Keysight M8070A System Software for M8000 Series of BER Test Solutions

Programming Guide

KEYSIGHT TECHNOLOGIES

Notices

© Keysight Technologies 2016

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies as governed by United States and international copyright laws.

Trademarks

PCI Express® and PCIe® are registered trademarks of PCI-SIG.

Manual Part Number

M8020-91040

Edition

Edition 5.1, March 2016

Keysight Technologies Deutschland GmbH

Herrenberger Strasse 130,

71034 Böblingen, Germany

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement

("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS," AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR ANY INFORMATION CONTAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT WILL CONTROL.

Safety Notices

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Safety Summary

The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings or operating instructions in the product manuals violates safety standards of design, manufacture and intended use of the instrument. Keysight Technologies assumes no liability for the customer's failure to comply with these requirements. Product manuals are provided with your instrument on CD-ROM and/or in printed form. Printed manuals are an option for many products. Manuals may also be available on the Web. Go to www.keysight.com and type in your product number in the Search field at the top of the page.

General

This product is a Safety Class 1 instrument (provided with a protective earth terminal). The protective features of this product may be impaired if it is used in a manner not specified in the operation instructions.

All Light Emitting Diodes (LEDs) used in this product are Class 1 LEDs as per IEC 60825-1.

Environment Conditions

This instrument is intended for indoor use in an installation category II, pollution degree 2 environment. It is designed to operate at a maximum relative humidity of 95% and at altitudes of up to 2000 meters.

Refer to the specifications tables for the ac mains voltage requirements and ambient operating temperature range.

Before Applying Power

Verify that all safety precautions are taken. The power cable inlet of the instrument serves as a device to disconnect from the mains in case of hazard. The instrument must be positioned so that the operator can easily access the power cable inlet. When the instrument is rack mounted the rack must be provided with an easily accessible mains switch.

Ground the Instrument

To minimize shock hazard, the instrument chassis and cover must be connected to an electrical protective earth ground. The instrument must be connected to the ac power mains through a grounded power cable, with the ground wire firmly connected to an electrical ground (safety ground) at the power outlet. Any interruption of the protective (grounding) conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in personal injury.

Do Not Operate in an Explosive Atmosphere

Do not operate the instrument in the presence of flammable gases or fumes.

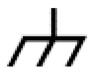
Do Not Remove the Instrument Cover

Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made only by qualified personnel.

Instruments that appear damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.

Safety Symbols

Table 1 Safety Symbol

Symbol	Description
	Indicates warning or caution. If you see this symbol on a product, you must refer to the manuals for specific Warning or Caution information to avoid personal injury or damage to the product.
	Frame or chassis ground terminal. Typically connects to the equipment's metal frame.
	KC is the Korean certification mark to demonstrate that the equipment is Class A suitable for professional use and is for use in electromagnetic environments outside of the home.
	Indicates that antistatic precautions should be taken.
	Indicates the time period during which no hazardous or toxic substance elements are expected to leak or deteriorate during normal use. Forty years is the expected useful life of the product.
	The RCM Mark is a compliance mark to the ACMA (Australian Spectrum Management Agency). This indicates compliance with all Australian EMC regulatory information.

Compliance and Environmental Information

Table 2 Compliance and Environmental Information

Safety Symbol	Description
	This product complies with WEEE Directive (2002/96/EC) marking requirements. The affixed label indicates that you must not discard this electrical/electronic product in domestic household waste. Product Category: With reference to the equipment types in WEEE Directive Annex I, this product is classed as a “Monitoring and Control instrumentation” product. Do not dispose in domestic household waste. To return unwanted products, contact your local Keysight office, or see http://about.keysight.com/en/companyinfo/environment/takeback.shtml for more information.

Contents

1 Programming Basics

KEYSIGHT TECHNOLOGIES

Modular Configuration

In order to describe remote programming for the M8020A and M8030A, the modular configuration must be discussed.

Both M8020A and M8030A are modular instruments. They can consist of several numbers of components. A component can be an AXIe chassis, modules, channels, etc. All these components ‘form’ an instrument. These instruments can be controlled by M8070A system software.

The M8020A instrument supports the following modules.

- M8041A high-performance BERT generator-analyzer-clock 8/16 Gb/s
- M8051A high-performance BERT generator-analyzer 8/16 Gb/s
- M8061A multiplexer 2:1 with de-emphasis 32 Gb/s
- M8062A 32Gb/s Front-end for J-BERT M8020A High-Performance BERT

The M8020A modules must be installed in the M9505A 5-slot chassis.

The M8030A instrument supports the following modules.

- M8041A high-performance BERT generator-analyzer-clock 8/16 Gb/s
- M8051A high-performance BERT generator-analyzer 8/16 Gb/s
- M8192A Multi-channel synchronization module

The M8030A modules must be installed in the M9514A AXIe 14-slot chassis.

For complete details on the features and hardware components of each of the above mentioned modules, refer to *M8020A and M8030A Getting Started Guide*.

M8020A Modular Configuration

The following section describes and illustrates various setup combinations in which you can install the M8020A modules.

The following configurations are possible in an M9505A 5-slot chassis:

- 1 or 2-channel, 16 Gb/s - (1) M8041A
- 3 or 4-channel, 16 Gb/s - (1) M8041A + (1) M8051A
- 1-channel, 32 Gb/s (Pattern Generator only) - (1) M8041A + (1) M8061A
- 1-channel, 32 Gb/s (Pattern Generator only or full BERT) -(1) M8041A + (1) M8062A

1 or 2-channel System Configuration

The 1 or 2-channel configuration is a single channel system (a second channel can be added with license) consisting of the 5-slot M9505A AXIe Chassis and an M8041A module. The M8041A occupies three slots. A maximum of two M8020A modules can be installed in a 5-slot chassis.

Figure 1 M8020A configuration for 16 Gb/s BERT with 1 to 2 channel

3 or 4-channel System Configuration

The four channel configuration consists of the 5-slot M9505A AXIe Chassis, an M8041A module, and an M8051A module. The M8041A occupies three slots and the M8051A occupies two slots.

NOTE This configuration requires a cable (provided with the M8051A) that connects the M8041A SYNC OUT to the M8051A SYNC IN to synchronize the two modules to a common system clock.

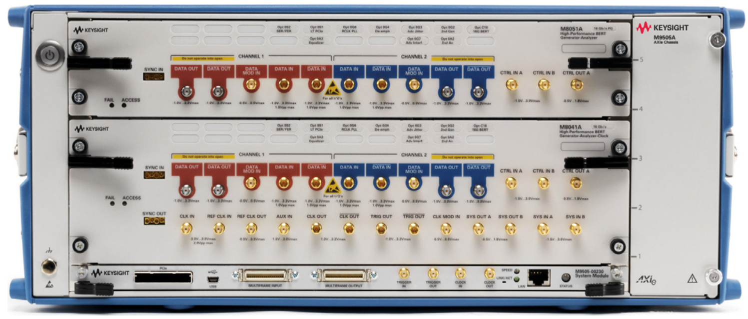


Figure 2 M8020A configuration for 16 Gb/s BERT for up to 4 channels

1-channel, 32 Gb/s (Pattern Generator Only)

A typical configuration using the M8061A 32 Gb/s multiplexer with de-emphasis consists of the 5-slot M9505A AXIe Chassis, an M8041A module, and an M8061A module. The M8041A occupies three slots and the M8061A occupies two slots.

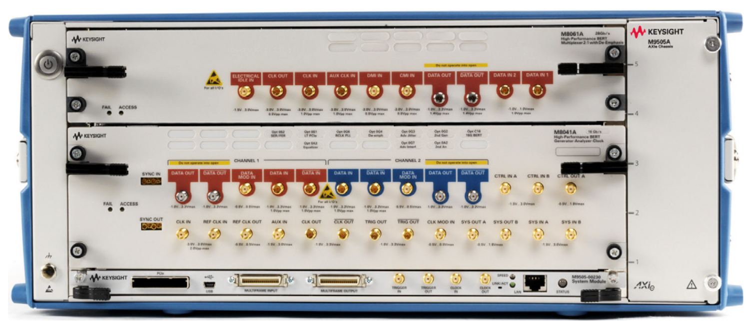


Figure 3 M8020A configuration for 32 Gb/s BERT (external de-multiplexer is recommended)

1-channel, 32 Gb/s (Pattern Generator only or full BERT)

A typical configuration for an M8020A 32 Gb/s full BERT consists of the 5-slot M9505A AXIe Chassis, an M8041A module, and an M8062A module. The M8041A occupies three slots and the M8062A occupies two slots.

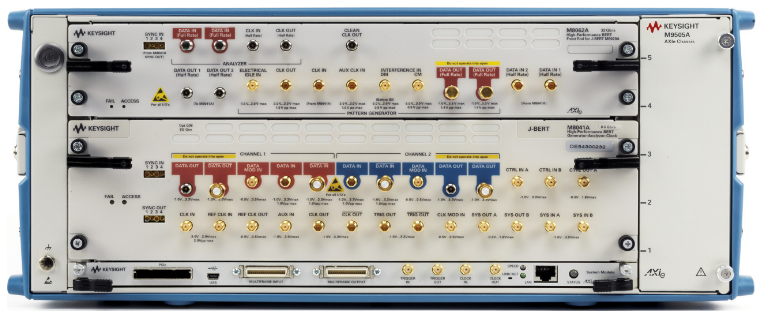


Figure 4 M8020A configuration for 32 Gb/s BERT

Modular Configuration of M8030A

The M8030A is a modular test solution which can be tailored to specific needs from two channels with one M8041A to up to 10 channels.

The modules must be installed in the M9514A AXIe 14-slot chassis as shown in Table 3 on page -17:

Table 3 M8030A modules' arrangement in the M9514A AXIe chassis

Slot Number	Module
# 1	For M8030A-BU1, M9536A AXIe embedded controller. For M8030A-BU2, this slot is empty and covered with filler front-panel
# 2-4	M8041A module
# 5-6	M8051A module
# 7	M9521A AXIe system module, always included in M8030A-BU1 or M8030A-BU2, mandatory
# 8-9	M8051A module

Slot Number	Module
# 10-11	M8051A module
# 12-13	M8051A module
# 14	M8192A multi-channel synchronization module, mandatory

Figure 5 on page -18 shows an example of modules arrangement in the M9514A AXIe 14-slot chassis.

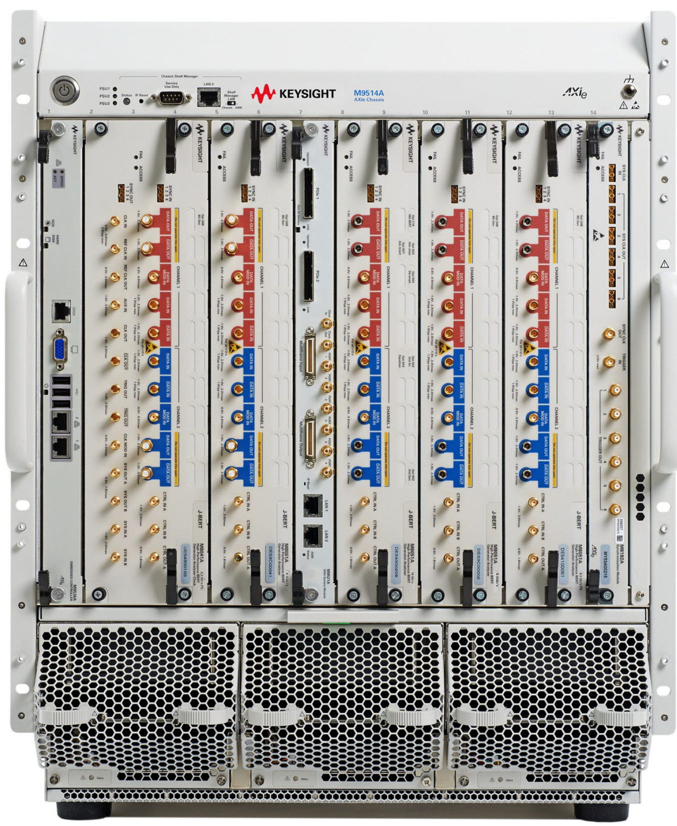


Figure 5 Example of M8030A module arrangement

Controlling Multiple Instruments

The M8070A system software (3.x.x.x version) allows a PC to control different connected instruments. The possible setup combinations are described and illustrated in this section.

A PC can be connected to an AXI 5 Slot frame and runs a M8041A as clock/data module and a M8051A as data module. It's a four channel instrument. The M8051A module can be replaced by a M8061A or by a M8062A, depending upon the requirement.

