MEAS<x>:POLarity, 2-719
MEASUrement:MEAS<x>:POPUlation:
GLOBal, 2-720
MEASUrement:MEAS<x>:POPUlation:LIMIT:
STATE, 2-720
MEASUrement:MEAS<x>:POPUlation:LIMIT:
VALue, 2-721
MEASUrement:MEAS<x>:POWERFACtor, 2-721
MEASUrement:MEAS<x>:REFLevels:ABSolute:
FALLHigh, 2-722
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:FALLLow, 2-722
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:FALLMid, 2-723
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:HYSTeresis, 2-724
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:RISEHigh, 2-724
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:RISELow, 2-725
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:RISEMid, 2-725
MEASUrement:MEAS<x>:REFLevels<x>:
ABSolute:TYPE, 2-726
MEASUrement:MEAS<x>:REFLevels<x>:
BASETop, 2-726
MEASUrement:MEAS<x>:REFLevels<x>:
METHod, 2-727
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
FALLHigh, 2-728
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
FALLLow, 2-728
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
FALLMid, 2-729
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
HYSTeresis, 2-729
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
RISEHigh, 2-730
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
RISELow, 2-731
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
RISEMid, 2-731
MEASUrement:MEAS<x>:REFLevels<x>:PERCent:
TYPE, 2-732
MEASUrement:MEAS<x>:REFMode, 2-732
MEASUrement:MEAS<x>:REFVoltage, 2-733
MEASUrement:MEAS<x>:
REFVOLTAGE<n>Val, 2-733
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
MAXimum?, 2-734
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
MEAN?, 2-734
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
MINimum?, 2-735
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
PK2PK?, 2-735
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
POPUlation?, 2-736
MEASUrement:MEAS<x>:RESUlts:ALLAcqs:
STDDev?, 2-736
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
MAXimum?, 2-737
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
MEAN?, 2-737
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
MINimum?, 2-738
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
PK2PK?, 2-738
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
POPUlation?, 2-738
MEASUrement:MEAS<x>:RESUlts:CURRentacq:
STDDev?, 2-739

MEASUrement:MEAS<x>:
RIPPLEFREQ<n>Val, 2-739
MEASUrement:MEAS<x>:
SIGNALFREQUEncy, 2-740
MEASUrement:MEAS<x>:SIGNALType, 2-741
MEASUrement:MEAS<x>:SOUrce<x>, 2-741
MEASUrement:MEAS<x>:SSC:NOMinalfreq, 2-742
MEASUrement:MEAS<x>:SSC:NOMinalfreq:
SELECTIONtype, 2-742
MEASUrement:MEAS<x>:STATUS?, 2-743
MEASUrement:MEAS<x>:TCKAVG, 2-743
MEASurement:MEAS<x>:TIMINGMode, 2-744
MEASUrement:MEAS<x>:TOEdge, 2-744
MEASUrement:MEAS<x>:
TOEDGESEARCHDIRect, 2-745
MEASUrement:MEAS<x>:TOSYmbol:
LOGIC2SOUrce, 2-746
MEASUrement:MEAS<x>:TOSYmbol:
LOGIC3SOUrce, 2-746
MEASUrement:MEAS<x>:TOSYmbol:
LOGIC4SOUrce, 2-747
MEASUrement:MEAS<x>:TOSYmbol:
LOGIC5SOUrce, 2-748
MEASUrement:MEAS<x>:TOSYmbol:
MEASUREAT, 2-749
MEASUrement:MEAS<x>:TRANSition, 2-750
MEASUrement:MEAS<x>:TYPe, 2-750
MEASUrement:MEAS<x>:WAITTime, 2-760
MEASUrement:MEAS<x>:WINDOWLENgth, 2-760
MEASUrement:MEAS<x>:XUNIT?, 2-761
MEASUrement:MEAS<x>:YUNIT?, 2-761
MEASUrement:MEASRange:MAX, 2-647
MEASUrement:MEASRange:MIN, 2-647
MEASUrement:MEASRange:STATE, 2-648
MEASUrement:MINUI, 2-762
MEASUrement:POPUlation:LIMIT:STATE, 2-762
MEASUrement:POPUlation:LIMIT:VALue, 2-763
MEASUrement:REF<x>:REFLevels:ABSolute:
FALLHigh, 2-774
MEASUrement:REF<x>:REFLevels:ABSolute:
FALLLow, 2-774
MEASUrement:REF<x>:REFLevels:ABSolute:
FALLMid, 2-775
MEASUrement:REF<x>:REFLevels:ABSolute:
HYSTeresis, 2-775
MEASUrement:REF<x>:REFLevels:ABSolute:
RISEHigh, 2-776
MEASUrement:REF<x>:REFLevels:ABSolute:
RISELow, 2-776
MEASUrement:REF<x>:REFLevels:ABSolute:
RISEMid, 2-777
MEASUrement:REF<x>:REFLevels:ABSolute:
TYPE, 2-777
MEASUrement:REF<x>:REFLevels:
BASETop, 2-778
MEASUrement:REF<x>:REFLevels:METHod, 2-778
MEASUrement:REF<x>:REFLevels:PERCent:
FALLHigh, 2-779
MEASUrement:REF<x>:REFLevels:PERCent:
FALLLow, 2-779
MEASUrement:REF<x>:REFLevels:PERCent:
FALLMid, 2-780
MEASUrement:REF<x>:REFLevels:PERCent:
HYSTeresis, 2-780
MEASUrement:REF<x>:REFLevels:PERCent:
RISEHigh, 2-781
MEASUrement:REF<x>:REFLevels:PERCent:
RISELow, 2-781
MEASUrement:REF<x>:REFLevels:PERCent:
RISEMid, 2-782
MEASUrement:REF<x>:REFLevels:PERCent:
TYPE, 2-782
MEASUrement:REFLevels:ABSolute:
FALLHigh, 2-763
MEASUrement:REFLevels:ABSolute:
FALLLow, 2-764
MEASUrement:REFLevels:ABSolute:
FALLMid, 2-764
MEASUrement:REFLevels:ABSolute:
HYSTeresis, 2-765
MEASUrement:REFLevels:ABSolute:
RISEHigh, 2-765
MEASUrement:REFLevels:ABSolute:
RISELow, 2-766
MEASUrement:REFLevels:ABSolute:
RISEMid, 2-766
MEASUrement:REFLevels:ABSolute:TYPE, 2-766
MEASUrement:REFLevels:BASETop, 2-767
MEASUrement:REFLevels:JITTERMODE, 2-767
MEASUrement:REFLevels:METHod, 2-768
MEASUrement:REFLevels:MODE, 2-769
MEASUrement:REFLevels:PERCent:
FALLHigh, 2-769

MEASUrement:REFLevels:PERCent:
FALLLow, 2-770
MEASUrement:REFLevels:PERCent:
FALLMid, 2-770
MEASUrement:REFLevels:PERCent:
HYSTeresis, 2-771
MEASUrement:REFLevels:PERCent:
RISEHigh, 2-771
MEASUrement:REFLevels:PERCent:
RISELow, 2-772
MEASUrement:REFLevels:PERCent:
RISEMid, 2-772
MEASUrement:REFLevels:PERCent:TYPE, 2-773
MEASUrement:REFLevels:TYPE, 2-773
MEASUrement:STATIstics:CYCLEMode, 2-783

N

NEWpass, 2-784

O

*OPC, 2-784
*OPT?, 2-785

P

PASSWord, 2-785
PAUSe, 2-786
PLOT:ADDNew, 2-786
PLOT:DELete, 2-787
PLOT:LIST?, 2-787
PLOT:PLOT<x>:BATHtub:BER, 2-788
PLOT:PLOT<x>:BATHtub:XAXISUnits, 2-788
PLOT:PLOT<x>:BITType, 2-788
PLOT:PLOT<x>:EXPORTRaw?, 2-789
PLOT:PLOT<x>:EXTENDuis, 2-790
PLOT:PLOT<x>:EYERender, 2-790
PLOT:PLOT<x>:LABel:COLor, 2-791
PLOT:PLOT<x>:LABel:FONT:BOLD, 2-791
PLOT:PLOT<x>:LABel:FONT:ITALic, 2-791
PLOT:PLOT<x>:LABel:FONT:SIZE, 2-792
PLOT:PLOT<x>:LABel:FONT:TYPE, 2-792
PLOT:PLOT<x>:LABel:FONT:UNDERline, 2-793
PLOT:PLOT<x>:LABel:NAMe, 2-793
PLOT:PLOT<x>:LABel:XPOS, 2-794
PLOT:PLOT<x>:LABel:YPOS, 2-794
PLOT:PLOT<x>:MASK?, 2-795
PLOT:PLOT<x>:NUMBins, 2-795
PLOT:PLOT<x>:RAILNUM, 2-795
PLOT:PLOT<x>:SOUrce<x>, 2-796
PLOT:PLOT<x>:SPECtrum:BASE, 2-796
PLOT:PLOT<x>:SPECtrum:DYNRange, 2-797
PLOT:PLOT<x>:TYPe, 2-797
POWer:ADDNew, 2-798
POWer:DELete, 2-799
POWer:POWer<x>:AUTOSet, 2-799
POWer:POWer<x>:CLRESPONSE:
AMP[1-10]Val, 2-800
POWer:POWer<x>:CLRESPONSE:
AMPMode, 2-800
POWer:POWer<x>:CLRESPONSE:
CONNECTSTATus?, 2-801
POWer:POWer<x>:CLRESPONSE:
CONSTAMPlitude, 2-802
POWer:POWer<x>:CLRESPONSE:
FREQ[1-11]Val, 2-802
POWer:POWer<x>:CLRESPONSE:GENerator, 2-804
POWer:POWer<x>:CLRESPONSE:
GENIPADDress, 2-804
POWer:POWer<x>:CLRESPONSE:
IMPEDance, 2-805
POWer:POWer<x>:CLRESPONSE:
INPUTSOurce, 2-805
POWer:POWer<x>:CLRESPONSE:
OUTPUTSOurce, 2-806
POWer:POWer<x>:CLRESPONSE:PPD, 2-806
POWer:POWer<x>:CLRESPONSE:
STARTFREQuency, 2-807
POWer:POWer<x>:CLRESPONSE:
STOPFREQuency, 2-808
POWer:POWer<x>:CLRESPONSE:
TESTCONNection, 2-808
POWer:POWer<x>:CYCLEAmp:
INPUTSOurce, 2-809
POWer:POWer<x>:CYCLEBase:
INPUTSOurce, 2-809
POWer:POWer<x>:CYCLEMAX:
INPUTSOurce, 2-810
POWer:POWer<x>:CYCLEMin:
INPUTSOurce, 2-811
POWer:POWer<x>:CYCLEPKPK:
INPUTSOurce, 2-811
POWer:POWer<x>:CYCLETop:
INPUTSOurce, 2-812
POWer:POWer<x>:DIDT:INPUTSOurce, 2-812