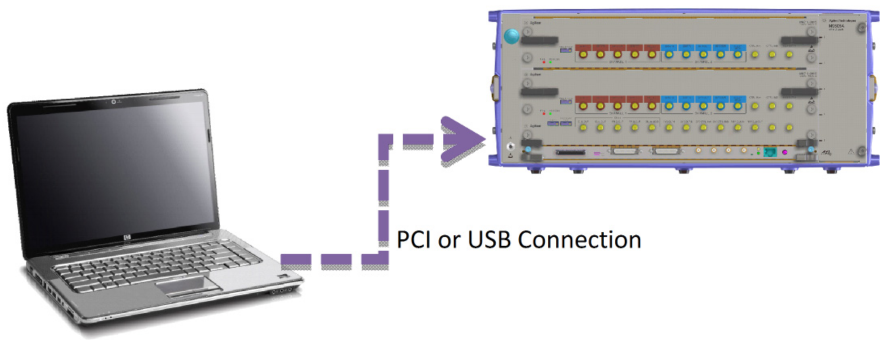


Figure 6 PC connected to AXI 5 slot frame

It is also possible to connect a PC to smaller AXI frame (2 slots) as shown in the following figure:

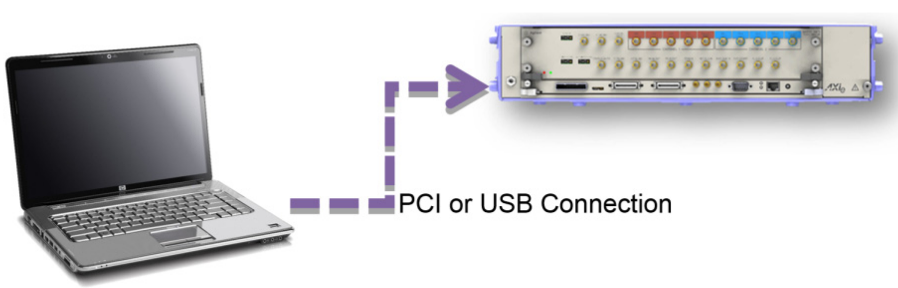


Figure 7 PC connected to AXI 2 slot frame

Also, it is also possible to connect a PC to large AXI frame (14 slots) as shown in the following figure:

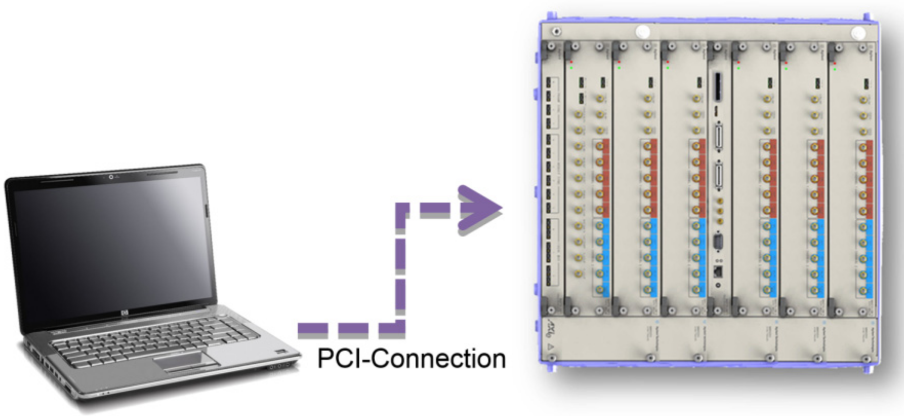


Figure 8 PC connected to AXI 14 slot frame

All these configurations are controlled by a single instance of a M8070A system software.

Controlling Multiple Instruments by Multiple Instances M8070A System Software

A single PC can also control multiple instruments by multiple instance of the M8070A software. For every connected instrument a separate instance of the M8070A software has to be started. You have to pass command line options to the M8070A system software to address/specify connected instruments.

Instrument settings, pattern etc. are stored in a so called workspace. To avoid conflicts between multiple running M8070A system software instances working on a single workspace you must specify a new name for every workspace settings are saved. For every running M8070A software, a separate workspace must be defined.

Following are some possible configurations:

1 **One PC with Connected to Multiple Instruments**

Figure 9 on page -21 shows one PC connected to multiple instruments connected via USB.

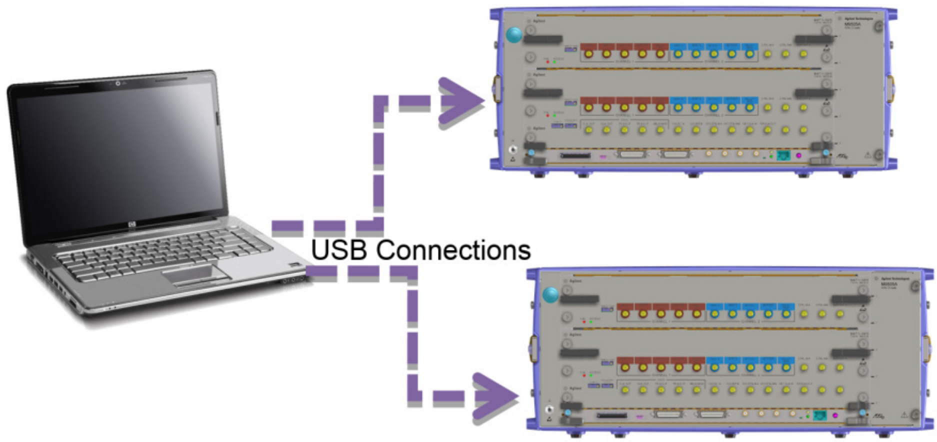


Figure 9 PC connected to multiple instruments via USB

For this configuration you have to start the M8070A software twice with different command line options.

1 Start a command line window

2 Type in the following syntax:

```
> agilent.M8070A.exe /chassisid 1 /workspace
"InstrumentOne"
> agilent.M8070A.exe /chassisid 2 /workspace
"InstrumentTwo"
```

In this configuration only one PC is required to run two M8070A instances simultaneously and both control their assigned instrument.

2 **One PC with Multiple Instruments in a M9514A Frame**

Figure 10 on page -22 shows two instruments (shaded transparent rectangles) hosted by a M9514A and controlled by a single PC only.

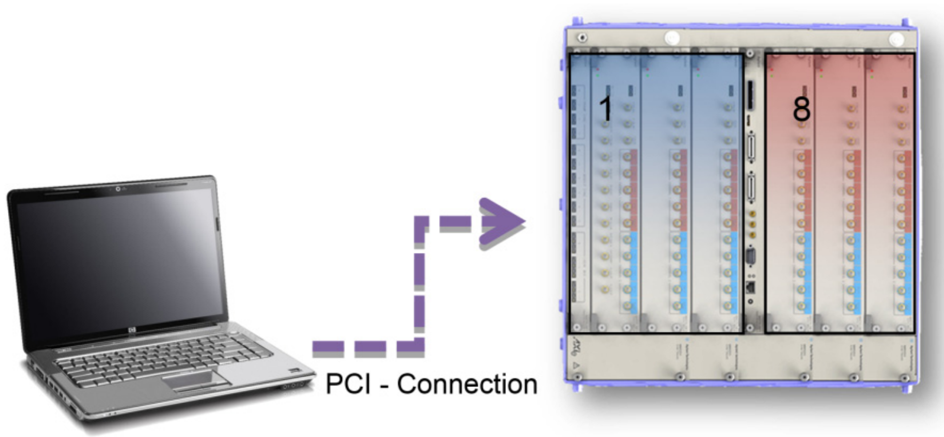


Figure 10 PC connected to AXI frame 14 slot via PCI

In this setup, the two instruments (shaded transparent rectangles) are hosted by a M9514A AXI frame and controlled by a single PC. The first instrument will be addressed by chassis identifier 1 and slot number 1 and the second instrument will be addressed by chassis identifier 1 and slot number 8 (slot numbers are printed above a slot). The slot number specifies the first module to be built an instrument.

For starting the software specify following command line option

```
> agilent.M8070A.exe /chassisid 1 /slotnumber 1
/workspace "InstrumentOne"
> agilent.M8070A.exe /chassisid 1 /slotnumber 2
/workspace "InstrumentTwo"
```

3 Mix Connection (One PC with Multiple M8070A Instances and Multiple Instruments in a M9514A Frame)

In this configuration, multiple Instruments in a M9514A frame are connected via PCI-connection to PC while Multiple M8070A instances are connected via USB connection to PC. See Figure 11 on page -23.

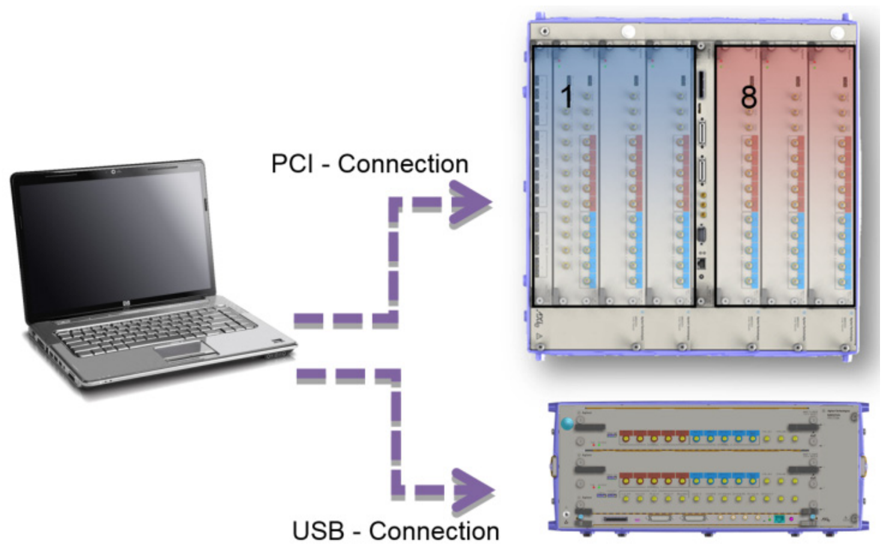


Figure 11 PC connected to multiple instruments via USB and PCI

To run this configuration type in the following syntax:

```
> agilent.M8070A.exe /chassisid 1 /slotnumber 1
/workspace "InstrumentOne"
> agilent.M8070A.exe /chassisid 1 /slotnumber 8
/workspace "InstrumentTwo"
> agilent.M8070A.exe /chassisid 2 /workspace
"InstrumentThree"
```

Remote Control of Multiple M8070A Software Instances

You can also use SCPI to remotely control multiple instances of M8070A software. To know the VISA Resource Strings for SCPI Access, click on the front panel on **Utilities** > **SCPI Server Information...**.

The dialog shows the VISA Resource Strings for SCPI access as shown in the following figures:

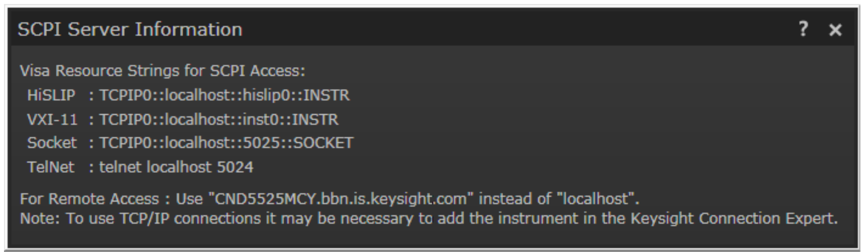


Figure 12 SCPI Server Information showing VISA Resource Strings for SCPI Access

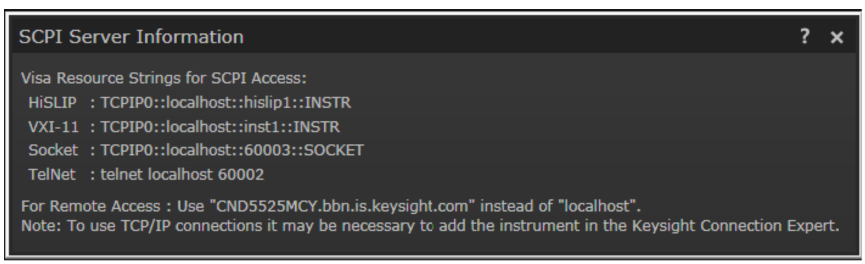


Figure 13 SCPI Server Information showing VISA Resource Strings for SCPI Access

Command Line Options

The following command line option are required for connecting a PC with instrument(s).

/chassisid	Chassis Identifier - defines a number for using the corresponding frame
/slotnumber	Slot Number – defines a slot number of the first module to be built an instrument
/workspace	Workspace Name – defines a name for a workspace the settings are stored

Creating M8070A Software Shortcut on the Desktop

Follow the given steps to create M8070A software shortcut on desktop:

1 Open file explorer and go to *C:\Program Files\Keysight\M8070A\bin* location as shown in the Figure 14 on page -25:

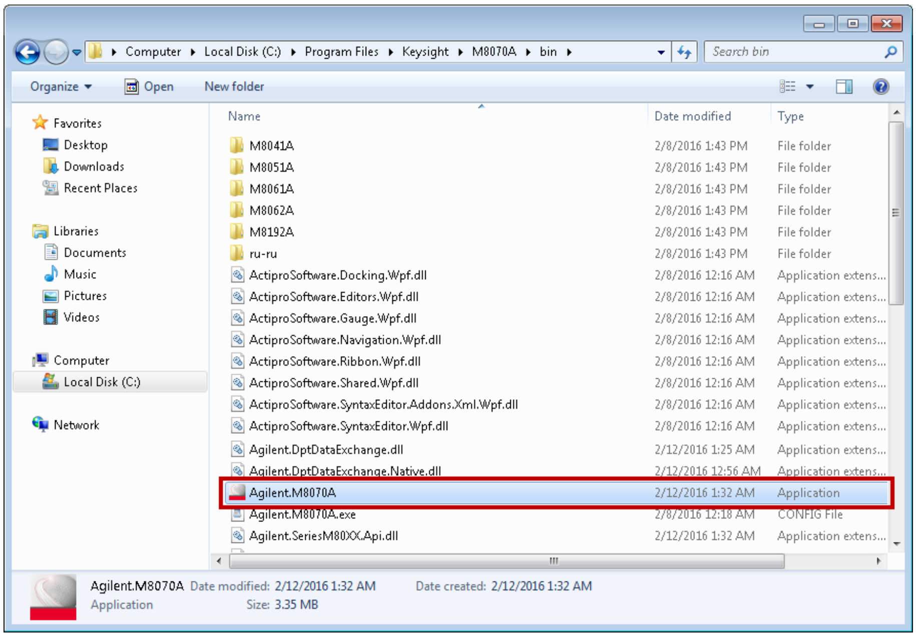


Figure 14 File explorer

2 Click on the highlighted executable with the right mouse button and move the cursor with pressed button on the desktop, release the mouse button and select in the context menu 'Create shortcut'.