POWer:POWer<x>:DIDT:
SOURCEEDGEType, 2-813
POWer:POWer<x>:DVDT:INPUTSOurce, 2-813
POWer:POWer<x>:DVDT:
SOURCEEDGEType, 2-814
POWer:POWer<x>:EFFICIENCY:INPUTType, 2-814
POWer:POWer<x>:EFFICIENCY:
IOUT1SOUrce, 2-815
POWer:POWer<x>:EFFICIENCY:
IOUT2SOUrce, 2-816
POWer:POWer<x>:EFFICIENCY:
IOUT3SOUrce, 2-817
POWer:POWer<x>:EFFICIENCY:ISOUrce, 2-817
POWer:POWer<x>:EFFICIENCY:
NUMOFOutputs, 2-818
POWer:POWer<x>:EFFICIENCY:
OUTPUT1Type, 2-819
POWer:POWer<x>:EFFICIENCY:
OUTPUT2Type, 2-819
POWer:POWer<x>:EFFICIENCY:
OUTPUT3Type, 2-820
POWer:POWer<x>:EFFICIENCY:
OUTPUTType, 2-821
POWer:POWer<x>:EFFICIENCY:
VOUT1SOUrce, 2-821
POWer:POWer<x>:EFFICIENCY:
VOUT2SOUrce, 2-822
POWer:POWer<x>:EFFICIENCY:
VOUT3SOUrce, 2-823
POWer:POWer<x>:EFFICIENCY:VSOUrce, 2-824
POWer:POWer<x>:FREQUENCY:EDGe, 2-824
POWer:POWer<x>:FREQUENCY:
INPUTSOurce, 2-825
POWer:POWer<x>:GATing, 2-826
POWer:POWer<x>:GATing:GLOBal, 2-826
POWer:POWer<x>:HARMONICS:CLASs, 2-827
POWer:POWer<x>:HARMONICS:CMEThod, 2-827
POWer:POWer<x>:HARMONICS:
FUNDCURRent, 2-828
POWer:POWer<x>:HARMONICS:HORDer, 2-829
POWer:POWer<x>:HARMONICS:HSOURce, 2-829
POWer:POWer<x>:HARMONICS:IPOWer, 2-830
POWer:POWer<x>:HARMONICS:ISOURce, 2-830
POWer:POWer<x>:HARMONICS:
LINEFREQUEncy, 2-831
POWer:POWer<x>:HARMONICS:ODDEVen, 2-832
POWer:POWer<x>:HARMONICS:PFACtor, 2-832
POWer:POWer<x>:HARMONICS:
POWERRating, 2-833
POWer:POWer<x>:HARMONICS:RCURRent, 2-833
POWer:POWer<x>:HARMONICS:STANDard, 2-834
POWer:POWer<x>:HARMONICS:
STARTFREQUEncy, 2-835
POWer:POWer<x>:HARMONICS:UNITs, 2-835
POWer:POWer<x>:HARMONICS:VSOURce, 2-836
POWer:POWer<x>:IMPEDANCE:
AMP[1-10]Val, 2-836
POWer:POWer<x>:IMPEDANCE:AMPMode, 2-837
POWer:POWer<x>:IMPEDANCE:
CONNECTSTATus?, 2-837
POWer:POWer<x>:IMPEDANCE:
CONSTAMPlitude, 2-838
POWer:POWer<x>:IMPEDANCE:
FREQ[1-10]Val, 2-839
POWer:POWer<x>:IMPEDANCE:GENerator, 2-839
POWer:POWer<x>:IMPEDANCE:
GENIPADDress, 2-840
POWer:POWer<x>:IMPEDANCE:
IMPEDANCE, 2-841
POWer:POWer<x>:IMPEDANCE:
INPUTSOurce, 2-841
POWer:POWer<x>:IMPEDANCE:
OUTPUTSOurce, 2-842
POWer:POWer<x>:IMPEDANCE:PPD, 2-842
POWer:POWer<x>:IMPEDANCE:
STARTFREQuency, 2-843
POWer:POWer<x>:IMPEDANCE:
STOPFREQuency, 2-844
POWer:POWer<x>:IMPEDANCE:
TESTCONNection, 2-844
POWer:POWer<x>:INDUCTANCE:
EDGESource, 2-845
POWer:POWer<x>:INDUCTANCE:ISOUrce, 2-845
POWer:POWer<x>:INDUCTANCE:VSOURce, 2-846
POWer:POWer<x>:INPUTCAP:ISOURce, 2-847
POWer:POWer<x>:INPUTCAP:
PEAKCURRent, 2-848
POWer:POWer<x>:INPUTCAP:
PEAKVOLTage, 2-848
POWer:POWer<x>:INPUTCAP:VSOURce, 2-849
POWer:POWer<x>:INRUSHcurrent:
INPUTSOurce, 2-850
POWer:POWer<x>:INRUSHcurrent:
PEAKCURRent, 2-850

POWer:POWer<x>:IVSINTEGRALV:
ISOURce, 2-851
POWer:POWer<x>:IVSINTEGRALV:
VSOURce, 2-852
POWer:POWer<x>:LABel, 2-852
POWer:POWer<x>:LINERIPPLE:
INPUTSOurce, 2-853
POWer:POWer<x>:LINERIPPLE:
LFREQuency, 2-853
POWer:POWer<x>:MAGNETICLOSS:
ISOURce, 2-854
POWer:POWer<x>:MAGNETICLOSS:
VSOURce, 2-854
POWer:POWer<x>:MAGPROPERTY:
AREAofcrosssection, 2-855
POWer:POWer<x>:MAGPROPERTY:
EDGESOURce, 2-856
POWer:POWer<x>:MAGPROPERTY:
ISOURce, 2-856
POWer:POWer<x>:MAGPROPERTY:LENgth, 2-857
POWer:POWer<x>:MAGPROPERTY:
PRIMARYTURNs, 2-858
POWer:POWer<x>:MAGPROPERTY:
SEC1SOURce, 2-859
POWer:POWer<x>:MAGPROPERTY:
SEC1TURNs, 2-859
POWer:POWer<x>:MAGPROPERTY:
SEC2SOURce, 2-860
POWer:POWer<x>:MAGPROPERTY:
SEC2TURNs, 2-861
POWer:POWer<x>:MAGPROPERTY:
SEC3SOURce, 2-862
POWer:POWer<x>:MAGPROPERTY:
SEC3TURNs, 2-863
POWer:POWer<x>:MAGPROPERTY:
SEC4SOURce, 2-864
POWer:POWer<x>:MAGPROPERTY:
SEC4TURNs, 2-864
POWer:POWer<x>:MAGPROPERTY:
SEC5SOURce, 2-865
POWer:POWer<x>:MAGPROPERTY:
SEC5TURNs, 2-866
POWer:POWer<x>:MAGPROPERTY:
SEC6SOURce, 2-867
POWer:POWer<x>:MAGPROPERTY:
SEC6TURNs, 2-868
POWer:POWer<x>:MAGPROPERTY:
SECWINDings, 2-869
POWer:POWer<x>:MAGPROPERTY:UNITs, 2-870
POWer:POWer<x>:MAGPROPERTY:
VSOURce, 2-870
POWer:POWer<x>:NDUTYCYCLE:
EDGEType, 2-871
POWer:POWer<x>:NDUTYCYCLE:
INPUTSOurce, 2-871
POWer:POWer<x>:NPULSEWIDTH:
INPUTSOurce, 2-872
POWer:POWer<x>:PDUTYCYCLE:
EDGEType, 2-873
POWer:POWer<x>:PDUTYCYCLE:
INPUTSOurce, 2-873
POWer:POWer<x>:PERIOD:EDGe, 2-874
POWer:POWer<x>:PERIOD:INPUTSOurce, 2-874
POWer:POWer<x>:POWERQUALITY:
CCYCles, 2-875
POWer:POWer<x>:POWERQUALITY:
FREFerence, 2-875
POWer:POWer<x>:POWERQUALITY:
ISOURce, 2-876
POWer:POWer<x>:POWERQUALITY:
VSOURce, 2-876
POWer:POWer<x>:PPULSEWIDTH:
INPUTSOurce, 2-877
POWer:POWer<x>:PRESET, 2-877
POWer:POWer<x>:PSRR:AMP[1-10]Val, 2-878
POWer:POWer<x>:PSRR:AMPMode, 2-879
POWer:POWer<x>:PSRR:
CONNECTSTATus?, 2-879
POWer:POWer<x>:PSRR:CONSTAMPlitude, 2-880
POWer:POWer<x>:PSRR:FREQ[1-11]Val, 2-880
POWer:POWer<x>:PSRR:GENerator, 2-882
POWer:POWer<x>:PSRR:GENIPADDress, 2-882
POWer:POWer<x>:PSRR:IMPEDance, 2-883
POWer:POWer<x>:PSRR:INPUTSOurce, 2-884
POWer:POWer<x>:PSRR:OUTPUTSOurce, 2-884
POWer:POWer<x>:PSRR:PPD, 2-885
POWer:POWer<x>:PSRR:STARTFREQuency, 2-885
POWer:POWer<x>:PSRR:STOPFREQuency, 2-886
POWer:POWer<x>:PSRR:TESTCONNection, 2-886
POWer:POWer<x>:RDSON:DEVICEType, 2-887
POWer:POWer<x>:RDSON:ISOURce, 2-888
POWer:POWer<x>:RDSON:VSOURce, 2-888

POWer:POWer<x>:REFLevels:ABSolute:FALLHigh, 2-889
POWer:POWer<x>:REFLevels:ABSolute:FALLLow, 2-890
POWer:POWer<x>:REFLevels:ABSolute:FALLMid, 2-890
POWer:POWer<x>:REFLevels:ABSolute:HYSTeresis, 2-891
POWer:POWer<x>:REFLevels:ABSolute:RISEHigh, 2-891
POWer:POWer<x>:REFLevels:ABSolute:RISELow, 2-892
POWer:POWer<x>:REFLevels:ABSolute:RISEMid, 2-893
POWer:POWer<x>:REFLevels:ABSolute:TYPE, 2-893
POWer:POWer<x>:REFLevels:BASETop, 2-894
POWer:POWer<x>:REFLevels:METHod, 2-894
POWer:POWer<x>:REFLevels:PERCent:FALLHigh, 2-895
POWer:POWer<x>:REFLevels:PERCent:FALLLow, 2-895
POWer:POWer<x>:REFLevels:PERCent:FALLMid, 2-896
POWer:POWer<x>:REFLevels:PERCent:HYSTeresis, 2-897
POWer:POWer<x>:REFLevels:PERCent:RISEHigh, 2-897
POWer:POWer<x>:REFLevels:PERCent:RISELow, 2-898
POWer:POWer<x>:REFLevels:PERCent:RISEMid, 2-898
POWer:POWer<x>:REFLevels:PERCent:TYPE, 2-899
POWer:POWer<x>:RESUlts:ALLAcqs:MAXimum?, 2-900
POWer:POWer<x>:RESUlts:ALLAcqs:MEAN?, 2-902
POWer:POWer<x>:RESUlts:ALLAcqs:MINimum?, 2-904
POWer:POWer<x>:RESUlts:ALLAcqs:PK2PK?, 2-906
POWer:POWer<x>:RESUlts:ALLAcqs:POPUlation?, 2-909
POWer:POWer<x>:RESUlts:ALLAcqs:STDDev?, 2-911
POWer:POWer<x>:RESUlts:CURRentacq:F1MAG?, 2-913
POWer:POWer<x>:RESUlts:CURRentacq:F3MAG?, 2-913
POWer:POWer<x>:RESUlts:CURRentacq:FREQUENCY?, 2-914
POWer:POWer<x>:RESUlts:CURRentacq:IRMS?, 2-914
POWer:POWer<x>:RESUlts:CURRentacq:MAXimum?, 2-915
POWer:POWer<x>:RESUlts:CURRentacq:MEAN?, 2-917
POWer:POWer<x>:RESUlts:CURRentacq:MINimum?, 2-919
POWer:POWer<x>:RESUlts:CURRentacq:PK2PK?, 2-921
POWer:POWer<x>:RESUlts:CURRentacq:POHCL?, 2-923
POWer:POWer<x>:RESUlts:CURRentacq:POHCM?, 2-924
POWer:POWer<x>:RESUlts:CURRentacq:POHCS?, 2-924
POWer:POWer<x>:RESUlts:CURRentacq:POPUlation?, 2-924
POWer:POWer<x>:RESUlts:CURRentacq:RMS?, 2-927
POWer:POWer<x>:RESUlts:CURRentacq:STATUS?, 2-927
POWer:POWer<x>:RESUlts:CURRentacq:STDDev?, 2-928
POWer:POWer<x>:RESUlts:CURRentacq:THDF?, 2-930
POWer:POWer<x>:RESUlts:CURRentacq:THDR?, 2-930
POWer:POWer<x>:RESUlts:CURRentacq:TRPWR?, 2-931
POWer:POWer<x>:RESUlts:CURRentacq:VRMS?, 2-931
POWer:POWer<x>:SEQSETup, 2-932
POWer:POWer<x>:SEQuence, 2-932
POWer:POWer<x>:SOA:ISOURce, 2-933
POWer:POWer<x>:SOA:POINT, 2-933
POWer:POWer<x>:SOA:RECAllmask, 2-934
POWer:POWer<x>:SOA:RECAllmask:FILEName, 2-934
POWer:POWer<x>:SOA:SAVemask, 2-935

POWer:POWer<x>:SOA:SAVemask:
AUTOINCrement, 2-935
POWer:POWer<x>:SOA:SAVemask:
FILEName, 2-936
POWer:POWer<x>:SOA:SAVemask:FOLDer, 2-936
POWer:POWer<x>:SOA:VSOURce, 2-937
POWer:POWer<x>:SWITCHINGLOSS:
DEVICEType, 2-937
POWer:POWer<x>:SWITCHINGLOSS:
GATESOurce, 2-938
POWer:POWer<x>:SWITCHINGLOSS:
ILEVELAbs, 2-938
POWer:POWer<x>:SWITCHINGLOSS:
ILEVELPct, 2-939
POWer:POWer<x>:SWITCHINGLOSS:
ISOURce, 2-940
POWer:POWer<x>:SWITCHINGLOSS:
LEVELUNIts, 2-940
POWer:POWer<x>:SWITCHINGLOSS:
RDSOn, 2-941
POWer:POWer<x>:SWITCHINGLOSS:
SWLCONFIGType, 2-941
POWer:POWer<x>:SWITCHINGLOSS:
VCESat, 2-942
POWer:POWer<x>:SWITCHINGLOSS:
VGLevel, 2-943
POWer:POWer<x>:SWITCHINGLOSS:
VLEVELAbs, 2-943
POWer:POWer<x>:SWITCHINGLOSS:
VLEVELPct, 2-944
POWer:POWer<x>:SWITCHINGLOSS:
VSOURce, 2-944
POWer:POWer<x>:SWITCHINGRIPPLE:
INPUTSOurce, 2-945
POWer:POWer<x>:SWITCHINGRIPPLE:
LFREQuency, 2-946
POWer:POWer<x>:TURNOFFtime:
FREQuency, 2-946
POWer:POWer<x>:TURNOFFtime:
INPUTLEVel, 2-947
POWer:POWer<x>:TURNOFFtime:
INPUTSOurce, 2-947
POWer:POWer<x>:TURNOFFtime:
MAXTIMe, 2-948
POWer:POWer<x>:TURNOFFtime:
MAXVoltage, 2-949
POWer:POWer<x>:TURNOFFtime:
NUMOUTputs, 2-949
POWer:POWer<x>:TURNOFFtime:
OUTPUT1SOURce, 2-951
POWer:POWer<x>:TURNOFFtime:
OUTPUT1VOLTage, 2-951
POWer:POWer<x>:TURNOFFtime:
OUTPUT2SOURce, 2-952
POWer:POWer<x>:TURNOFFtime:
OUTPUT2VOLTage, 2-953
POWer:POWer<x>:TURNOFFtime:
OUTPUT3SOURce, 2-953
POWer:POWer<x>:TURNOFFtime:
OUTPUT3VOLTage, 2-954
POWer:POWer<x>:TURNOFFtime:
OUTPUT4SOURce, 2-955
POWer:POWer<x>:TURNOFFtime:
OUTPUT4VOLTage, 2-955
POWer:POWer<x>:TURNOFFtime:
OUTPUT5SOURce, 2-956
POWer:POWer<x>:TURNOFFtime:
OUTPUT5VOLTage, 2-957
POWer:POWer<x>:TURNOFFtime:
OUTPUT6SOURce, 2-958
POWer:POWer<x>:TURNOFFtime:
OUTPUT6VOLTage, 2-958
POWer:POWer<x>:TURNOFFtime:
OUTPUT7SOURce, 2-959
POWer:POWer<x>:TURNOFFtime:
OUTPUT7VOLTage, 2-960
POWer:POWer<x>:TURNOFFtime:TYPE, 2-960
POWer:POWer<x>:TURNONtime:
FREQuency, 2-961
POWer:POWer<x>:TURNONtime:
INPUTLEVel, 2-962
POWer:POWer<x>:TURNONtime:
INPUTSOurce, 2-962
POWer:POWer<x>:TURNONtime:MAXTIMe, 2-963
POWer:POWer<x>:TURNONTIME:
MAXVoltage, 2-963
POWer:POWer<x>:TURNONtime:
NUMOUTputs, 2-964
POWer:POWer<x>:TURNONtime:
OUTPUT1SOURce, 2-965
POWer:POWer<x>:TURNONtime:
OUTPUT1VOLTage, 2-966