3 Rename the created shortcut to a meaningful name, right click and select in the context menu 'Properties'.

4 Choose the 'Shortcut' tab and change the 'Target' entry field to for example

"C:\Program Files\Keysight\M8070A\bin\Agilent.M8070A.exe" /chassisid 1 /slotnumber 1 /workspace "InstrumentOne"

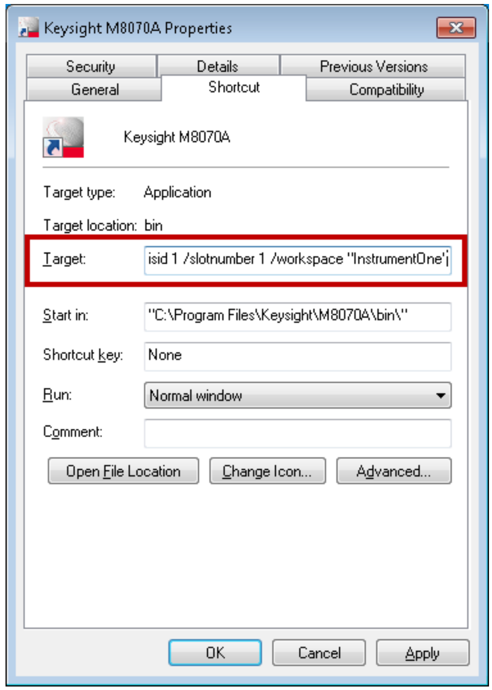


Figure 15 Changing shortcut target

5 Click ‘Apply’.

You can now start the ‘configured’ M8070A system software by a simple double click on the desktop icon.

Using Identifiers

There are three different types of Identifiers:

- Location
- Group Name
- Measurement Name

Location Identifiers

When an instrument has several output channels with identical capabilities or a subset of capabilities and has more than one module, the selection of which module/channel to use is done using its location identifier.

The location identifier is a predefined notation that can be used to address certain properties, like high level, low level etc., of channels. The location identifier is surrounded by single quotes (‘identifier’) or double quotes (“identifier”). A location identifier will be appended to a SCPI command followed by its argument(s). The location identifier addresses the corresponding components. The location identifier consists of two parts: the first part specifies the module, the second part addresses, in most cases, a visible ‘component’ of a module, like an input or output connector.

For example, SCPI command :OUTP ‘M1.DataOut1’, OFF turns channel 1 (DataOut1) of module 1 (M1) output off.

If a SCPI command is to be sent to a specific module only and not a specific channel, simply use “M1” or “M2” as the location identifier.

If you want to set a value that acts on all components of a modular instrument (for example, set all jitter sources to off), omit the location identifier.

Multiple Source Per Channel Identifier

Some features have multiple sources per channel. To identify these sources, the “(*)” suffix is used.

For example, there are two periodic jitter sources per channel in a module. The following SCPI command specifies periodic jitter source 2 as the source for module 1/data out channel 1:

[:SOURce]:JITTer:PERiodic2[:STATe] ‘m1.dataout1’,ON

Group Name Identifiers

Group name identifiers are used to affect a group of inputs or outputs using a single SCPI command. That is, all properties belonging to these identifiers can be addressed simultaneously using the group name.

The following SCPI command shows the syntax for creating a group name then adding the desired location identifiers.

SYSTem:INSTrument:GROup:DEFine
'GroupName','Identifier','Identifier','Identifier','...'

The following is an example showing 'Outputs' as the group name followed by 'M1.DataOut1' and 'M1.DataOut2' location identifiers assigned to the 'Outputs' group name identifier.

:SYST:INST:GRO:DEF 'Outputs','M1.DataOut1','M1.DataOut2'

The following SCPI command will set all voltage amplitudes in the 'Outputs' group to 0.05.

[:SOURce]:VOLTage[:AMPLitude]VOLT:AMPL 'Outputs',0.05

Measurement Name Identifiers

Measurement name identifiers are used to set up parameters associated with a specific measurement name. For example, the following shows the bit error ratio plugin SCPI command used to define a measurement name called 'MyMeasurement':

:PLUGin:ERATio:NEW 'MyMeasurement'

You can now set up parameters within the :PLUGin:ERATio subsystem to be associated with the 'MyMeasurement' identifier.

The following shows the accumulation duration mode (set to fixed time) and duration (set to 120 seconds) parameters associated with the 'MyMeasurement' identifier.

PLUGin:ERATio:ACQuisition:DURation 'MyMeasurement', FTIM

:PLUGin:ERATio:ACQuisition:TIME 'MyMeasurement',120

PLUGin Subsystem

The M8020A/M8030A platform supports a plugin interface. C# assemblies which implement certain interfaces are recognized by the software and integrated into the M8070A GUI and instrument software. Error ratio, Output Timing and Jitter Tolerance measurements are examples of the plugin concept. These C# assemblies are placed in a predefined location and are integrated into the M8070A software at start up time automatically.

These measurements can be controlled by the GUI and by SCPI commands. This means that the SCPI tree has to be defined by the plugin itself and is integrated into the existing SCPI tree of the M8020A/M8030A platform. SCPI commands for a plugin are placed below a predefined node named “:PLUGin” followed by the type of the plug in.

There is a set of “general” SCPI commands for creating, deleting and controlling a plugin. Also, there is a flexible set of SCPI commands provided by the plugin itself for setting and reading measurement parameters.

Additionally, a user can create multiple measurements of the same type. A measurement name identifier is used to address a created measurement.

2 Recommended Programming Techniques

KEYSIGHT TECHNOLOGIES

Output Protection

Pattern Generator Output Termination

The pattern generator's output ports must be terminated with 50 Ω if they are not connected. Termination of output ports improves the test performance.

NOTE

Refer also to the M8020A/M8030A Getting Started Guide for information on terminating the M8061A outputs to ensure proper performance.

Output Protection Algorithm

The instrument has an internal protection algorithm that protects the instrument from improper termination of the pattern generator's output ports.

The algorithm checks whether the termination voltage and termination resistance are correct before the output is enabled. This check happens once only.

In non-balanced dc coupled mode, the termination voltage must be within ± 0.1 V of the specified value. The resistance must be between 40 and 60 Ω.

In balanced mode, the resistance must be between 80 and 120 Ω (differential termination).

In ac coupled mode, which is intended for balanced only (for example, 8b/10b coded patterns), the output is enabled if no termination voltage and no termination resistance are detected. This is the case if ac capacitors are in front of the outputs. The check distinguishes between a terminated or an open output.

NOTE

Do not operate the output driver into an open.

Controlling the Output Levels

Controlling the Output Levels - Concepts

When the output levels are changed at the M8020A/M8030A data and clock output ports, four parameters are changed:

- V_{high}
- V_{low}
- V_{amptl}
- V_{offs}

The M8020A/M8030A groups these parameters into "pairs" (V_{ampl}/V_{offs}, V_{high}/V_{low}). If one of these values is modified, its "partner" remains constant and the values in the other pair are modified accordingly. For example, if V_{ampl} is changed, V_{offs} stays constant and V_{high} and V_{low} are modified accordingly.

Controlling the Output Levels - Procedures

Changing the Voltages with SCPI

The following commands show how you would set the data output so that it has an amplitude of 1 V and an offset of 0.5 V:

:SOUR:VOLT:AMPL ‘M1.DataOut1’,1; OFFS ‘M1.DataOut1’,0.5

This sets the output accordingly (V_{high} = 1 V, V_{low} = 0 V).

Conversely, you could set V_{high} and V_{low} directly:

:SOUR:VOLT:HIGH ‘M1.DataOut1’,1;

LOW ‘M1.DataOut1’,0

Allowing the M8020A/M8030A to Settle

Allowing the M8020A/M8030A to Settle - Concepts

When patterns are sent to the pattern generator or error detector, the M8020A/M8030A requires some time to settle. The following topics explain how the instruments react to pattern changes.

How Pattern Changes Affect the Pattern Generator

The M8020A/M8030A requires some time to change the patterns at the pattern generator and error detector. This is particularly true for large text-based (ASCII) patterns that have to be loaded from the file system.

In such a case, it is a recommended technique to always query the M8020A/M8030A's Operation Complete status after changing the pattern.

How Pattern Changes Affect the Error Detector

When the pattern changes, the error detector has to resynchronize to the new incoming signal. Depending on the signal, the alignment method used and the desired BER threshold, this procedure can take up to half a minute or more.

Checking Operation Status and Sync using SCPI Commands

The :STATus:OPERation:RUN? SCPI command returns the status of all pattern generators, error detectors and clock generator. Query the desired hardware component using the identifier (for example, ‘M1.DataOut1’).

The :STATus:INSTrument:SLOSs? SCPI command is a query that indicates the status of the error detector synchronization. This query returns either true (1) if all error detectors are synchronized, or false (0) if the error detectors are not synchronized.

Reading the M8020A/M8030A's Status

Reading the M8020A/M8030A's Status - Concepts

The M8020A/M8030A has a set of status registers that you can use to monitor the status of the hardware, software and any running tests.

Overview of Registers

Specifically, it has the following registers:

- Status Byte
  The Status Byte is a single register that stores the events occurring in the other registers.
- Standard Event Status Register
  The Standard Event Status Register monitors some non-critical errors and basic operations.
- Questionable Data Status Register
  The bits in the Questionable Data Status Register are set when certain events occur in the M8020A/M8030A that can lead to questionable results.
- Operation Status Register
  The Operation Status Register indicates when certain operations have been completed.

How the M8020A/M8030A Uses Status Registers

You can determine the state of certain instrument hardware and firmware events and conditions by programming the status register system.

The following subsections provide you with details about the M8020A/M8030A's status system.

Overview of the M8020A/M8030A's Status System

The M8020A/M8030A has status reporting features that give important information about events and conditions within the instrument. For example, a flag may be set to indicate the end of a measurement or perhaps a command error. To access this information, it is necessary to query a set of registers using SCPI.

M8020A/M8030A Status System Structure

The M8020A/M8030A's status system is comprised of multiple registers that are arranged in a hierarchical order. The lower-level status registers propagate their data to the higher-level registers in the data structures by means of summary bits. The Status Byte register is at the top of the hierarchy and contains general status information for the M8020A/M8030A's events and conditions. All other individual registers are used to determine the specific events or conditions.

Status Register Group Model

Figure 16 on page -36 illustrates the typical structure of a status register.

The M8020A/M8030A instrument status register model follows the structure described in IEEE 488.2, section 11.4.2.

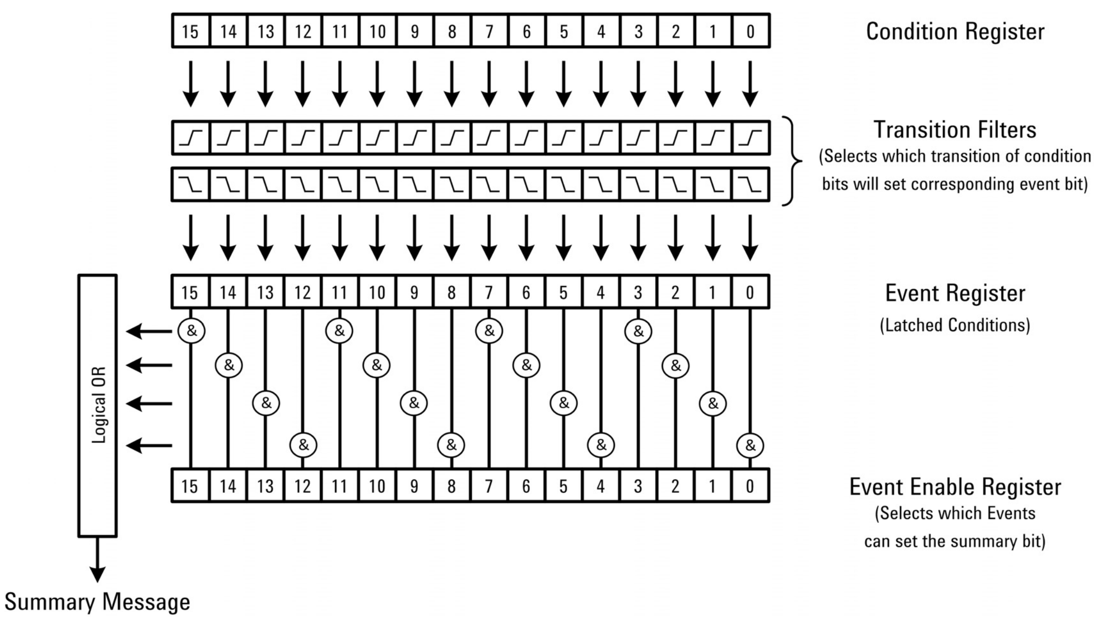


Figure 16 Status register structure

A condition is a device state which is either TRUE or FALSE. A condition register reflects these states in its condition bits.

A Condition Register is embedded in a register structure consisting of Transition Filters, an Event Register and an Event Enable Register.

This register is read by the CONDition? SCPI commands.

Transition Filters: A positive transition filter allows an event to be reported when a condition changes from false to true. A negative transition filter allows an event to be reported when a condition changes from true to false. Setting both positive and negative filters true allows an event to be reported anytime the condition changes. Clearing both filters disables event reporting.

This register is set and read by the NTRansition[?] and PTRansition[?] SCPI commands.

Event Register: An event register latches transition events from the condition register as specified by the positive and negative transition filters. Bits in the event register are latched and once set, they remain set until cleared by either querying the register contents or sending the *CLS command.

This register is read by the EVENt? SCPI commands.

Event Enable Register: An enable register specifies the bits in the event register that can generate a summary bit. Summary bits are, in turn, used by the next higher register.

This register is set and read by the ENABle[?] SCPI commands.