POWer:POWer<x>:TURNONtime:
OUTPUT2SOURce, 2-967
POWer:POWer<x>:TURNONtime:
OUTPUT2VOLTage, 2-967
POWer:POWer<x>:TURNONtime:
OUTPUT3SOURce, 2-968
POWer:POWer<x>:TURNONtime:
OUTPUT3VOLTage, 2-969
POWer:POWer<x>:TURNONtime:
OUTPUT4SOURce, 2-969
POWer:POWer<x>:TURNONtime:
OUTPUT4VOLTage, 2-970
POWer:POWer<x>:TURNONtime:
OUTPUT5SOURce, 2-971
POWer:POWer<x>:TURNONtime:
OUTPUT5VOLTage, 2-971
POWer:POWer<x>:TURNONtime:
OUTPUT6SOURce, 2-972
POWer:POWer<x>:TURNONtime:
OUTPUT6VOLTage, 2-973
POWer:POWer<x>:TURNONtime:
OUTPUT7SOURce, 2-974
POWer:POWer<x>:TURNONtime:
OUTPUT7VOLTage, 2-974
POWer:POWer<x>:TURNONtime:TYPE, 2-975
POWer:POWer<x>:TYPe, 2-976
*PSC, 2-976
*PUD, 2-977

R

RECAll:MASK, 2-978
RECAll:SESsion, 2-978
RECAll:SETUp, 2-979
RECAll:WAVEform, 2-980
REF:ADDNew, 2-981
REF:DELete, 2-981
REF:LIST?, 2-981
REF:REF<x>:DESKew, 2-982
REF:REF<x>:LABel:COLor, 2-982
REF:REF<x>:LABel:FONT:BOLD, 2-983
REF:REF<x>:LABel:FONT:ITALic, 2-983
REF:REF<x>:LABel:FONT:SIZE, 2-983
REF:REF<x>:LABel:FONT:TYPE, 2-984
REF:REF<x>:LABel:FONT:UNDERline, 2-984
REF:REF<x>:LABel:NAMe, 2-985
REF:REF<x>:LABel:XPOS, 2-985
REF:REF<x>:LABel:YPOS, 2-986
REF:REF<x>:SOUrce, 2-986
REF<x>_D<x>:LABel:COLor, 2-991
REF<x>_D<x>:LABel:FONT:BOLD, 2-991
REF<x>_D<x>:LABel:FONT:ITALic, 2-992
REF<x>_D<x>:LABel:FONT:SIZE, 2-992
REF<x>_D<x>:LABel:FONT:TYPE, 2-993
REF<x>_D<x>:LABel:FONT:UNDERline, 2-993
REF<x>_D<x>:LABel:NAMe, 2-994
REF<x>_D<x>:LABel:XPOS, 2-994
REF<x>_D<x>:LABel:YPOS, 2-994
REF<x>_DALL:LABel:COLor, 2-987
REF<x>_DALL:LABel:FONT:BOLD, 2-987
REF<x>_DALL:LABel:FONT:ITALic, 2-988
REF<x>_DALL:LABel:FONT:SIZE, 2-988
REF<x>_DALL:LABel:FONT:TYPE, 2-989
REF<x>_DALL:LABel:FONT:UNDERline, 2-989
REF<x>_DALL:LABel:NAMe, 2-990
REF<x>_DALL:LABel:XPOS, 2-990
REF<x>_DALL:LABel:YPOS, 2-990
REM, 2-995
ROSc:SOUrce, 2-995
ROSc:STATE?, 2-996
*RST, 2-996

S

SAVe:EVENTtable:BUS, 2-998
SAVe:EVENTtable:CUSTom, 2-998
SAVe:EVENTtable:CUSTom:COMMents, 2-999
SAVe:EVENTtable:CUSTom:DATAFormat, 2-999
SAVe:EVENTtable:CUSTom:
INCLUDEREFs, 2-1000
SAVe:EVENTtable:MEASUrement, 2-1000
SAVe:EVENTtable:SEARCHTable, 2-1001
SAVe:IMAGe, 2-1002
SAVe:IMAGe:COMPosition, 2-1002
SAVe:IMAGe:VIEWTYpe, 2-1003
SAVe:PLOTData, 2-1003
SAVe:REPOrt, 2-1004
SAVe:REPOrt:COMMents, 2-1005
SAVe:SESsion, 2-1005
SAVe:SETUp, 2-1006
SAVe:SETUp:INCLUDEREFs, 2-1006
SAVe:WAVEform, 2-1007
SAVe:WAVEform:GATing, 2-1009
SAVe:WAVEform:GATing:RESAMPLErate, 2-1009
SAVe:WAVEform:SOURCELIst?, 2-1010
SAVEON:FILE:DEST, 2-1010

SAVEON:FILE:NAME, 2-1011
SAVEON:IMAGe, 2-1011
SAVEON:IMAGe:FILEFormat, 2-1012
SAVEON:TRIGger, 2-1013
SAVEON:WAVEform, 2-1014
SAVEON:WAVEform:FILEFormat, 2-1014
SAVEON:WAVEform:SOURce, 2-1015
SAVEONEVent:FILEDest, 2-1015
SAVEONEVent:FILEName, 2-1016
SAVEONEVent:IMAGe:FILEFormat, 2-1016
SAVEONEVent:WAVEform:FILEFormat, 2-1017
SAVEONEVent:WAVEform:SOUrce, 2-1018
SEARCH:ADDNew, 2-1018
SEARCH:DELete, 2-1019
SEARCH:DELETEALL, 2-1019
SEARCH:LIST?, 2-1019
SEARCH:SEARCH<x>:COPy, 2-1020
SEARCH:SEARCH<x>:NAVigate, 2-1020
SEARCH:SEARCH<x>:TOTAL?, 2-1021
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:CONDition, 2-1021
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:HIVALue, 2-1022
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:QUALifier, 2-1022
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:DATa:VALue, 2-1023
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:ERRTYPe, 2-1024
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:HIVALue, 2-1024
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:QUALifier, 2-1025
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:LABel:VALue, 2-1026
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SDI:VALue, 2-1026
SEARCH:SEARCH<x>:TRIGger:A:BUS:ARINC429A:SSM:VALue, 2-1027
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:CONDition, 2-1027
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HITDMVALue, 2-1028
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:HIVALue, 2-1029
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:OFFSet, 2-1029
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:QUALifier, 2-1030
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:TDMVALue, 2-1031
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:VALue, 2-1031
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUDio:DATa:WORD, 2-1032
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:CONDition, 2-1032
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:HIVALue, 2-1033
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:OFFSet, 2-1034
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:DATa:SIZe, 2-1035
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:DESTinationaddr:VALue, 2-1036
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:PROTOcol:VALue, 2-1037
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:IPHeader:SOUrceaddr:VALue, 2-1038
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:DESTination:VALue , 2-1039
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:ADDRess:SOUrce:VALue, 2-1040
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:HIVALue, 2-1041
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:MAC:LENgth:VALue, 2-1041
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QTAG:VALue, 2-1042
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:QUALifier, 2-1043
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:ACKnum:VALue, 2-1044
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:DESTinationport:VALue, 2-1045

SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SEQnum:VALue, 2-1046
SEARCH:SEARCH<x>:TRIGger:A:BUS:AUTOETHERnet:TCPHeader:SOUrceport:VALue, 2-1047
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:NRZ:CONDition, 2-1171
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:BLOCKDATa:VALue, 2-1173
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:CONDition, 2-1173
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:ECU:SENSor:VALue, 2-1174
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:REGion:A:VALue, 2-1175
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATa:REGion:B:VALue, 2-1176
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:DATABITs, 2-1176
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:ERRORTYPe, 2-1177
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:FUNCTIONCODETYPe, 2-1178
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:REGister:ADDRess:VALue, 2-1178
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:REGISTERADDRess, 2-1179
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:SENSor:ADDRess:VALue, 2-1180
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:SENSORSTATus:TYPe, 2-1180
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:PSIFIVe:STATus:VALue, 2-1181
SEARCH:SEARCH<x>:TRIGger:A:BUS:B:S8B10B:CONDition, 2-1183
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:CONDition, 2-1047
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:DIRection, 2-1048
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:OFFSet, 2-1049
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:QUALifier, 2-1050
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:SIZe, 2-1050
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:DATa:VALue, 2-1051
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:ERRType, 2-1051
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:BRSBit, 2-1052
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FD:ESIBit, 2-1053
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:FRAMEtype, 2-1053
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:MODe, 2-1054
SEARCH:SEARCH<x>:TRIGger:A:BUS:CAN:IDentifier:VALue, 2-1055
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:BLUe:VALue, 2-1055
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:CONDition, 2-1056
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:SIZe, 2-1057
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:DATa:VALue, 2-1057
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ERRor:TYPe, 2-1058
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:ESCAPEMODe:COMMand, 2-1058
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:GREen:VALue, 2-1059
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:MODe:TYPe, 2-1060
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:LIST, 2-1060
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PACKets:TYPe, 2-1063
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:NUMBer, 2-1064
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:PIXel:SEARCHOPTion, 2-1065
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:RED:VALue, 2-1065
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:WORDCOUNt:VALue, 2-1066
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CB, 2-1066
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:CR, 2-1067

SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YCBCR:Y, 2-1067
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:U, 2-1068
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:V, 2-1068
SEARCH:SEARCH<x>:TRIGger:A:BUS:DPHY:YUV:Y, 2-1069
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:CONDition, 2-1070
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:HIVALue, 2-1071
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:OFFSet, 2-1071
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:QUALifier, 2-1072
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:SIZe, 2-1073
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:DATa:VALue, 2-1073
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:DESTinationaddr:VALue, 2-1074
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:PROTOcol:VALue, 2-1075
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:IPHeader:SOUrceaddr:VALue, 2-1076
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:DESTination:VALue, 2-1077
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:ADDRess:SOUrce:VALue, 2-1077
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:HIVALue, 2-1078
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:MAC:LENgth:VALue, 2-1079
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:QTAG:VALue, 2-1080
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:ACKnum:VALue, 2-1080
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:DESTinationport:VALue, 2-1081
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SEQnum:VALue, 2-1082
SEARCH:SEARCH<x>:TRIGger:A:BUS:ETHERnet:TCPHeader:SOUrceport:VALue, 2-1083
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:HIVALue, 2-1083
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ADDress:VALue, 2-1084
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:CONDition, 2-1085
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:HIVALue, 2-1086
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:OFFSet, 2-1086
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:QUAlifier, 2-1087
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:SIZe, 2-1088
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:TYPe, 2-1088
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATa:VALue, 2-1089
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:DATABITS:SIZe, 2-1090
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDPoint:VALue, 2-1090
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ENDSEARch, 2-1091
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:DATABITS:SIZe, 2-1092
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOP:QUAlifier, 2-1093
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MAX:SIZe, 2-1093
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:EOPBITS:MIN:SIZe, 2-1094
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:ERRType, 2-1091
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:HANDSHAKEType, 2-1095
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:ADDress:VALue, 2-1095
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:COMMand, 2-1096
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:DATa:VALue, 2-1097
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:RAP:OPTion, 2-1097

SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SOFFRAMENUMber, 2-1098
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPECIALType, 2-1098
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:ET:VALue, 2-1099
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:HUB:VALue, 2-1100
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:PORT:VALue, 2-1100
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SC:VALue, 2-1101
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SPLit:SE:VALue, 2-1102
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNC:QUALifier, 2-1102
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MAX:SIZe, 2-1103
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCBITS:MIN:SIZe, 2-1104
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:SYNCSEARch, 2-1104
SEARCH:SEARCH<x>:TRIGger:A:BUS:EUSB:TOKENType, 2-1105
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CONDition, 2-1106
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:HIVALue, 2-1106
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:QUALifier, 2-1107
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:CYCLEcount:VALue, 2-1107
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:HIVALue, 2-1108
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:OFFSet, 2-1109
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:QUALifier, 2-1109
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:SIZe, 2-1110
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:DATa:VALue, 2-1111
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:EOFTYPE, 2-1111
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:ERRTYPE, 2-1112
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:HIVALue, 2-1112
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:QUALifier, 2-1113
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEID:VALue, 2-1113
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXray:FRAMEType, 2-1114
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CRC, 2-1114
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:CYCLECount, 2-1115
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:FRAMEID, 2-1116
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:INDBits, 2-1116
SEARCH:SEARCH<x>:TRIGger:A:BUS:FLEXRAY:HEADER:PAYLENgth, 2-1117
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:MODe, 2-1117
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:ADDRess:VALue, 2-1118
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:CONDition, 2-1119
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:DIRection, 2-1119
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:SIZe, 2-1120
SEARCH:SEARCH<x>:TRIGger:A:BUS:I2C:DATa:VALue, 2-1121
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:MODe, 2-1121
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ADDRess:VALue, 2-1122
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCR:VALue, 2-1123
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BCRType:VALue, 2-1123
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BPACKets, 2-1124
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:BRGTID:VALue, 2-1124
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CCODe:VALue, 2-1125
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:CONDition, 2-1125

SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:DIRection, 2-1126
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:SIZe, 2-1127
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DATa:VALue, 2-1128
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCR:VALue, 2-1128
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DCRType:VALue, 2-1129
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:DWORd:VALue, 2-1129
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ERRORTYPe, 2-1130
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:EVENTBYTe:VALue, 2-1131
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:FREQBYTe:VALue, 2-1131
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSLSb:VALue, 2-1132
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:GSMSb:VALue, 2-1132
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:ID:VALue, 2-1133
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:INACCBYTe:VALue, 2-1133
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXREAD:VALue, 2-1134
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXRTURN:VALue, 2-1135
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:MAXWRITe:VALue, 2-1135
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:PACKets, 2-1136
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SADDress:VALue, 2-1136
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:BROADCASTPacket, 2-1137
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SDR:DIRECTPacket, 2-1138
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATEBYTe:VALue, 2-1139
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:STATic:ADDRess, 2-1140
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:SUPPORTBYTe:VALue, 2-1140
SEARCH:SEARCH<x>:TRIGger:A:BUS:I3C:TESTMODe:VALue, 2-1141
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:CONDition, 2-1141
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:HIVALue, 2-1142
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:QUALifier, 2-1142
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:SIZe, 2-1143
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:DATa:VALue, 2-1144
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:ERRTYPE, 2-1144
SEARCH:SEARCH<x>:TRIGger:A:BUS:LIN:IDentifier:VALue, 2-1145
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:CONDition, 2-1145
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:SIZe, 2-1146
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:DATa:VALue, 2-1147
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:ERRor:TYPe, 2-1148
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:HEADER:VALue, 2-1148
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:packetOffData:VALue, 2-1149
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:SYNC:VALue, 2-1150
SEARCH:SEARCH<x>:TRIGger:A:BUS:MANChester:TRAILER:VALue, 2-1150
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ADDress:VALue, 2-1151
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:CONDition, 2-1151
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DATa:VALue, 2-1152
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:DEVicetype:VALue, 2-1153
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:ERRType, 2-1153
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:OPCode:VALue, 2-1154
SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:PHYSicaladdress:VALue, 2-1155

SEARCH:SEARCH<x>:TRIGger:A:BUS:MDIO:REGisteraddress:VALue, 2-1155
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:HIVALue, 2-1156
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:QUALifier, 2-1156
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:ADDRess:VALue, 2-1157
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:COUNt, 2-1158
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:PARity, 2-1158
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:SUBADdress, 2-1159
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:COMMAND:TRBit, 2-1159
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:CONDition, 2-1160
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:PARity, 2-1161
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:DATa:VALue, 2-1161
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:ERRTYPe, 2-1162
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:HIVALue, 2-1163
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:QUALifier, 2-1163
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:ADDRess:VALue, 2-1164
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BCR, 2-1165
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:BUSY, 2-1165
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:DBCA, 2-1166
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:INSTR, 2-1167
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:ME, 2-1167
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SRQ, 2-1168
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:SUBSF, 2-1169
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:BIT:TF, 2-1169
SEARCH:SEARCH<x>:TRIGger:A:BUS:MIL1553B:STATus:PARity, 2-1170
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:SIZe, 2-1171
SEARCH:SEARCH<x>:TRIGger:A:BUS:NRZ:DATa:VALue, 2-1172
SEARCH:SEARCH<x>:TRIGger:A:BUS:PARallel:DATa:VALue, 2-1172
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:CONDition, 2-1182
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:SIZe, 2-1182
SEARCH:SEARCH<x>:TRIGger:A:BUS:RS232C:DATa:VALue, 2-1183
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:DISParity, 2-1184
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:ERRor, 2-1185
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:CSYMVALue, 2-1186
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:EBITVALue, 2-1186
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:MODe, 2-1187
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDNeg, 2-1188
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITRDPos, 2-1189
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TBITVALue, 2-1189
SEARCH:SEARCH<x>:TRIGger:A:BUS:S8B10B:SYMbol:TYPe, 2-1190
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ADDRess:TYPe, 2-1191
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:CONDition, 2-1191
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:SIZe, 2-1192
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:DATa:VALue, 2-1193
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:ERROR:TYPe, 2-1193
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:FRAMe:TYPe, 2-1194

SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:STADDress:VALue, 2-1195
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:SUPervisory:FRAMETYPe, 2-1195
SEARCH:SEARCH<x>:TRIGger:A:BUS:SDLC:UNNumbered:FRAMETYPe, 2-1196
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:CONDition, 2-1197
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType, 2-1198
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:ERRType:CRC, 2-1198
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:HIVALue, 2-1199
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:QUALifier, 2-1200
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN1A:VALue, 2-1201
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:HIVALue, 2-1201
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:QUALifier, 2-1202
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:CHAN2B:VALue, 2-1203
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:HIVALue, 2-1204
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:QUALifier, 2-1205
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:COUNTer:VALue, 2-1206
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:INVERTNIBble:VALue, 2-1206
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:FAST:STATus:VALue, 2-1207
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:QUALifier, 2-1208
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:HIVALue , 2-1209
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:PAUSE:TICKs:VALue, 2-1210
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:HIVALue, 2-1210
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:QUALifier, 2-1211
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:DATA:VALue, 2-1212
SEARCH:SEARCH<x>:TRIGger:A:BUS:SENT:SLOW:IDentifier:VALue, 2-1213
SEARCH:SEARCH<x>:TRIGger:A:BUS:SOUrce, 2-1213
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONDition, 2-1214
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCHARTYPe, 2-1215
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:CONTROLCODETYPe, 2-1215
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:SIZe, 2-1216
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:DATa:VALue, 2-1217
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:ERRORTYPe, 2-1217
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPACEWIRe:TIMECode:VALue, 2-1218
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:CONDition, 2-1219
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:SIZe, 2-1219
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:DATa:VALue, 2-1220
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPI:SOURCETYpe, 2-1221
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:CONDition, 2-1221
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:SIZe, 2-1222
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:DATa:VALue, 2-1223
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:MASTERADDRess:VALue, 2-1224
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:NORESPonse, 2-1225
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:REGISTERADDRess:VALue, 2-1225
SEARCH:SEARCH<x>:TRIGger:A:BUS:SPMI:SLAVEADDRess:VALue, 2-1226
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:RESPonse, 2-1227
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:COMMand:TYPe, 2-1228
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:CONDition, 2-1229

SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:ERRor:TYPe, 2-1229
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLoad:TYPe, 2-1230
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:PAYLOAD:VALue, 2-1231
SEARCH:SEARCH<x>:TRIGger:A:BUS:SVID:SLAVE:ADDRESS, 2-1231
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:HIVALue, 2-1232
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ADDress:VALue, 2-1233
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:CONDition, 2-1233
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:HIVALue, 2-1234
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:OFFSet, 2-1235
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:QUALifier, 2-1235
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:SIZe, 2-1236
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:TYPe, 2-1237
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:DATa:VALue, 2-1237
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ENDPoint:VALue, 2-1238
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:ERRTYPE, 2-1239
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:HANDSHAKEType, 2-1239
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SOFFRAMENUMber, 2-1240
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPECIALType, 2-1241
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:ET:VALue, 2-1241
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:HUB:VALue, 2-1242
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:PORT:VALue, 2-1243
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SC:VALue, 2-1243
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:SPLIT:SE:VALue, 2-1244
SEARCH:SEARCH<x>:TRIGger:A:BUS:USB:TOKENType, 2-1245
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTDETectmethod, 2-1246
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLatency, 2-1246
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:BURSTLENGTH, 2-1247
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSActive, 2-1248
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSLevel, 2-1248
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSMode, 2-1249
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:CSSource, 2-1250
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATARate, 2-1251
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:DATASource, 2-1252
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:HYSteresis, 2-1253
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC1SOUrce:SYMBol, 2-1253
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC2SOUrce:SYMBol, 2-1254
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC3SOUrce:SYMBol, 2-1255
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:LOGIC4SOUrce:SYMBol, 2-1256
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MARgin, 2-1256
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MAXCAS, 2-1257
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:MINCas, 2-1258
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:POSTAMBLE:LENGth, 2-1259
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:PREAMBLE:TYPE, 2-1259
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:HIGH, 2-1260
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:LOW, 2-1261
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:DATA:MID, 2-1261

SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:HIGH, 2-1262
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:LOW, 2-1263
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLevel:STROBE:MID, 2-1263
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:REFLEVELMode, 2-1264
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STANdard, 2-1265
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:STROBESource, 2-1265
SEARCH:SEARCH<x>:TRIGger:A:DDRREAD:TOLERance, 2-1266
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTDETectmethod, 2-1267
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLatency, 2-1268
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:BURSTLENGTH, 2-1268
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSActive, 2-1269
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSLevel, 2-1270
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSMode, 2-1271
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:CSSource, 2-1271
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATARate, 2-1272
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:DATASource, 2-1273
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:HYSteresis, 2-1274
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC1SOUrce:SYMBol, 2-1275
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC2SOUrce:SYMBol, 2-1276
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC3SOUrce:SYMBol, 2-1276
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:LOGIC4SOUrce:SYMBol, 2-1277
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MARgin, 2-1278
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MAXCAS, 2-1279
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:MINCas, 2-1280
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:POSTAMBLE:LENGth, 2-1280
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:PREAMBLE:TYPE, 2-1281
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:HIGH, 2-1282
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:LOW, 2-1282
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:DATA:MID, 2-1283
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:HIGH, 2-1284
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:LOW, 2-1284
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLevel:STROBE:MID, 2-1285
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:REFLEVELMode, 2-1286
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STANdard, 2-1287
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:STROBESource, 2-1287
SEARCH:SEARCH<x>:TRIGger:A:DDRREADWRITE:TOLERance, 2-1288
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTDETectmethod, 2-1289
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLatency, 2-1290
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:BURSTLENGTH, 2-1290
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSActive, 2-1291
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSLevel, 2-1292

SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSMode, 2-1293
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:CSSource, 2-1293
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATARate, 2-1294
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:DATASource, 2-1295
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:HYSteresis, 2-1296
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC1SOUrce:SYMBol, 2-1297
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC2SOUrce:SYMBol, 2-1297
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC3SOUrce:SYMBol, 2-1298
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:LOGIC4SOUrce:SYMBol, 2-1299
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MARgin, 2-1300
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MAXCAS, 2-1300
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:MINCas, 2-1301
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:POSTAMBLE:LENGth, 2-1302
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:PREAMBLE:TYPE, 2-1303
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:HIGH, 2-1303
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:LOW, 2-1304
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:DATA:MID, 2-1305
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:HIGH, 2-1305
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:LOW, 2-1306
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLevel:STROBE:MID, 2-1307
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:REFLEVELMode, 2-1307
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STANdard, 2-1308
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:STROBESource, 2-1309
SEARCH:SEARCH<x>:TRIGger:A:DDRWRITE:TOLERance, 2-1310
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SLOpe, 2-1310
SEARCH:SEARCH<x>:TRIGger:A:EDGE:SOUrce, 2-1311
SEARCH:SEARCH<x>:TRIGger:A:EDGE:THReshold, 2-1312
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:CLOCk:THReshold, 2-1312
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:DELTatime, 2-1313
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:FUNCtion, 2-1313
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:INPUT:CLOCK:SOUrce, 2-1314
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:CH<x>, 2-1314
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:MATH<x>, 2-1315
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LEVel:REF<x>, 2-1315
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x>, 2-1316
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:CH<x>_D<x>, 2-1316
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:MATH<x>, 2-1317
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:LOGICPattern:REF<x>, 2-1317
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:POLarity, 2-1318
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:USEClockedge, 2-1319
SEARCH:SEARCH<x>:TRIGger:A:LOGIc:WHEn, 2-1319
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:HIGHLimit, 2-1320
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOGICQUALification, 2-1320
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:LOWLimit, 2-1321
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:POLarity, 2-1321
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:SOUrce, 2-1322

SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:THReshold, 2-1322
SEARCH:SEARCH<x>:TRIGger:A:PULSEWidth:WHEn, 2-1323
SEARCH:SEARCH<x>:TRIGger:A:RUNT:LOGICQUALification, 2-1324
SEARCH:SEARCH<x>:TRIGger:A:RUNT:POLarity, 2-1324
SEARCH:SEARCH<x>:TRIGger:A:RUNT:SOUrce, 2-1325
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:HIGH, 2-1325
SEARCH:SEARCH<x>:TRIGger:A:RUNT:THReshold:LOW, 2-1326
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WHEn, 2-1326
SEARCH:SEARCH<x>:TRIGger:A:RUNT:WIDth, 2-1327
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:EDGE, 2-1327
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:SOUrce, 2-1328
SEARCH:SEARCH<x>:TRIGger:A:SETHold:CLOCk:THReshold, 2-1329
SEARCH:SEARCH<x>:TRIGger:A:SETHold:HOLDTime, 2-1329
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:CH<x>, 2-1330
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:MATH<x>, 2-1330
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LEVel:REF<x>, 2-1331
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>, 2-1331
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:CH<x>_D<x>, 2-1332
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:MATH<x>, 2-1332
SEARCH:SEARCH<x>:TRIGger:A:SETHold:LOGICPattern:REF<x>, 2-1333
SEARCH:SEARCH<x>:TRIGger:A:SETHold:SETTime, 2-1334
SEARCH:SEARCH<x>:TRIGger:A:STATE, 2-1334
SEARCH:SEARCH<x>:TRIGger:A:STOPAcq, 2-1335
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:LOGICQUALification, 2-1335
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:POLarity, 2-1336
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:SOUrce, 2-1336
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:THReshold, 2-1337
SEARCH:SEARCH<x>:TRIGger:A:TIMEOut:TIMe, 2-1337
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:DELTATime, 2-1338
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:LOGICQUALification, 2-1338
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:POLarity, 2-1339
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:SOUrce, 2-1339
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:HIGH, 2-1340
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:THReshold:LOW, 2-1340
SEARCH:SEARCH<x>:TRIGger:A:TRANsition:WHEn, 2-1341
SEARCH:SEARCH<x>:TRIGger:A:TYPe, 2-1341
SEARCH:SEARCH<x>:TRIGger:A:WINdow:CROSSIng, 2-1343
SEARCH:SEARCH<x>:TRIGger:A:WINdow:LOGICQUALification, 2-1344
SEARCH:SEARCH<x>:TRIGger:A:WINdow:POLarity, 2-1344
SEARCH:SEARCH<x>:TRIGger:A:WINdow:SOUrce, 2-1345
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:HIGH, 2-1346
SEARCH:SEARCH<x>:TRIGger:A:WINdow:THReshold:LOW, 2-1346
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WHEn, 2-1347
SEARCH:SEARCH<x>:TRIGger:A:WINdow:WIDth, 2-1347
SEARCH:SEARCHx>:TRIGger:A:BUS:AUTOETHERnet:DATa:VALue, 2-1036
SEARCH:SELected, 2-1348
SEARCHTABle, 2-1348
SET?, 2-1349
SOCKETServer:ENAble, 2-1350
SOCKETServer:PORT, 2-1350
SOCKETServer:PROTOCol, 2-1351

*SRE, 2-1351
*STB?, 2-1352
SV:CH<x>:RF_AVErage:NUMAVg, 2-1353
SV:CH<x>:RF_MAGnitude:FORMat, 2-1353
SV:CH<x>:RF_PHASe:REFerence:DEGrees, 2-1354
SV:CH<x>:RF_PHASe:REFerence:POSition, 2-1355
SV:CH<x>:RF_PHASe:REFerence:TIMe, 2-1355
SV:CH<x>:RF_PHASe:WRAP:DEGrees, 2-1356
SV:CH<x>:RF_PHASe:WRAP:STATE, 2-1357
SV:CH<x>:SELect:RF_AVErage, 2-1357
SV:CH<x>:SELect:RF_FREQuency, 2-1358
SV:CH<x>:SELect:RF_MAGnitude, 2-1358
SV:CH<x>:SELect:RF_MAXHold, 2-1359
SV:CH<x>:SELect:RF_MINHold, 2-1360
SV:CH<x>:SELect:RF_NORMal, 2-1360
SV:CH<x>:SELect:RF_PHASe, 2-1361
SV:CH<x>:SELTrace, 2-1361
SV:CH<x>:SQUELCH:STATE, 2-1362
SV:CH<x>:SQUELCH:THReshold, 2-1363
SV:CH<x>:UNIts, 2-1364
SV:LOCKCenter, 2-1364
SV:LOCKSpectrum, 2-1365
SV:MARKER:PEAK:EXCURsion, 2-1365
SV:MARKER:PEAK:MAXimum, 2-1366
SV:MARKER:PEAK:STATE, 2-1367
SV:MARKER:PEAK:THReshold, 2-1367
SV:MARKER:REFERence, 2-1368
SV:MARKER:REFERence:AMPLITUDE?, 2-1368
SV:MARKER:REFERence:FREQuency?, 2-1369
SV:MARKER:TYPe, 2-1369
SV:RBW, 2-1370
SV:RBWMode, 2-1371
SV:RF_PHASe:REFerence:MASTer, 2-1371
SV:SPAN, 2-1372
SV:SPANRBWRatio, 2-1372
SV:WINDOW, 2-1373