The registers work together as follows:

1. The *Condition Register* corresponds to a condition on the hardware or in the software. If the monitored condition is present, the corresponding bit is high.
2. The *Transition Registers* monitor changes in the Condition Register. If the Positive Transition Register is configured to watch for a condition, when this condition occurs and the bit in the Condition Register goes high, the Positive Transition Register passes this event to the *Event Register*.
3. If this bit is enabled in the *Event Enable Register*, a summary bit is generated in the next higher register. For the higher register, this is the Condition Register and the event is handled the same way as described here.

M8020A/M8030A Status Reporting Structure

Figure 17 on page -38 shows the M8020A/M8030A's status reporting structure.

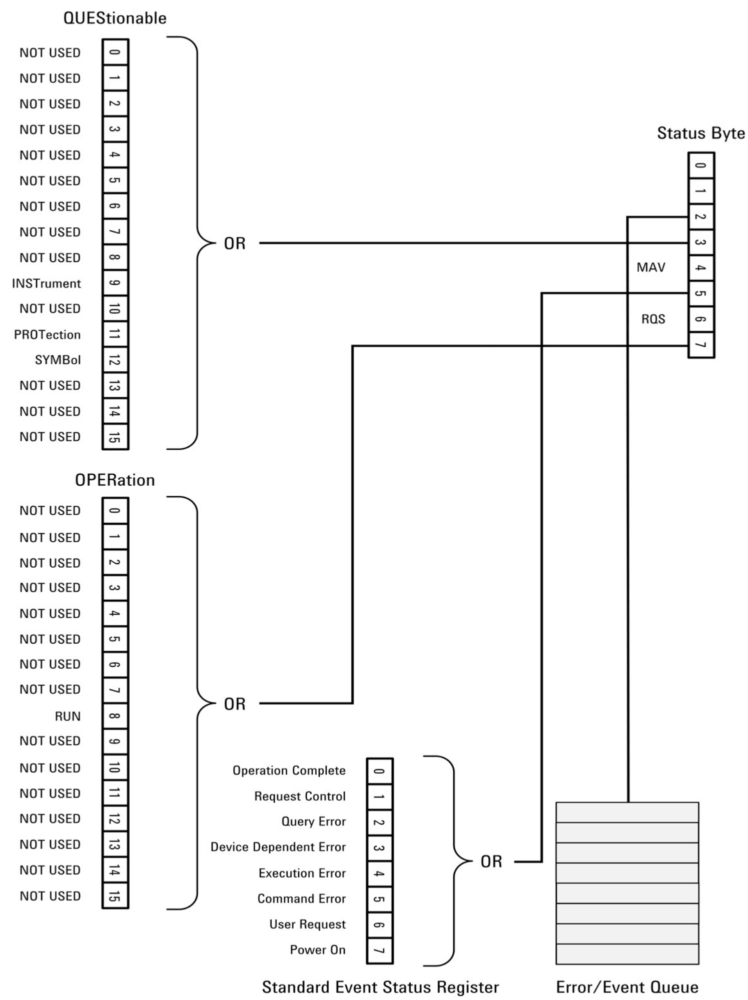


Figure 17 Status reporting structure

Status Byte

The Status Byte is the summary register to which the other registers report. Each reporting register is assigned a bit in the Status Byte Register.

The bits in the Status Byte byte have the following meaning:

Table 4 Status byte descriptions

Bit	Mnemonic	Description
0	Not used	
1	Not used	
2	EAV	Error available: the error queue contains at least one message.
3	QUES	A bit has been set in the Questionable Data Status register (indicates that a signal is of questionable quality).
4	MAV	Message available: There is at least one message in the message queue.
5	ESB	A bit in the Standard Event Register has been set.
6	RQS	ReQuest Service.
7	OPER	A bit in the Operation Status Register has been set.

Standard Event Status Register

The Standard Event Status register provides general-purpose information about the instrument. It sets bit 5 in the Status Byte.

Table 5 Standard event status register descriptions

Bit	Mnemonic	Description
0	Operation Complete	Operation Complete bit. It is set in response to the *OPC command, but only if the instrument has completed all its pending operations.
1	Request Control	Request Control bit.
2	Query Error	Query error bit. It indicates that there is a problem with the output data queue. There has been an attempt to read the queue when it is empty, the output data has been lost, or the query command has been interrupted.
3	Device Dependent Error	Device-dependent error bit. It is set when an instrument-specific error has occurred.
4	Execution Error	Execution error bit. It is set when a command (GPIB instrument specific) cannot be executed due to an out of range parameter or some instrument condition that prevents execution.
5	Command Error	Command error bit. It is set whenever the instrument detects an error in the format or content of the program message (usually a bad header, missing argument, or wrong data type etc.).
6	User Request	User Request bit.
7	Power On	Power-on bit. It is set each time the instrument is powered from off to on.

Questionable Status Register

The Questionable Status Register indicates that a currently running process is of questionable quality. The output of this register sets bit 3 of the Status Byte.

Table 6 Questionable status register descriptions

Bit	Mnemonic	Description
0 - 8	Not used	
9	INSTrument	This bit indicates the status of certain hardware components.
10	Not used	
11	PROTection	This bit indicates whether one or more of the instrument outputs detect an overload condition.
12	SYMBol	This bit indicates whether the error detector(s) has experienced a symbol alignment loss.
13 - 15	Not used	

Operation Status Register

The output of this register gives information about the current operation the instrument is performing. It sets bit 7 of the Status Byte.

Table 7 Operation status register descriptions

Bit	Mnemonic	Description
0 - 7	Not used	
8	RUN	
9 - 15	Not used	

Working with Patterns

Patterns can be accessed with a program and are stored in three different areas:

1. Local to current setting (“current:”)
2. Shared between settings (“shared:”)
3. Factory supplied standard patterns (“factory:”). These patterns are read only and cannot be modified.

Below these root nodes is a folder structure using ‘/’ as a separation character. So a complete pattern name might be something like “factory:SATA/LTDP-short”.

Patterns consist of a sequence of symbols. A symbol has a coding:

1. Binary (“Bit”)
2. 8b10b (“B8B10”)
3. 128/130 (“B128B130”)
4. 128/132 (“B128B132”)

A symbol can have additional attributes to modify behavior. All symbols support:

1. Mask: this attribute affects DataIn locations only. It specifies if the symbol is actually compared or masked (excluded from compare).
2. Squelch: this attribute affects DataOut locations only. If this attribute is 1, a squelch (out of band) level is used.

In addition, there are coding specific attributes, to control aspects of the coding such as bypassing or using a scrambler. Refer to :DATA:PATTern:IDATa[?] on page 425 for more information.

The :DATA:PATTern:USE SCPI command is used to specify the patterns and attributes. Refer to :DATA:PATTern:USE[?] on page 424 for details on how to specify a pattern and set up its parameters. The parameters include the pattern name, number of symbols, symbol coding (optional), mask (optional) and squelch (optional).

Symbol Coding Descriptions

Bit Coding

To encode a bit coded (binary) symbol 1, 2, or 3 bits are needed. This depends on the use of mask and squelch.

Table 8 Plain bit coding without using mask or squelch

Bit Offset/Range	Bit (range) Name	Description
0	Data	Data bit

Table 9 Mask is used

Bit Offset/Range	Bit (range) Name	Description
0	Data	Data bit
1	Mask	Mask (ignored on DataOut)

Table 10 Squelch is used

Bit Offset/Range	Bit (range) Name	Description
0	Data	Data bit
1	Squelch	Squelch (ignored on DataIn)

Table 11 Mask and squelch are used

Bit Offset/Range	Bit (range) Name	Description
0	Data	Data bit
1	Mask	Mask (ignored on DataOut)
2	Squelch	Squelch (ignored on DataIn)

Table 12 **Error is used**

Bit Offset/Range	Bit (range) Name	Description
0	Data	Data bit
1	Error	Error (ignored on DataOut, ignored on DataIn)

8b10b Coding

To encode an 8b10b symbol 16 bits (2 bytes) are used.

Table 13 8b10b coding

Bit Offset/Range	Bit (range) Name	Description
7:0	Data	Symbol data
8	K/D	0 = D-character, 1 = K-character
9		Reserved for future use. Must be set to 0
10	Mask	Mask (if present, ignored on DataOut)
11	Squelch	Squelch (if present, ignored on DataIn)
12	Enable scrambler	Enable scrambler (ignored on DataIn)
13	Pause scrambler	Pause scrambler (ignored on DataIn)
14	Reset scrambler	Reset scrambler (ignored on DataIn)
15	Start of Frame	Start of Frame (ignored on DataOut)

Table 14 Variant for captured pattern

Bit Offset/Range	Bit (range) Name	Description
7:0	Data	Symbol data
8	K/D	0 = D-character, 1 = K-character
9		Reserved for future use. Must be set to 0
10	Error	Error (ignored on DataOut, ignored on DataIn)
15:11	Squelch	Reserved for future use. Must be set to 0

128/130 Coding

To encode 128/130 symbol 144 bits (18 bytes) are used.

Table 15 128/130 coding

Bit Offset/Range	Bit (range) Name	Description
1:0	Framing	Framing bits
129:2	Data	Data
130	Mask	Mask (if present, ignored on DataOut)
131	Squelch	Squelch (if present, ignored on DataIn)
132	Reset Scrambler	Reset scrambler (ignored on DataIn)
133	Pause scrambler	Pause scrambler (ignored on DataIn)
134	Bypass scrambler	Scrambler bypass
135	Bypass byte 0 scrambler	Scrambler bypass byte 0
136	Do DC balancing	Do DC balancing (ignored on DataIn)
137	Reset DC balancing	Reset DC balancing (ignored on DataIn)
138	Send scrambler state	Send scrambler state
139	Reset Parity	Reset Parity (ignored on DataIn)
140	Pause Parity	Pause Parity (ignored on DataIn)
141	Start of Frame	Start of Frame (ignored on DataOut)
143:142		Reserved for future use. Must be set to 0

128/132 Coding

To encode a 128/132 symbol 144 bits (18 bytes) are used.

Table 16 128/132 coding

Bit Offset/Range	Bit (range) Name	Coding-128/132 content
3:0	Framing	Framing bits
131:4	Data	Data
132	Mask	Mask (if present, ignored on DataOut)
133	Squelch	Squelch (if present, ignored on DataIn)
134	Reset Scrambler	Scrambler reset (ignored on DataIn)
135	Pause Scrambler	Scrambler pause (ignored on DataIn)
136	Bypass Scrambler	Scrambler bypass
137	Bypass Byte 0 Scrambler	Scrambler bypass byte 0
138	Do DC Balancing	Do DC balancing (ignored on DataIn)
139	Reset DC Balancing	Rest DC balancing (ignored on DataIn)
140	Send Scrambler State	Send scrambler state (ignored on DataIn)
141	Reset Parity	Reset parity (ignored on DataIn)
142	Pause Parity	Pause parity (ignored on DataIn)
143	Start of Frame	Start of frame (ignored on DataOut)

Symbol Sequences

When combining symbols to a sequence of symbols they are filled in one after the other without any gaps. For the non-bit codings that use multiple bytes, the symbols naturally start again on a byte boundary. For the bit codings (binary) they start at the next available bit. Bits are used in a fashion that allows reading from “left to right”. This means the symbol bit 0 is encoded as bit 7 in a byte.

Byte	Bit							
	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
...	...							

1. Symbol
2. Symbol

8. Symbol

Figure 18 Example 1: plain bit coding

Byte	Bit							
	7	6	5	4	3	2	1	0
0	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0	1
2	0	1	0	1	0	1	0	1
3	0	1	0	1	0	1	0	1
...	...							

1. Symbol
2. Symbol

8. Symbol

Figure 19 Example 2: mask and squelch bit coding

Byte	Bit							
	7	6	5	4	3	2	1	0
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	0	1	2	3	4	5	6	7
3	8	9	10	11	12	13	14	15
...	...							

1. Symbol
2. Symbol

Figure 20 Example 3: 8b10b coding

Creating Pattern Sequences

A pattern sequence is configured using a language. The pattern sequence language is based on XML.

Sequence Editor

Pattern sequences are created and edited using the Sequence Editor in the M8070A software interface as shown in Figure 21 on page -49.

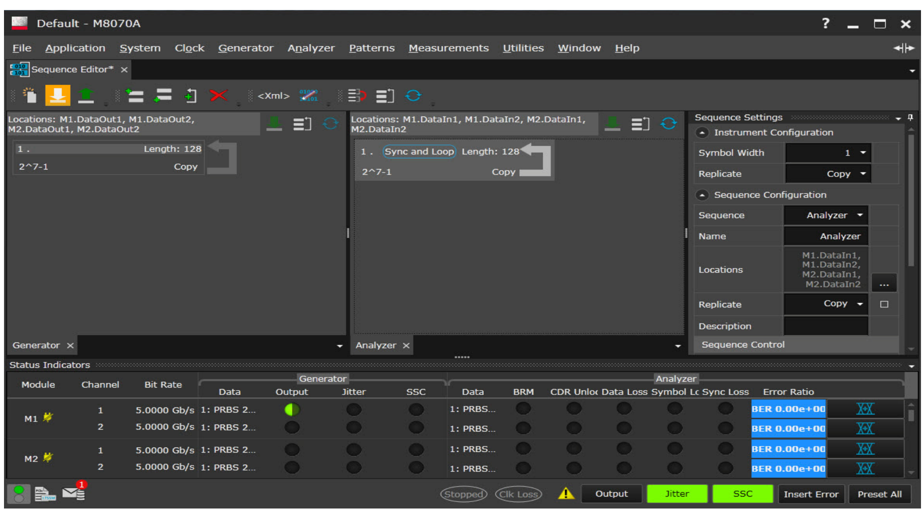


Figure 21 Sequence Editor in M8070A software interface

A default “Generator” and “Analyzer” sequence are configured in the Sequence Editor as shown in Figure 21 on page -49. You can edit these sequences or create new ones.

As a pattern sequence is being configured, the corresponding XML elements are generated and can be viewed in the XML editor by clicking on the <Xml> button shown in Figure 22 on page -50.

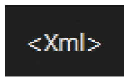


Figure 22 <Xml> button

Figure 23 on page -50 shows the XML elements for the default Generator and Analyzer. Clicking the UI button will return to the graphical interface.

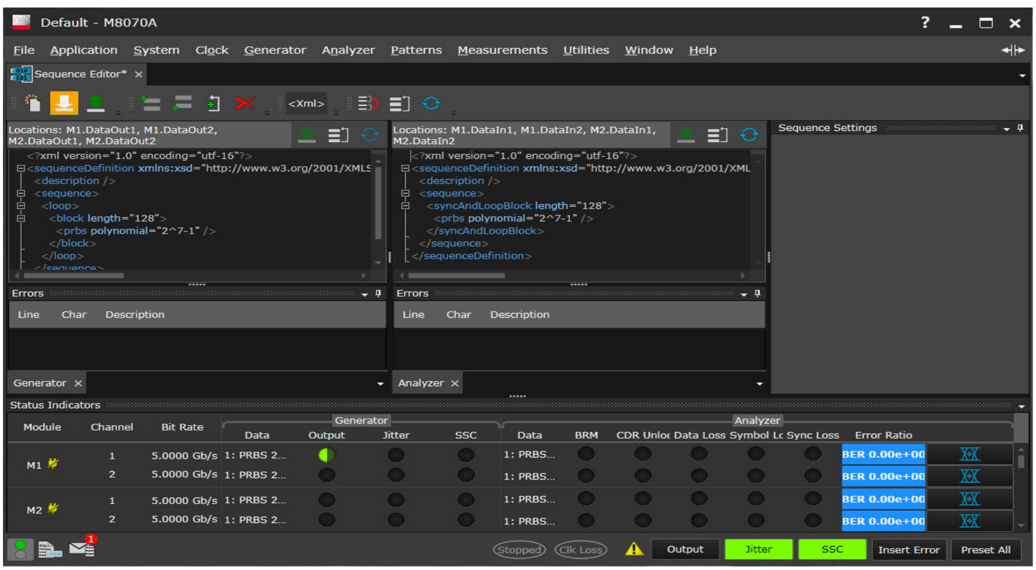


Figure 23 XML view

In UI view, click in a pattern sequence block to display its sequence settings as shown in Figure 24 on page -51.

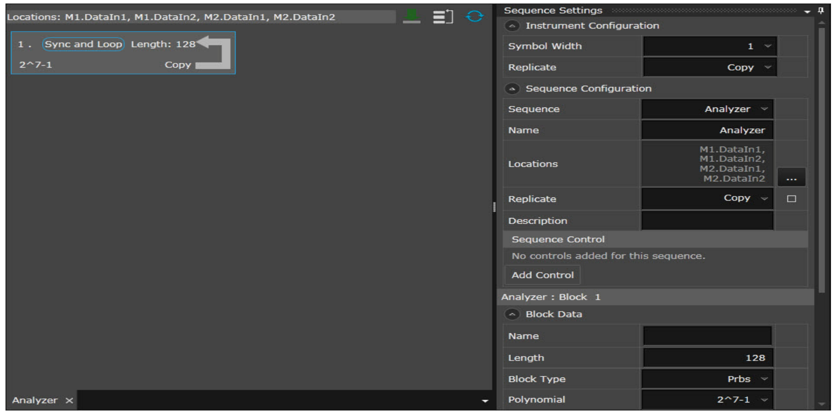


Figure 24 Sequence settings

Pattern Sequence Building Blocks

Minimal Sequence

A minimalistic sequence looks like the following:

```
<?xml version="1.0" encoding="utf-16"?>
<sequenceDefinition
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.agilent.com/schemas/M8000/DataSequence">
</sequenceDefinition>
```

This sequence generates a 0 signal to all assigned locations.

Description

To specify a description for the sequence, insert the following:

```
<description>Default 2^7-1 PRBS Sequence</description>
```

Version Number

To specify a version number for the sequence syntax, insert the following:

```
<version>1.0.0</version>
```

Symbol Width

To specify the symbol encoding this sequence works with, insert the following:

<symbolWidth>1</symbolWidth>

Available symbol widths are:

1: this is for bit coded symbols

10: this is for 8b/10b coded and bit coded symbols

130: for 128/130 coded and bit coded symbols

132: for 128/132 coded and bit coded symbols

Pattern Generation Granularity

Pattern generation has granularity. A single bit cannot be generated. Instead, multiple bits are generated at once. Compared to other BERTs like the J-BERT N4903B, different granularities are supported to allow for less restrictions in the pattern generation. Also, granularity is hidden as much as possible by unrolling patterns internally.

Table 17 Pattern generation granularity

Symbol Width	Granularity
1	64
10	80
130	130
132	132

The actual pattern sequence attributes are specified within:

<sequence replicate="Serialize"></sequence>

Replicate allows global selection of how to distribute PRBS or memory patterns to multiple locations.

- Serialize: first bit/symbol is assigned to first location, next bit/symbol to next location and so on. This means a pattern needs to have a minimum block length * locations bits. In the case of a PRBS, a PRWS is issued, setting the phases of the PRBS generators in a way that they will form a PRBS after a MUX.
- Copy: each location gets a copy of the PRBS /pattern.

Blocks and Loops

The basic building blocks of a sequence are blocks and loops.

A block is the basic unit of a sequence. Here patterns are assigned to locations or branches and are specified to other blocks. A block transitions to the next as a default action. But it also might be enclosed in a loop. Nested loops are also possible. There are three counted loop levels plus an infinite loop level:

- Loop level 1 can be used to repeat a block.
- Loop level 2 and 3 can span multiple blocks, but it must be ensured that either the start or end of a loop is on a fresh block.
- Loop level 4 is for infinite loops.

The following illustrates valid loop nesting with the full 4 loop levels:

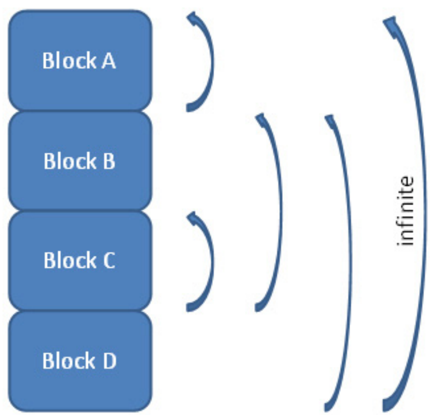


Figure 25 Loop nesting

Count attribute specifies the number of loop iterations. The valid value range is 1 to 2^{32}. If omitted, infinite is assumed. The enabled attribute allows the complete loop, including all loop content, to be temporarily disabled. If omitted, the default is enabled.

```
<loop count="1" enabled="1"></loop>
```

Length Attribute

The length attribute specifies the "length" of the block in number of bits. The enabled attribute allows the block to be temporarily disabled (ignored). If omitted, the default is enabled.

```
<block length="1" enabled="1"></block>
```

Synchronization and Loop Block

A special combination of sync and loop is available for achieving synchronization:

```
<syncAndLoopBlock length="100" enabled="1"></syncAndLoopBlock>
```

Other attributes for block or syncAndLoopBlock:

1 name: for naming a block.
2 cdr: for freezing CDR when CDR control is "sequence".
3 errorInsertion: for suppressing error insertion.
4 disparity: for setting the disparity in 8b10b coding.
5 startWordAlign: for initiating a word alignment in 8b10b and 128/130 coding. In the syncAndLoopBlock this flag is used for the initial block only, not for the looped block.

Pattern Selection

Within a block or syncAndLoopBlock, select the type of patterns to use. The choices are clock, pulse, Prbs, static and memory pattern. The XML elements for pattern selection are as follows:

```
<prbs polynomial="2^7-1" seed="127" invert="0" replicate="Serialize"
enabled="1"></prbs>

    <pattern source="factory:SATA/LTDP-short_b8b10"
replicate="Serialize" enabled="1"></pattern>
    <static value="0" compare="1"/>
    <pulse offset="0" width="10" compare="1" squelch="1"/>
    <clock divider="2" compare="1"/>

    <linkTraining direction="up"/>
```

Sequencer Triggers

Trigger output pulses can be generated and controlled from sequence blocks. These pulses are sent to the front panel TRIG OUT connectors.

This requires that the trigger mode be set to sequencer controlled. The following describes how to change the trigger mode.

1 In the M8070A software interface, select **System** > **Module View**.
2 Click on **Trig Out**.
3 Under Configuration parameters, change **Operating Mode** to **Sequencer Controlled**.

Refer to :OUTPut:TRIGger:MODE[?] on page 218 for information about setting the trigger mode using this SCPI command.

Two variants are available:

1 A trigger pulse can be generated relative to a sequence block.
2 A trigger pulse can be generated whenever a specific bit sequence is generated by a PRBS generator.

This is controlled via nodes within a block node.

NOTE If multiple sequences are used, only one sequence can actually have control of the trigger.

Triggering Relative to a Sequence Block

The following is an example of a trigger pulse relative to a sequencer block:

```
<trigger>
  <pulse offset="0" width="32"/>
</trigger>
```

The offset can be omitted and defaults to 0. The width can also be omitted and defaults to 32.

Triggering Based on Specific Bit Sequence

The following is an example of a trigger pulse generated based on a specific PRBS bit sequence:

```
<trigger>
  <pulseOnPrbsMatch width="32" matchPattern="0000000"
polynomial="2^7-1" seed="127" invert="0"/>
</trigger>
```

All attributes can be omitted. The width defaults to 32. The match pattern defaults to all 0's or all 1's depending on the invert attribute. The polynomial defaults to the polynomial used by the generators or analyzers. Invert also defaults to the PRBS invert setting of the generators or analyzers.

Block Branching

Within a block a conditional branch to another block can be done. The most simple form is as follows:

```
<if source="Break"</if>
```

This example simply jumps to the next block when a break command is done. The following describes how to set up a break command.

1 In the M8070A software interface, select **Patterns** > **Sequence Editor**.
2 Under the **Block Branches** settings, change **Condition** to **Break**.

Refer to :DATA:SEQuence:BREak on page 434 for information on setting the break command using this SCPI command.

When a branch to a different block is desired, use the “goto” attribute:

1 In the M8070A software interface, select **Patterns** > **Sequence Editor**.
2 Under the **Block Branches** settings, click in the **Go To Block** field and enter the block name.

Other potential conditions are SysInA, SysInB, CtrlInA, or CtrlInB. Break, SysInA and SysInB are global events and affect all locations of a sequence. CtrlInA and CtrlInB are local to a module and are most suitable when all locations of a sequence are using a single module.

It is also possible to specify a reaction based on a static level or a transition. Using the “event” attribute, you can select “low”, “high”, “positiveEdge” and “negativeEdge”.

1 In the M8070A software interface, select **Patterns** > **Sequence Editor**.
2 Under the **Block Branches** settings, change **Transition** to **Positive**, **Negative**, **Static High**, or **Static Low**.

Link Training Events

- **Link Training Target State**: indicates that the Link Training is in the target state (that is loopback)
  <if source="LinkTrainingTargetState"</if>
- **Link Training Detect State**: indicates that the Link Training is in the detect state (that is the reset state)
  <if source="LinkTrainingDetectState"</if>
- **Link Training Error**: indicates that an error occurred when trying to bring up the link (that is Link Training reached the target state but not the target speed)
  <if source="LinkTrainingError"</if>

Block Controls

Within a block it can be specified to generate a signal on CTRL OUT A, SYS OUT A, or SYS OUT B front panel connector. This can be useful for triggering an external device such as an oscilloscope or to generate auxiliary signals such as a laser enable signal. However, be aware that there is no specified timing relationship to the DataOut.

```
<control sink="CtrlOutA" sinkLocation="M1" value="pulse"/>
```

In a two module configuration, the sinkLocation attribute is used to specify on what module the CTRL OUT A signal is generated.

The shape of the signal is specified with the value attribute. It can be a static 0, static 1, or a pulse.

NOTE

SysOutA and SysOutB are internally controlled from the Trigger Sequencer.

Therefore, these signals can only be used in the same sequence where the trigger out is controlled.

NOTE CtrlOutA can only be used if it is not currently being used by the global block controls.

Global Block Controls

Global block controls are used to set up global routing of signals. Signals can be routed to the CTRL OUT A front panel connector only.

One common use could be for generating a pulse on CTRL OUT A whenever an error on the M1.DataIn1 or M1.DataIn2 of the corresponding module occurs:

```
<controls>
  <control sink="CtrlOutA" source="Error"
  sourceLocation="M1.DataIn1"/>
</controls>
```

With the additional “signalShape” attribute, it can be selected if a 0/1 pulse is generated whenever an error occurs.

Coding Configuration

Various aspects of coding can be configured. This is done with the <codingConfigurations> node. It is inserted just before the <symbolWidth> node. Within this node the 8b10b specific attributes are configured in a <b8b10> node. The align symbol is configured with an optional attribute. If not specified the default align symbol is K28.5. Also, you can specify a substitution symbol in an attribute that is inserted into the received data stream when an invalid code word is recognized. The default for this is K28.0. So the complete node would look like the following:

```
<b8b10 align="K28.5" substitution="K28.0">
```

And can be shortened to the following:

```
<b8b10>
```

Up to four filler primitives can be specified. Each filler primitive can consist of up to 4 symbols, where a symbol can be any valid 8b10b symbol or a ‘*’ that is used as a wildcard matching any symbol.