T

TEKSecure, 2-1374
TIMe, 2-1374
TIMe:ZONe, 2-1375
TIMe:ZONe:UTCDELTa, 2-1375
TOTaluptime?, 2-1376
TOUCHSCReen:CALibrate, 2-1376
TOUCHSCReen:STATe, 2-1376
*TRG, 2-1377
TRIGger, 2-1377
TRIGger:{A|B}:BUS:B<x>:ARINC429A:CONDition, 2-1378
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:HIVALue, 2-1379
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:QUALifier, 2-1380
TRIGger:{A|B}:BUS:B<x>:ARINC429A:DATa:VALue, 2-1380
TRIGger:{A|B}:BUS:B<x>:ARINC429A:ERRTYPe, 2-1381
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:HIVALue, 2-1382
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:QUALifier, 2-1382
TRIGger:{A|B}:BUS:B<x>:ARINC429A:LABel:VALue, 2-1383
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SDI:VALue, 2-1384
TRIGger:{A|B}:BUS:B<x>:ARINC429A:SSM:VALue, 2-1384
TRIGger:{A|B}:BUS:B<x>:AUDio:CONDition, 2-1385
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HITDMVALue, 2-1385
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:HIVALue, 2-1386
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:OFFSet, 2-1387
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:QUALifier, 2-1387
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:TDMVALue, 2-1388
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:VALue, 2-1389
TRIGger:{A|B}:BUS:B<x>:AUDio:DATa:WORD, 2-1389
TRIGger:{A|B}:BUS:B<x>:CAN:CONDition, 2-1390
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:DIRection, 2-1391
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:OFFSet, 2-1391
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:QUALifier, 2-1392
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:SIZe, 2-1393
TRIGger:{A|B}:BUS:B<x>:CAN:DATa:VALue, 2-1394

TRIGger:{A|B}:BUS:B<x>:CAN:ERRType, 2-1394
TRIGger:{A|B}:BUS:B<x>:CAN:FD:
BRSBit, 2-1395
TRIGger:{A|B}:BUS:B<x>:CAN:FD:ESIBit, 2-1396
TRIGger:{A|B}:BUS:B<x>:CAN:
FRAMEtype, 2-1396
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:
MODe, 2-1397
TRIGger:{A|B}:BUS:B<x>:CAN:IDentifier:
VALue, 2-1398
TRIGger:{A|B}:BUS:B<x>:ETHERnet:
CONDition, 2-1398
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:
HIVALue, 2-1399
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:
OFFSet, 2-1400
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:
QUALifier, 2-1401
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:
SIZe, 2-1402
TRIGger:{A|B}:BUS:B<x>:ETHERnet:DATa:
VALue, 2-1402
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:
DESTinationaddr:VALue, 2-1403
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:
PROTOcol:VALue, 2-1404
TRIGger:{A|B}:BUS:B<x>:ETHERnet:IPHeader:
SOUrceaddr:VALue, 2-1405
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:
ADDRess:DESTination:VALue, 2-1406
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:
ADDRess:SOUrce:VALue, 2-1407
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:
LENgth:HIVALue, 2-1407
TRIGger:{A|B}:BUS:B<x>:ETHERnet:MAC:
LENgth:VALue, 2-1408
TRIGger:{A|B}:BUS:B<x>:ETHERnet:QTAG:
VALue, 2-1409
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:
ACKnum:VALue, 2-1410
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:
DESTinationport:VALue, 2-1411
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:
SEQnum:VALue, 2-1411
TRIGger:{A|B}:BUS:B<x>:ETHERnet:TCPHeader:
SOUrceport:VALue, 2-1412
TRIGger:{A|B}:BUS:B<x>:FLEXray:
CONDition, 2-1413
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:
HIVALue, 2-1414
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:
QUALifier, 2-1415
TRIGger:{A|B}:BUS:B<x>:FLEXray:CYCLEcount:
VALue, 2-1415
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:
HIVALue, 2-1416
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:
OFFSet, 2-1417
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:
QUALifier, 2-1417
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:
SIZe, 2-1418
TRIGger:{A|B}:BUS:B<x>:FLEXray:DATa:
VALue, 2-1419
TRIGger:{A|B}:BUS:B<x>:FLEXray:
EOFTYPE, 2-1419
TRIGger:{A|B}:BUS:B<x>:FLEXray:
ERRTYPE, 2-1420
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:
HIVALue, 2-1421
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:
QUALifier, 2-1422
TRIGger:{A|B}:BUS:B<x>:FLEXray:FRAMEID:
VALue, 2-1422
TRIGger:{A|B}:BUS:B<x>:FLEXray:
FRAMEType, 2-1423
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:
CRC, 2-1424
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:
CYCLEcount, 2-1424
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:
FRAMEID, 2-1425
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:
INDBits, 2-1426
TRIGger:{A|B}:BUS:B<x>:FLEXray:HEADER:
PAYLength, 2-1426
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:
MODe, 2-1427
TRIGger:{A|B}:BUS:B<x>:I2C:ADDRess:
VALue, 2-1427
TRIGger:{A|B}:BUS:B<x>:I2C:CONDition, 2-1428
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:
DIRection, 2-1429

TRIGger:{A|B}:BUS:B<x>:I2C:DATa:SIZe, 2-1429
TRIGger:{A|B}:BUS:B<x>:I2C:DATa:
VALue, 2-1430
TRIGger:{A|B}:BUS:B<x>:LIN:CONDition, 2-1431
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:
HIVALue, 2-1431
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:
QUALifier, 2-1432
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:SIZe, 2-1433
TRIGger:{A|B}:BUS:B<x>:LIN:DATa:
VALue, 2-1433
TRIGger:{A|B}:BUS:B<x>:LIN:ERRTYPE, 2-1434
TRIGger:{A|B}:BUS:B<x>:LIN:IDentifier:
VALue, 2-1434
TRIGger:{A|B}:BUS:B<x>:MIL1553B:
COMMAND:ADDRess:HIVALue, 2-1435
TRIGger:{A|B}:BUS:B<x>:MIL1553B:
COMMAND:ADDRess:QUALifier, 2-1436
TRIGger:{A|B}:BUS:B<x>:MIL1553B:
COMMAND:ADDRess:VALue, 2-1436
TRIGger:{A|B}:BUS:B<x>:MIL1553B:
COMMAND:COUNt, 2-1437
TRIGger:{A|B}:BUS:B<x>:MIL1553B:
COMMAND:PARity, 2-1437
TRIGger:{A|B}:BUS:B<x>:MIL1553B:
COMMAND:SUBADdress, 2-1438
TRIGger:{A|B}:BUS:B<x>:MIL1553B:
COMMAND:TRBit, 2-1439
TRIGger:{A|B}:BUS:B<x>:MIL1553B:
CONDition, 2-1439
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:
PARity, 2-1440
TRIGger:{A|B}:BUS:B<x>:MIL1553B:DATa:
VALue, 2-1441
TRIGger:{A|B}:BUS:B<x>:MIL1553B:
ERRTYPe, 2-1441
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:
ADDRess:HIVALue, 2-1442
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:
ADDRess:QUALifier, 2-1442
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:
ADDRess:VALue, 2-1443
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:
BCR, 2-1444
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:
BUSY, 2-1444
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:
DBCA, 2-1445
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:
INSTR, 2-1445
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:
ME, 2-1446
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:
SRQ, 2-1447
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:
SUBSF, 2-1447
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:BIT:
TF, 2-1448
TRIGger:{A|B}:BUS:B<x>:MIL1553B:STATus:
PARity, 2-1449
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:
LESSLimit, 2-1449
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:
MORELimit, 2-1450
TRIGger:{A|B}:BUS:B<x>:MIL1553B:TIMe:
QUALifier, 2-1451
TRIGger:{A|B}:BUS:B<x>:PARallel:DATa:
VALue, 2-1451
TRIGger:{A|B}:BUS:B<x>:RS232C:
CONDition, 2-1452
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:
SIZe, 2-1452
TRIGger:{A|B}:BUS:B<x>:RS232C:DATa:
VALue, 2-1453
TRIGger:{A|B}:BUS:B<x>:SENT:
CONDition, 2-1454
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType, 2-1454
TRIGger:{A|B}:BUS:B<x>:SENT:ERRType:
CRC, 2-1455
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:
HIVALue, 2-1456
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:
QUALifier, 2-1456
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN1A:
VALue, 2-1457
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:
HIVALue, 2-1458
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:
QUALifier, 2-1459
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:CHAN2B:
VALue, 2-1460
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:
HIVALue, 2-1460

TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:QUALifier, 2-1461
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:COUNTer:VALue, 2-1462
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:INVERTNIBble:VALue, 2-1463
TRIGger:{A|B}:BUS:B<x>:SENT:FAST:STATus:VALue, 2-1464
TRIGger:{A|B}:BUS:B<x>:SENT:PAUSE:QUALifier, 2-1464
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:HIVALue, 2-1465
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:QUALifier, 2-1466
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:DATA:VALue, 2-1467
TRIGger:{A|B}:BUS:B<x>:SENT:SLOW:IDentifier:VALue, 2-1468
TRIGger:{A|B}:BUS:B<x>:SPI:CONDition, 2-1468
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:SIZe, 2-1469
TRIGger:{A|B}:BUS:B<x>:SPI:DATa:VALue, 2-1469
TRIGger:{A|B}:BUS:B<x>:SPMI:CONDition, 2-1470
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:SIZe, 2-1471
TRIGger:{A|B}:BUS:B<x>:SPMI:DATa:VALue, 2-1472
TRIGger:{A|B}:BUS:B<x>:SPMI:MASTERADDRess:VALue, 2-1473
TRIGger:{A|B}:BUS:B<x>:SPMI:NORESPonse, 2-1473
TRIGger:{A|B}:BUS:B<x>:SPMI:REGISTERADDRess:VALue, 2-1474
TRIGger:{A|B}:BUS:B<x>:SPMI:SLAVEADDRess:VALue, 2-1475
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:HIVALue, 2-1475
TRIGger:{A|B}:BUS:B<x>:USB:ADDRess:VALue, 2-1476
TRIGger:{A|B}:BUS:B<x>:USB:CONDition, 2-1477
TRIGger:{A|B}:BUS:B<x>:USB:DATa:HIVALue, 2-1478
TRIGger:{A|B}:BUS:B<x>:USB:DATa:OFFSet, 2-1478
TRIGger:{A|B}:BUS:B<x>:USB:DATa:QUALifier, 2-1479
TRIGger:{A|B}:BUS:B<x>:USB:DATa:SIZe, 2-1480
TRIGger:{A|B}:BUS:B<x>:USB:DATa:TYPe, 2-1480
TRIGger:{A|B}:BUS:B<x>:USB:DATa:VALue, 2-1481
TRIGger:{A|B}:BUS:B<x>:USB:ENDPoint:VALue, 2-1482
TRIGger:{A|B}:BUS:B<x>:USB:ERRType, 2-1482
TRIGger:{A|B}:BUS:B<x>:USB:HANDSHAKEType, 2-1483
TRIGger:{A|B}:BUS:B<x>:USB:SOFFRAMENUMber, 2-1484
TRIGger:{A|B}:BUS:B<x>:USB:SPECIALType, 2-1484
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:ET:VALue, 2-1485
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:HUB:VALue, 2-1486
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:PORT:VALue, 2-1487
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SC:VALue, 2-1487
TRIGger:{A|B}:BUS:B<x>:USB:SPLit:SE:VALue, 2-1488
TRIGger:{A|B}:BUS:B<x>:USB:TOKENType, 2-1489
TRIGger:{A|B}:BUS:SOUrce, 2-1490
TRIGger:{A|B}:EDGE:COUPling, 2-1490
TRIGger:{A|B}:EDGE:SLOpe, 2-1491
TRIGger:{A|B}:EDGE:SOUrce, 2-1492
TRIGger:{A|B}:LEVel:CH<x>, 2-1492
TRIGger:{A|B}:LOGIc:DELTatime, 2-1493
TRIGger:{A|B}:LOGIc:FUNCtion, 2-1493
TRIGger:{A|B}:LOGIc:INPut:CLOCk:SOUrce, 2-1494
TRIGger:{A|B}:LOGIc:POLarity, 2-1495
TRIGger:{A|B}:LOGIc:USECLockedge, 2-1495
TRIGger:{A|B}:LOGIc:WHEn, 2-1496
TRIGger:{A|B}:LOGICPattern:{CH<x>|CH<x>_D<x>}, 2-1496
TRIGger:{A|B}:LOWerthreshold:CH<x>, 2-1497
TRIGger:{A|B}:PULSEWidth:HIGHLimit, 2-1498
TRIGger:{A|B}:PULSEWidth:LOGICQUALification, 2-1498
TRIGger:{A|B}:PULSEWidth:LOWLimit, 2-1499
TRIGger:{A|B}:PULSEWidth:POLarity, 2-1499
TRIGger:{A|B}:PULSEWidth:SOUrce, 2-1500

TRIGger:{A|B}:PULSEWidth:WHEn, 2-1500
TRIGger:{A|B}:RUNT:
LOGICQUALification, 2-1501
TRIGger:{A|B}:RUNT:POLarity, 2-1502
TRIGger:{A|B}:RUNT:SOUrce, 2-1502
TRIGger:{A|B}:RUNT:WHEn, 2-1503
TRIGger:{A|B}:RUNT:WIDth, 2-1504
TRIGger:{A|B}:SETHold:CLOCk:EDGE, 2-1504
TRIGger:{A|B}:SETHold:CLOCk:SOUrce, 2-1505
TRIGger:{A|B}:SETHold:HOLDTime, 2-1505
TRIGger:{A|B}:SETHold:SETTime, 2-1506
TRIGger:{A|B}:SETHOLDLogicval:
{CH<x>|CH<x>_D<x>}, 2-1506
TRIGger:{A|B}:TIMEOut:
LOGICQUALification, 2-1507
TRIGger:{A|B}:TIMEOut:POLarity, 2-1508
TRIGger:{A|B}:TIMEOut:SOUrce, 2-1508
TRIGger:{A|B}:TIMEOut:TIMe, 2-1509
TRIGger:{A|B}:TRANsition:DELTatime, 2-1509
TRIGger:{A|B}:TRANsition:
LOGICQUALification, 2-1510
TRIGger:{A|B}:TRANsition:POLarity, 2-1510
TRIGger:{A|B}:TRANsition:SOUrce, 2-1511
TRIGger:{A|B}:TRANsition:WHEn, 2-1511
TRIGger:{A|B}:TYPe, 2-1512
TRIGger:{A|B}:UPPerthreshold:CH<x>, 2-1513
TRIGger:{A|B}:WINdow:CROSSIng, 2-1514
TRIGger:{A|B}:WINdow:
LOGICQUALification, 2-1515
TRIGger:{A|B}:WINdow:SOUrce, 2-1515
TRIGger:{A|B}:WINdow:WHEn, 2-1516
TRIGger:{A|B}:WINdow:WIDth, 2-1516
TRIGger:{A|B|B:RESET}, 2-1378
TRIGger:A:HOLDoff:BY, 2-1517
TRIGger:A:HOLDoff:TIMe, 2-1517
TRIGger:A:LOGICQUALification, 2-1518
TRIGger:A:MODe, 2-1519
TRIGger:AUXLevel, 2-1519
TRIGger:B:BY, 2-1520
TRIGger:B:EVENTS:COUNt, 2-1520
TRIGger:B:RESET, 2-1521
TRIGger:B:RESET:EDGE:COUPling, 2-1521
TRIGger:B:RESET:EDGE:LEVel, 2-1522
TRIGger:B:RESET:EDGE:SLOpe, 2-1522
TRIGger:B:RESET:EDGE:SOUrce, 2-1523
TRIGger:B:RESET:TIMEOut:TIMe, 2-1523
TRIGger:B:RESET:TYPe, 2-1524
TRIGger:B:STATE, 2-1524
TRIGger:B:TIMe, 2-1525
TRIGger:STATE?, 2-1525
*TST?, 2-1526

U

UNDO, 2-1526
UNLock, 2-1527
USBDevice:CONFigure, 2-1527

V

VERBose, 2-1528
VERTical:DESKew:FROM:
CUSTOMPROPAgation, 2-1529
VERTical:DESKEW:FROM:SOUrce, 2-1529
VERTical:DESKEW:STATIC, 2-1530
VERTical:DESKew:TO:
CUSTOMPROPAgation, 2-1530
VERTical:DESKEW:TO:SOUrce, 2-1531
VISual:AREA<x>:ASPEctratio, 2-1531
VISual:AREA<x>:FLIP:HORizontal, 2-1532
VISual:AREA<x>:FLIP:VERTical, 2-1532
VISual:AREA<x>:HEIGht, 2-1532
VISual:AREA<x>:HITType, 2-1533
VISual:AREA<x>:RESET, 2-1534
VISual:AREA<x>:ROTAtion, 2-1534
VISual:AREA<x>:SHAPE, 2-1534
VISual:AREA<x>:SOUrce, 2-1535
VISual:AREA<x>:VERTICES, 2-1536
VISual:AREA<x>:WIDTH, 2-1536
VISual:AREA<x>:XPOSition, 2-1537
VISual:AREA<x>:YPOSition, 2-1537
VISual:DELETEALL, 2-1538
VISual:ENAble, 2-1538
VISual:EQUation, 2-1539
VISual:SHOWAReas, 2-1539
VISual:SHOWCRiteria, 2-1540
VISual:SHOWEQuation, 2-1540

W

*WAI, 2-1541
WAVFrm?, 2-1541
WFMOutpre?, 2-1542
WFMOutpre:ASC_Fmt?, 2-1542
WFMOutpre:BIT_Nr, 2-1543
WFMOutpre:BN_Fmt, 2-1544

WFMOutpre:BYT_Nr, 2-1544
WFMOutpre:BYT_Or, 2-1545
WFMOutpre:CENTERFREQuency?, 2-1545
WFMOutpre:DOMain?, 2-1546
WFMOutpre:ENCdg, 2-1546
WFMOutpre:NR_Pt?, 2-1547
WFMOutpre:PT_Fmt?, 2-1548
WFMOutpre:PT_Off?, 2-1548
WFMOutpre:RESample, 2-1549
WFMOutpre:SPAN?, 2-1549
WFMOutpre:WFId?, 2-1550
WFMOutpre:WFMTYPe?, 2-1551
WFMOutpre:XINcr?, 2-1552
WFMOutpre:XUNit?, 2-1552
WFMOutpre:XZEro?, 2-1553
WFMOutpre:YMULt?, 2-1553
WFMOutpre:YOFf?, 2-1554
WFMOutpre:YUNit?, 2-1554
WFMOutpre:YZEro?, 2-1555