A filler symbol specification looks like the following:

```
<fillerPrimitive symbol1="K28.1" symbol2="K28.1"/>
```

After specifying a filler primative, various aspects of the (de)scrambler can be configured with the <scrambler> node. With the polynomial attribute the USB/PCIe or SATA polynomial can be selected. The reset value for the scrambler and the reset value for the scrambler after a hold sequence can be configured with the resetValue and resetValueAfterHold attribute. In addition, it can be specified if the scrambler is reset after a filler primitive was removed with the resetAfterFillerRemove attribute. All these nodes are optional, with defaults suitable for USB testing. If just <scrambler> node is given without attributes, it is equivalent to the following:

```
<scrambler polynomial="USB/PCIe" resetValue="65515"
resetValueAfterHold="6143" resetAfterFillerRemove="false">
```

Within the scrambler node, sequences of up to 4 symbols can be configured that control the scrambler when they are seen in the incoming data stream. With resetPrimitive the scrambler is reset, with holdPrimitive it is paused and with pauseStartPrimitive it is paused until a pauseEndPrimitive is seen.

The following are examples for the different standards.

USB 3.0

```
<codingConfigurations>
 <b8b10>
  <fillerPrimitive symbol1="K28.1" symbol2="K28.1"/>
  <scrambler>
    <resetPrimitive symbol1="K28.5"/>
  </scrambler>
 </b8b10>
</codingConfigurations>
```

PCIe 1/2

```
<codingConfigurations>
 <b8b10>
  <fillerPrimitive symbol1="K28.5" symbol2="K28.0"/>
  <fillerPrimitive symbol1="K28.0"/>
  <scrambler>
    <resetPrimitive symbol1="K28.5"/>
  </scrambler>
 </b8b10>
</codingConfigurations>
```

SATA

```
<codingConfigurations>
 <b8b10>
  <fillerPrimitive symbol1="K28.5" symbol2="D10.2" symbol3="D10.2"
  symbol4="D27.3"/>
  <fillerPrimitive symbol1="K28.0"/>
  <scrambler polynomial="SATA" resetValue="61686"
  resetValueAfterHold="30349">
   <resetPrimitive symbol1="K28.3" symbol2="D21.5" symbol3="D23.1"
   symbol4="D23.1"/>
   <holdPrimitive symbol1="K28.3" symbol2="*" symbol3="*"
   symbol4="*"/>
   <pauseStartPrimitive symbol1="K28.3" symbol2="D10.5"
   symbol3="D25.4" symbol4="D25.4"/>
   <pauseEndPrimitive symbol1="K28.3" symbol2="*" symbol3="*"
   symbol4="*"/>
  </scrambler>
 </b8b10>
</codingConfigurations>
```

MIPI Phy

```
<codingConfigurations>
 <b8b10>
   <fillerPrimitive symbol1="K28.1"/>
  </b8b10>
</codingConfigurations>
```

SAS

```
<codingConfigurations>
 <b8b10>
   <fillerPrimitive symbol1="K28.5" symbol2="D10.2" symbol3="D10.2"
symbol4="D27.3"/>
   <fillerPrimitive symbol1="K28.5" symbol2="D7.0" symbol3="D7.0"
symbol4="D7.0"/>
   <fillerPrimitive symbol1="K28.5" symbol2="D1.3" symbol3="D1.3"
symbol4="D1.3"/>
   <fillerPrimitive symbol1="K28.5" symbol2="D27.3" symbol3="D27.3"
symbol4="D27.3"/>
  </b8b10>
</codingConfigurations>
```

Link Training Configuration

Link Training Configuration describes the parameters controlling the LTSSM (Link Training and Status State Machine). In the first release it is only defined and implemented for PCI Express 3.0. This is done with the <linkTrainingConfigurations> node. It is inserted just before the <codingConfigurations> node. The following parameters can be configured as attributes.

Table 18 Link Training Configuration

Name	Possible Values	Default Value	Description	Used for DUT Type
DUT Type	AddInCard SystemBoard	AddInCard	Specifies which role the DUT should play during link training. It can either be an upstream device for testing a downstream port or vice versa.	Both
Link Equalization	Bypass PresetsOnly Full	Bypass	Determines whether link equalization should be performed. It can either be aborted after phase 1 (Bypass) or fully executed. In the second case it can be determined whether only preset or all (that is, individual cursor) requests should be accepted.	Both
Lane Number	0 - 15	0	This is the lane number being used.	Both
Link Number	0 - 255	0	This is the link number being used.	Add-in Card
Start Preset	P0 - P9	P4	This is the preset used by the BERT's TX port after switching to Gen 3 operation and when operating as an upstream device.	Add-in Card

Name	Possible Values	Default Value	Description	Used for DUT Type
DUT Preset Hint	-6 dB -7 dB -8 dB -9 dB -10 dB -11 dB -12 dB Reserved	Reserved	This is the preset hint being sent by the BERT to the DUT during phase 0 of the link equalization procedure. This is only used when the BERT operates as an upstream device.	Add-in Card
DUT Initial Preset	P0 - P9	P0	This is the preset the BERT transfers to the DUT in phase 0 of the link equalization procedure. This is only used when the BERT operates as an upstream device.	Add-in Card
DUT Target Preset	P0 - P9	P0	This is the preset the BERT requests the DUT to switch to during link equalization. Dependent on the role the BERT is playing, this is done in either phase 2 or 3 of the link equalization training.	Both
User Calibrated Presets	True False	False	BERT's TX port should use user-calibrated presets or standard presets during link training. If not specified standard presets are used.	Both

The following is an example of a Link Training Configuration for PCIe 3.0:

```
<linkTrainingConfigurations>
  <pcie3 dut="AddInCard" lane="0" link="0"
linkEQ="PresetsOnly" generatorStartPreset="P4"
dutRxPresetHint="Reserved" dutTxInitialPreset="P0"
dutTxTargetPreset="P7" calibratedPresets="true" />
</linkTrainingConfigurations>
```

Downloading Pattern Sequence to Hardware

Once a sequence has been configured, it is downloaded to the hardware using the download button in the Sequence Editor shown in Figure 26 on page -64. When the sequence has been downloaded, this button turns green.

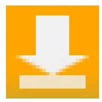


Figure 26 Download button

You can also download the pattern sequence configuration by copying and pasting the XML elements as a sequence string using the :DATA:SEQuence:VALue SCPI command. For an example, refer to :DATA:SEQuence:VALue[?] on page 433.

3 Programming Examples

Introduction

This chapter provides programming examples for setting up some of the most common functions including:

- Initialization
- SJ
- PJ
- Sampling point alignment
- BER

If you have programmed the N4903B, examples are provided for comparison to the M8020A. This will facilitate code conversion from the N4903B to the M8020A.

Initializing the Connection

N4903B Initialization

The ResourceName ("TCPIP1::10.0.0.207::inst0::INSTR") must be replaced by the instrument's address string from the VISA Assistant.

```
' First our declarations...
Private myN490X As AgilentN490x
Private myBERT As AgilentBert
Private myPG As AgilentBertLib.IAgilentBertPG
Private myPGClock As AgilentBertLib.IAgilentBertPGClock
Private myPGOut As AgilentBertLib.IAgilentBertPGOutput
Private myEDDataIn As AgilentBertLib.IAgilentBertEDDataIn
Private Sub Form_Load()
Set myN490X = New AgilentN490x
Set myBERT = myN490x.IAgilentBert
myBERT.Initialize ("TCPIP1::10.0.0.207::inst0::INSTR",
True, True,"QueryInstrStatus=true")
End Sub
Private Sub Form_Unload(Cancel As Integer)
myBERT.Close
End Sub
```

M8020A Initialization

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace baseInstrument.instruments
{
    class M8020A : BERT
    {
        private string m_PG_channel = "'M1.DataOut1'";
        private string m_ED_channel = "'M1.DataIn1'";
        private string m_system = "'M1.ClkGen'";
        double m_ED_clk = 16.0e9;
        double m_PG_clk = 16.0e9;
        double m_prevBER = 0.0;
        double m_prevBitCount = 0.0;
        double m_prevErrorCount = 0.0;
        double m_prevTimeStamp = 0.0;

        public M8020A(string resourceName,
                    string module = "M1",
                    string channel = "1") : base(resourceName)
        {
            if (module.IndexOf("MUX") == -1)
            {
                m_system = "'" + module + ".ClkGen'";
                m_PG_channel = "'" + module + ".DataOut" + channel + "'";
                m_ED_channel = "'" + module + ".DataIn" + channel + "'";
            }
            else
            {
                m_system = "'M1.ClkGen'";
                m_PG_channel = "'MUX'";
            }
        }
```

SJ Example

N4903B SJ Example

```
#region SJ
public override bool getSJEnabled()
// SOUR8:SIN?
{
    if ((int.Parse(this.Query(":SOUR8:SIN?"))) == 1)
        return true;
    else
        return false;
}

public override double getSJFreq()
// SOUR8:SIN:FREQ?
{
    return double.Parse(this.Query(":SOUR8:SIN:FREQ?"));
}

public override double getSJFreqMax()
// SOUR8:SIN:FREQ? MAX
{
    return double.Parse(this.Query(":SOUR8:SIN:FREQ? MAX"));
}

public override double getSJFreqMin()
// SOUR8:PER:FREQ? MIN
{
    return double.Parse(this.Query(":SOUR8:SIN:FREQ? MIN"));
}

public override List<string> setSJEnabled(bool b)
// SOUR8:SIN 1 ==> ON
// SOUR8:SIN 0 ==> OFF
{
    string strValue = "";

    if (b) strValue = "1";
    else strValue = "0";

    return this.Send(":SOUR8:SIN: " + strValue);
}
```

```
        public override List<string> setSJFreq(double SJFreq)
// SOUR8:SIN:FREQ
{
    string strValue = "";

    if ((this.getSJFreqMax() > SJFreq) && (this.getSJFreqMin() <= SJFreq))
        strValue = SJFreq.ToString();
    else
        strValue = this.getSJFreq().ToString();

    return this.Send(":SOUR8:SIN:FREQ " + strValue);
}

public override double getSJAmp()
// SOUR8:SIN:LEV?
{
    return double.Parse(this.Query(":SOUR8:SIN:LEV?"));
}

public override double getSJAmpMax()
// SOUR8:SIN:LEV? MAX
{
    return double.Parse(this.Query(":SOUR8:SIN:LEV? MAX"));
}

public override double getSJAmpMin()
// SOUR8:SIN:LEV? MIN
{
    return double.Parse(this.Query(":SOUR8:SIN:LEV? MIN"));
}

public override List<string> setSJAmp(double SJAmp)
// SOUR8:SIN:LEV
{
    string strValue = "";

    if ((this.getSJAmpMax() > SJAmp) && (this.getSJAmpMin() <= SJAmp))
        strValue = SJAmp.ToString();
    else
        strValue = this.getSJAmp().ToString();

    return this.Send(":SOUR8:SIN:LEV " + strValue);
}
#endregion
```

M8020A SJ Example

```
    #region SJ - implemented for M8020A
public override bool getSJEnabled()
// M8020A
{
  string str = "";
  str = this.Query(":SOUR:JITT:LFR:PER:STAT? " + m_PG_channel);

  if (str.IndexOf("1") != -1)
    return true;
  else
    return false;
}

public override double getSJFreq()
// M8020A
{
  string retString = "";
  double val = 0.0;

  retString = this.Query(":SOUR:JITT:LFR:PER:FREQ? " + m_PG_channel);
  val = double.Parse(retString);
  return val;
}

public override double getSJFreqMax()
// M8020A
{
  string retString = "";
  double val = 0.0;

  retString = this.Query(":SOUR:JITT:LFR:PER:FREQ? " + m_PG_channel + ",MAX");
  val = double.Parse(retString);
  return val;
}
```

```
        public override double getSJFreqMin()
// M8020A
{
    string retString = "";
    double val = 0.0;

    retString = this.Query(":SOUR:JITT:LFR:PER:FREQ? " + m_PG_channel + ",MIN");

    val = double.Parse(retString);

    return val;
}

public override List<string> setSJEnabled(bool b)
// M8020A
{
    string strValue = "";

    if (b) strValue = "ON";
    else strValue = "OFF";

    return this.Send(":SOUR:JITT:LFR:PER:STAT " + m_PG_channel + "," + strValue);
}

public override List<string> setSJFreq(double SJFreq)
// M8020A
{
    string strValue = "";

    SJFreq = Math.Round(SJFreq, 3);

    if ((this.getSJFreqMax() > SJFreq) && (this.getSJFreqMin() <= SJFreq))
        strValue = SJFreq.ToString();
    else if (this.getSJFreqMax() < SJFreq)
        strValue = this.getSJFreqMax().ToString();
    else if (this.getSJFreqMin() > SJFreq)
        strValue = this.getSJFreqMin().ToString();
    else
        strValue = this.getSJFreq().ToString();

    return this.Send(":SOUR:JITT:LFR:PER:FREQ " + m_PG_channel + "," + strValue +"Hz");
}
```

```
        public override double getSJAmp()
// M8020A
{
    string retString = "";
    string unit = "";
    double val = 0.0;

    unit = this.Query(":SOUR:JITT:LFR:UNIT? " + m_PG_channel);

    this.Send(":SOUR:JITT:LFR:UNIT " + m_PG_channel + ",UINT");

    retString = this.Query(":SOUR:JITT:LFR:PER:AMPL? " + m_PG_channel);

    this.Send(":SOUR:JITT:LFR:UNIT " + m_PG_channel + "," + unit);

    val = double.Parse(retString);

    return val;
}

public override double getSJAmpMax()
// M8020A
{
    string retString = "";
    string unit = "";
    double val = 0.0;

    unit = this.Query(":SOUR:JITT:LFR:UNIT? " + m_PG_channel);

    this.Send(":SOUR:JITT:LFR:UNIT " + m_PG_channel + ",UINT");

    retString = this.Query(":SOUR:JITT:LFR:PER:AMPL? " + m_PG_channel + ",MAX");

    this.Send(":SOUR:JITT:LFR:UNIT " + m_PG_channel + "," + unit);

    val = double.Parse(retString);

    return val;
}
```

```
        public override double getSJAmpMin()
// M8020A
{
    string retString = "";
    string unit = "";
    double val = 0.0;

    unit = this.Query(":SOUR:JITT:LFR:UNIT? " + m_PG_channel);

    this.Send(":SOUR:JITT:LFR:UNIT " + m_PG_channel + ",UINT");

    retString = this.Query(":SOUR:JITT:LFR:PER:AMPL? " + m_PG_channel + ",MIN");

    this.Send(":SOUR:JITT:LFR:UNIT " + m_PG_channel + "," + unit);

    val = double.Parse(retString);

    return val;
}

public override List<string> setSJAmp(double SJAmp)
// M8020A
{
    string strValue = "";
    string unit = "";
    double amp = 0.0;

    if (m_PG_channel == "'MUX'")
        SJAmp /= 2.0;
    if ((this.getSJAmpMax() > SJAmp) && (this.getSJAmpMin() <= SJAmp))
        amp = SJAmp;
    else
        amp = this.getSJAmp();

    amp = Math.Round(amp, 3);
    strValue = amp.ToString();

    unit = this.Query(":SOUR:JITT:LFR:UNIT? " + m_PG_channel);

    this.Send(":SOUR:JITT:LFR:UNIT " + m_PG_channel + ",UINT");

    this.Send(":SOUR:JITT:LFR:PER:AMPL " + m_PG_channel + "," + strValue);

    return this.Send(":SOUR:JITT:LFR:UNIT " + m_PG_channel + "," + unit);
}
#endregion
```

PJ Example

N4903B PJ Example

```
        #region PJ
public override bool getPJEnabled()
// SOUR8:PER?
{
    if ((int.Parse(this.Query(":SOUR8:PER?"))) == 1)
        return true;
    else
        return false;
}

public override double getPJFreq()
// SOUR8:PER:FREQ?
{
    return double.Parse(this.Query(":SOUR8:PER:FREQ?"));
}

public override double getPJFreqMax()
// SOUR8:PER:FREQ? MIN MAX
{
    return double.Parse(this.Query(":SOUR8:PER:FREQ? MAX"));
}

public override double getPJFreqMin()
// SOUR8:PER:FREQ? MIN MAX
{
    return double.Parse(this.Query(":SOUR8:PER:FREQ? MIN"));
}

public override List<string> setPJEnabled(bool b)
// SOUR8:PER 1 ==> ON
// SOUR8:PER 0 ==> OFF
{
    string strValue = "";

    if (b) strValue = "1";
    else strValue = "0";

    return this.Send(":SOUR8:PER: " + strValue);
}
```

```
        public override List<string> setPJFreq(double PJFreq)
// SOUR8:PER:FREQ
{
    string strValue = "";

    if ((this.getPJFreqMax() > PJFreq) && (this.getPJFreqMin() <= PJFreq))
        strValue = PJFreq.ToString();
    else if (this.getPJFreqMax() < PJFreq)
        strValue = this.getPJFreqMax().ToString();
    else if (this.getPJFreqMin() > PJFreq)
        strValue = this.getPJFreqMin().ToString();
    else
        strValue = this.getPJFreq().ToString();

    return this.Send(":SOUR8:PER:FREQ " + strValue);
}

public override double getPJAmp()
// SOUR8:PER:LEV?
{
    return double.Parse(this.Query(":SOUR8:PER:LEV?"));
}

public override double getPJAmpMax()
// SOUR8:PER:LEV? MAX
{
    return double.Parse(this.Query(":SOUR8:PER:LEV? MAX"));
}

public override double getPJAmpMin()
// SOUR8:PER:LEV? MIN
{
    return double.Parse(this.Query(":SOUR8:PER:LEV? MIN"));
}

public override List<string> setPJAmp(double PJAmp)
// SOUR8:PER:LEV
{
    string strValue = "";

    if ((this.getPJAmpMax() > PJAmp) && (this.getPJAmpMin() <= PJAmp))
        strValue = PJAmp.ToString();
    else
        strValue = this.getPJAmp().ToString();

    return this.Send(":SOUR8:PER:LEV " + strValue);
}
#endregion
```

M8020A PJ Example

```
        #region PJ - as PJ1 implemented for M8020A
public override bool getPJEnabled()
// M80820A - checked
{
    string str = "";

    str = this.Query(":SOUR:JITT:HFR:PER1:STAT? " + m_PG_channel);

    if (str.IndexOf("1") != -1)
        return true;
    else
        return false;
}

public override double getPJFreq()
// M8020A - checked
{
    string retString = "";
    double val = 0.0;

    retString = this.Query(":SOUR:JITT:HFR:PER1:FREQ? " + m_PG_channel);

    val = double.Parse(retString);

    return val;
}

public override double getPJFreqMax()
// M8020A - checked
{
    string retString = "";
    double val = 0.0;

    retString = this.Query(":SOUR:JITT:HFR:PER1:FREQ? " + m_PG_channel + ",MAX");

    val = double.Parse(retString);

    return val;
}
```

```
public override double getPJFreqMin()
// M8020A - checked
{
   string retString = "";
   double val = 0.0;

   retString = this.Query(":SOUR:JITT:HFR:PER1:FREQ? " + m_PG_channel + ",MIN");

   val = double.Parse(retString);

   return val;
}

public override List<string> setPJEnabled(bool b)
// M8020A - checked
{
   string strValue = "";

   if (b) strValue = "ON";
   else strValue = "OFF";

   return this.Send(":SOUR:JITT:HFR:PER1:STAT " + m_PG_channel + "," + strValue);
}

public override List<string> setPJFreq(double PJFreq)
// M8020A - checked
{
   string strValue = "";

   PJFreq = Math.Round(PJFreq, 3);

   if ((this.getPJFreqMax() > PJFreq) && (this.getPJFreqMin() <= PJFreq))
      strValue = PJFreq.ToString();
   else if (this.getPJFreqMax() < PJFreq)
      strValue = this.getPJFreqMax().ToString();
   else if (this.getPJFreqMin() > PJFreq)
      strValue = this.getPJFreqMin().ToString();
   else
      strValue = this.getPJFreq().ToString();

   return this.Send(":SOUR:JITT:HFR:PER1:FREQ " + m_PG_channel + "," + strValue + "Hz");
}
```

```
public override double getPJAmp()
// M8020A - checked
{
    string retString = "";
    string unit = "";
    double val = 0.0;
                unit = this.Query(":SOUR:JITT:HFR:UNIT? " + m_PG_channel);

    this.Send(":SOUR:JITT:HFR:UNIT " + m_PG_channel + ",UINT");

    retString = this.Query(":SOUR:JITT:HFR:PER1:AMPL? " + m_PG_channel);

    this.Send(":SOUR:JITT:HFR:UNIT " + m_PG_channel + "," + unit);

    val = double.Parse(retString);

    return val;
}
          public override double getPJAmpMax()
// M8020A - checked
{
    string retString = "";
    string unit = "";
    double val = 0.0;
                 unit = this.Query(":SOUR:JITT:HFR:UNIT? " + m_PG_channel);

    this.Send(":SOUR:JITT:HFR:UNIT " + m_PG_channel + ",UINT");

    retString = this.Query(":SOUR:JITT:HFR:PER1:AMPL? " + m_PG_channel + ",MAX");

    this.Send(":SOUR:JITT:HFR:UNIT " + m_PG_channel + "," + unit);

    val = double.Parse(retString);

    return val;
}
          public override double getPJAmpMin()
// M8020A - checked
{
    string retString = "";
    string unit = "";
    double val = 0.0;
                unit = this.Query(":SOUR:JITT:HFR:UNIT? " + m_PG_channel);

    this.Send(":SOUR:JITT:HFR:UNIT " + m_PG_channel + ",UINT");

    retString = this.Query(":SOUR:JITT:HFR:PER1:AMPL? " + m_PG_channel + ",MIN");

    this.Send(":SOUR:JITT:HFR:UNIT " + m_PG_channel + "," + unit);

    val = double.Parse(retString);

    return val;
}
```

```
        public override List<string> setPJAmp(double PJAmp)
// M8020A - checked
{
    string strValue = "";
    string unit = "";
    double amp = 0.0;

    if (m_PG_channel == "'MUX'")
        PJAmp /= 2.0;

    if ((this.getPJAmpMax() > PJAmp) && (this.getPJAmpMin() <= PJAmp))
        amp = PJAmp;
    else
        amp = this.getPJAmp();

    amp = Math.Round(amp, 3);
    strValue = amp.ToString();

    unit = this.Query(":SOUR:JITT:HFR:UNIT? " + m_PG_channel);

    this.Send(":SOUR:JITT:HFR:UNIT " + m_PG_channel + ",UINT");

    this.Send(":SOUR:JITT:HFR:PER1:AMPL " + m_PG_channel + "," + strValue);

    return this.Send(":SOUR:JITT:HFR:UNIT " + m_PG_channel + "," + unit);
}
#endregion
```

Sampling Point Alignment Example

N4903B Sampling Point Alignment Example

```
        public bool checkN900()
{
    return m_N4900;
}

# region Sampling Point Alignment

public override string autoAlign()
{
    string result = "";

    this.Send(":SENS1:EYE:ALIGN:AUTO 1");
    this.opc();
    do
    {
        result = this.Query(":SENS1:EYE:ALIGN:AUTO?");
    } while ((result != "CS_ABORTED") && (result != "CS_FAILED") && (result != "CS_SUCCESSFUL"));
    return result;
}

public override string dataCenter()
{
    string result = "";

    this.Send(":SENS1:EYE:TCEN 1");
    this.opc();
    do
    {
        result = this.Query(":SENS1:EYE:ALIGN:AUTO?");
    } while ((result != "CS_ABORTED") && (result != "CS_FAILED") && (result != "CS_SUCCESSFUL"));
    return result;
}

public override string thresholdCenter()
{
    string result = "";

    this.Send(":SENS1:EYE:ACEN 1");
    this.opc();
    do
    {
        result = this.Query(":SENS1:EYE:ALIGN:AUTO?");
    } while ((result == "CS_ABORTED") || (result == "CS_FAILED") || (result == "CS_SUCCESSFUL"));
    return result;
}
```

```
        public override List<string> sync()
{
    return this.Send(":SENS1:SYNC 1");
}

# endregion

#region sampling point
public override List<string> setSamplingPointDelay(double x)
{
    return this.Send(":INP1:DEL " + x.ToString());
}

public override double getSamplingPointDelay()
{
    string str = "";
    double delay = 0.0;

    str = this.Query(":INP1:DEL?");
    delay = double.Parse(str);
    return delay;
}

public override List<string> setSamplingPointDelayRel(double x)
{
    double current = 0.0;

    current = this.getSamplingPointDelay();
    x += current;

    return setSamplingPointDelay(x);
}
#endregion
```

M8020A Sampling Point Alignment Example

```
public override string autoAlign()
{
    string result = "";


    /*
     * do
    {
        result = this.Query(":STATus:OPERation:RUN:EVENt?");
    }
    while ((int.Parse(result) % 2) == 0);
     */

    this.Send(":INP:ALIG:EYE:AUTO " + m_ED_channel);
    this.opc();

    /*
    do
    {
        result = this.Query(":INP:ALIG:EYE:RES:DEL? " + m_ED_channel);
    } while (result == "");
    */
    return result;
}

public override string dataCenter()
{
    string result = "";

    this.Send(":INP:ALIG:EYE:TCEN " + m_ED_channel);
    this.opc();
    /*
    do
    {
        result = this.Query(":INP:ALIG:EYE:RES:DEL? " + m_ED_channel);
    } while (result == "");
    */
    return result;
}
```

```
public override string thresholdCenter()
{
    string result = "";

    this.Send(":INP:ALIG:EYE:ACEN " + m_ED_channel);
    this.opc();
    /*
    do
    {
        result = this.Query(":INP:ALIG:EYE:RES:THR? " + m_ED_channel);
    } while (result == "");
    */
    return result;
}

public override List<string> sync()
{
    return this.Send(":DATA:SYNC " + m_ED_channel);
}

# endregion

#region sampling point - implemented for M8020A
public override List<string> setSamplingPointDelay(double x)
{
    string delay = "";

    delay = (1.0e9 * x).ToString() + "e-9";

    return this.Send(":INP:DEL " + m_ED_channel + "," + x.ToString());
}

public override double getSamplingPointDelay()
{
    string retString = "";
    double val = 0.0;

    retString = this.Query(":INP:DEL? " + m_ED_channel);

    val = double.Parse(retString);

    return val;
}
#endregion
```

BER Example

N4903B BER Example

```
        # region ber

private double m_N4900_BitCount = 0.0;
private double m_N4900_ErrorCount = 0.0;

public override List<string> resetBERCounter()
{
    m_N4900_BitCount = 0.0;
    m_N4900_ErrorCount = 0.0;
    return this.startBERAcc();
}

public override List<string> setBERCounterResults()
{

    this.stopBERAcc();
    this.startBERAcc();

    double errors = 0.0;
    double bits = 0.0;

    errors = double.Parse(this.Query(":PFETCH:SENS2:ECO?"));
    bits = double.Parse(this.Query(":PFETCH:SENS2:BCO?"));
    m_ErrorCount = m_N4900_ErrorCount + errors;
    m_N4900_ErrorCount = m_ErrorCount;
    m_BitCount = m_N4900_BitCount + bits;
    m_N4900_BitCount = m_BitCount;
    m_BER = m_ErrorCount / m_BitCount;


    return this.ErrorQ();
}

public override List<string> startBERAcc()
{
    return this.Send(":SENS1:GATE ON");
}

public override List<string> stopBERAcc()
{
    return this.Send(":SENS1:GATE OFF");
}
        public override double getInstantaneousBER()
{
    return double.Parse(this.Query(":FETC:SENS2:ERAT?"));
}

# endregion
```

M8020A BER Example

```
private double m_prevAccBER_M8020 = 0.0;
private double m_prevAccBitCount_M8020 = 0.0;
private double m_prevAccErrorCount_M8020 = 0.0;

public override List<string> startBERAcc()
   // M8020A
{
   List<string> errorStr = new List<string>();

   errorStr = this.ErrorQ();

   return errorStr;
}

public override List<string> stopBERAcc()
   // M8020A - error counters can't be stopped yet
{
   List<string> errorStr = new List<string>();

   errorStr = this.ErrorQ();

   return errorStr;
}
        public override List<string> resetBERCounter()
{
   List<string> errorStr = new List<string>();
   List<double> bitResults = new List<double>();
   List<double> bitCounter = new List<double>();

   bitResults = this.getErrorCounter();
   errorStr = this.ErrorQ();

   m_prevAccBitCount_M8020 = bitResults[0];
   m_prevAccErrorCount_M8020 = bitResults[1];
   m_prevAccBER_M8020 = m_prevAccErrorCount_M8020 / m_prevAccBitCount_M8020;

   bitCounter = this.getErrorCounter();
   m_prevBitCount = bitCounter[0];
   m_prevErrorCount = bitCounter[1];
   m_prevTimeStamp = bitCounter[2];

   return errorStr;
}
```

```
public override double getInstantaneousBER()
    // M8020A
{
    double accumulatedBER = 0.0;
    List<double> result = new List<double>();
    result = this.getBERAccResults();
    accumulatedBER = result[2];
    return accumulatedBER;
}
        public override List<string> setBERCounterResults()
// M8020A
{
    List<double> berResults = new List<double>();
    List<double> bitCounter = new List<double>();
    double bitCount = 0.0;
    double errorCount = 0.0;
    double elapsedTime = 0.0;
    double timeStamp = 0.0;
    double ber = 0.5;

    bitCounter = this.getErrorCounter();

    bitCount = bitCounter[0] - m_prevBitCount;
    errorCount = bitCounter[1] - m_prevErrorCount;
    timeStamp = bitCounter[2];
    elapsedTime = 1000.0 * (timeStamp - m_prevTimeStamp);
    ber = errorCount / bitCount;

    m_BitCount = bitCount;
    m_ErrorCount = errorCount;
    m_BER = ber;

    return this.ErrorQ();
}
        private List<double> getErrorCounter()
// M8020A
{
    List<double> bitCounter = new List<double>();
    double bitCount = 0.0;
    double errorCount = 0.0;

    double timeStamp = 0.0;

    double counted1s = 0.0;
    double counted0s = 0.0;
    double erroneous1s = 0.0;
    double erroneous0s = 0.0;
    string identifier = "";
    string resultStr = "";
    string valStr = "";
    int index = 0;
```

```
        resultStr = this.Query(":FETC:BCO? " + m_ED_channel);

        // timestamp
        index = resultStr.IndexOf(",");
        valStr = resultStr.Substring(1, index - 1);
        timeStamp = double.Parse(valStr);
        resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

        // identifier
        index = resultStr.IndexOf(",");
        valStr = resultStr.Substring(0, index);
        identifier = valStr;
        resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

        // counted 1s
        index = resultStr.IndexOf(",");
        valStr = resultStr.Substring(0, index);
        counted1s = double.Parse(valStr);
        resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

        // counted 0s
        index = resultStr.IndexOf(",");
        valStr = resultStr.Substring(0, index - 1);
        counted0s = double.Parse(valStr);
        resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

        // erroneous 1s
        index = resultStr.IndexOf(",");
        valStr = resultStr.Substring(0, index - 1);
        erroneous1s = double.Parse(valStr);
        resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

        // erroneous 0s
        index = resultStr.IndexOf(")");
        valStr = resultStr.Substring(0, index - 1);
        erroneous0s = double.Parse(valStr);
        resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

        bitCount = counted0s + counted1s;
        //bitCount -= m_prevBitCount;
        errorCount = erroneous0s + erroneous1s;
        //errorCount -= m_prevErrorCount;

        bitCounter.Add(bitCount);
        bitCounter.Add(errorCount);
        bitCounter.Add(timeStamp);

        return bitCounter;
    }

    # endregion
```

NOTE

If the M8070A software is running on Windows 8 or 8.1, the VXI-11 protocol is not supported.

Example:

"SCPI Access (VXI-11): TCPIP0::localhost::inst0::INSTR"

does not work on Windows 8 or 8.1.

In this case, the HiSLIP protocol should be used as shown below:

"SCPI Access (HiSLIP): TCPIP0::localhost::hislip0::INSTR"

4 SCPI Command Language

SCPI Command Language - Introduction

The Serial BERT is compatible with the standard language for remote control of instruments. Standard Commands for Programmable Instruments (SCPI) is the universal programming language for instrument control.

SCPI can be subdivided into the following command sets:

- SCPI Common Commands
- SCPI Instrument Control Commands
- IEEE 488.2 Mandatory Commands

SCPI Common Commands

This is a common command set. It is compatible with IEEE 488.2 and contains general housekeeping commands. The common commands are always preceded by an asterisk. A typical example is the reset command:

*RST

The IEEE 488.2 command set also contains query commands. Query commands always end with a question mark.

SCPI Instrument Control Commands

The programming commands are compatible with the Standard Commands for Programmable Instruments (SCPI) standard. For more detailed information regarding the GPIB, the IEEE 488.2 standard, or the SCPI standard, refer to the following books:

- SCPI Consortium. SCPI- Standard Commands for Programmable Instruments, 1997 (http://www.scpiconsortium.org).
- International Institute of Electrical and Electronics Engineers. IEEE Standard 488.1- 1987, IEEE Standard Digital Interface for Programmable Instrumentation. New York, NY, 1987.
- International Institute of Electrical and Electronics Engineers. IEEE Standard 488.2- 1987, IEEE Standard Codes, Formats, Protocols and Common commands For Use with ANSI/IEEE Std 488.1- 1987. New York, NY, 1987.

IEEE 488.2 Mandatory Commands

In order to comply with the SCPI model as described in IEEE 488.2, the Serial BERT implements certain mandatory commands. Other commands are implemented optionally. For more detail on the IEEE 488.2 mandatory and optional commands, see IEEE Commands - Reference on page 103.

Overlapped and Sequential Commands

IEEE 488.2 defines the distinction between overlapped and sequential commands. A sequential command is one which finishes executing before the next command starts executing. An overlapped command is one which does not finish executing before the next command starts executing.

NOTE **It is not reliable to use wait statements in the control program to facilitate the use of overlapped commands.**

Because these commands may allow the execution of more than one command at a time, special programming techniques must be used to ensure valid results. The common commands *OPC, *WAI and *OPC? can be used for this purpose. They help synchronize a device controller with the execution of overlapped commands.

The behaviors of these commands, in brief, are as follows:

- *OPC
  The *OPC command sets the Operation Complete (OPC) bit of the Event Register when the No Operation Pending flag is TRUE (No Operation Pending flag is attached to each overlapped command). Until that time, the controller may continue to parse and execute previous commands. It is good technique, then, to periodically poll the OPC bit to determine if the overlapped command has completed.
- *WAI
  The *WAI command allows no further execution of commands or queries until the No Operation Pending flag is true, or receipt of a Device Clear (dcas) message, or a power on.
  The *WAI command can be used for overlapped commands. It stops the program execution until any pending overlapped commands have finished. Specifically, it waits until the No Operation Pending flag is TRUE, or receipt of a dcas message, or a power on.

- *OPC?
  The *OPC? query returns the ASCII character "1" in the Output Queue when the No Operation Pending flag is TRUE. At the same time, it also sets the Message Available (MAV) bit in the Status Byte Register. The *OPC? will not allow further execution of commands or queries until the No Operation Pending flag is true, or receipt of a Device Clear (dcas) message, or a power on.

NOTE **The command behaviors described above are for overlapped commands. When the same commands are used with sequential commands, the behaviors may be different.**

Data Types

The M8020A/M8030A has the capability of receiving and returning data in the following formats:

STRING

A string of human-readable ASCII characters, either quoted or nonquoted.

NUMERIC

The M8020A/M8030A handles the following numeric formats:

- <NR1>: Integer (0, 1, 2, - 1, etc.)
- <NR2>: Number with an embedded decimal point (0.1, 0.001. 3.3, etc.)
- <NR3>: Number with an embedded decimal point and exponent (1e33, 1.3e- 12, etc.)
- <NRf>: Represents <NR1>, <NR2> and <NR3>
- Binary preceded by #b (#B010101, #b011111, etc.)
- Octal preceded by #q (#Q777111, #q7331777, etc.)
- Hex preceded by #h (#haff, #h8989fffff, etc.)

BOOLEAN

Boolean values can be sent to the M8020A/M8030A as either

ON | OFF or 0 | 1. The M8020A/M8030A answers queries with 0 | 1.

Definite Length Arbitrary Block Data

Block data is used when a large quantity of related data is being returned. A definite length block is suitable for sending blocks of 8-bit binary information when the length is known beforehand. An indefinite length block is suitable for sending blocks of 8-bit binary information when the length is not known beforehand or when computing the length beforehand is undesirable.

It has the following format:

#<Length of length><Length of data><data>

<Length of length> is a single integer that contains the number of digits in <Length of data>, which in turn contains the length of the data. For example, a 512-byte pattern would be defined as:

#3512<data>

Important Points about SCPI

Important Points about SCPI - Concepts

There are a number of key areas to consider when using SCPI for the first time. These are as follows:

- Instrument Model
- Command Syntax
- Optional Parts of Commands
- Sending Commands
- Command Separators
- SCPI Command Structure

Instrument Model

SCPI guidelines require that the M8020A/M8030A is compatible with an instrument model. This ensures that when using SCPI, functional compatibility is achieved between instruments that perform the same tasks. For example, if two different instruments have a programmable clock frequency setting, then both instruments would use the same SCPI commands to set their frequency. The instrument model is made up of a number of subsystems.

The sub-system defines a group of functions within a module and has a unique identifier under SCPI, which is called the Root Keyword.

For more details on the instrument model, see STATus Subsystem on page 113.

Command Syntax

Commands may be up to twelve characters long. A short-form version is also available which has a preferred length of four characters or less. In this document the long-form and short-form versions are shown as a single word with the short-form being shown in upper-case letters.

For example, the long-form node command SOURce has the short-form SOUR. Using the short form saves time when entering a program; however, using the long form makes a program more descriptive and easier to understand.

SCPI commands may be commands only, commands and queries, or queries only. A question mark at the end of a command indicates that it is a query. If the question mark appears in brackets ([?]), the command has a command and query form.

Optional Command Keywords

Some layers in the SCPI command structure are optional. These optional keywords are indicated by square brackets ([]). A typical use for these types of keywords is with a command that is unique to one module. In this case, the top layer (Root Keyword) of the command structure may be omitted.

For example, the following command code segments are functionally identical:

[:SOURce]:JITTer[:GLOBal][:STATe] <ON|OFF|1|0>

:JITTer <ON|OFF|1|0>

:JITT <ON|OFF|1|0>

:jitt <ON|OFF|1|0>

Note that it is not necessary to include the syntax inside the square brackets ([]).

Query Responses

It is possible to interrogate the individual settings and status of a device using query commands. Retrieving data is a two-stage operation.

The query command is sent from the controller using the OUTPUT statement and the data is read from the device using the ENTER statement. A typical example is the SCPI IEEE 488.2 Common Command *IDN? which queries the identity of a device.

NOTE **When sending strings to the instrument, either the double quote (") or the single quote may be used ('), the latter being more suited to PASCAL programs, which make use of a single quote; the former being more suited to use in BASIC programs, which use a double quote as a delimiter.**

Command Separators

The SCPI command structure is hierarchical and is governed by commas, semicolons and colons:

- Commas are used to separate parameters in one command.
- Colons are used to separate levels.
- Semicolons are used to send more than one command to the instrument at a time.

It is possible to send several commands in one pass, as long as the commands all belong to the same node in the SCPI tree. The commands have to be separated by semicolons.

The following SCPI commands provide examples of this.

SOURce:VOLTage:OFFSet ‘M2.DataOut2’,-0.99

SOURce:VOLTage:AMPLitude ‘M2.DataOut2’,1.11

These commands can also be sent as follows:

VOLT:OFFS ‘M2.DataOut2’,-0.99; ‘M2.DataOut2’,AMPL 1.11

SCPI Command Structure Example

The SCPI command structure can be best examined by means of an example. For example, the command to set the pattern generator's output amplitude is:

[:SOURce]:VOLTage[:AMPLitude] ‘M1.DataOut1’,1.11

The structure of this command can be illustrated as follows:

[:SOURce]	This is the top layer of the command structure and identifies the source subsystem.
:VOLTage	This is the next layer and defines the subnode for setting a voltage level.
[:AMPLitude]	This is the command itself for setting the output amplitude level.
‘M1.DataOut1’,1.11	This specifies pattern generator 1, channel 1 and specifies an amplitude of 1.11.

NOTE

Any optional commands are enclosed in square brackets [ ] and any optional characters are shown in lower case.

A colon indicates a change of level in the command hierarchy. Commands at the same level in the hierarchy may be included in the same command line, if separated by a semi-colon.

The bar symbol (|) indicates mutually exclusive commands.

To translate this syntax into a command line, follow the convention described above. Remember, however, that the command line can be created in several different ways. It can be created with or without optional keywords and in a long or short form. The following example gives three possible forms of the command line; all are acceptable.

In long form:

:SOURce:VOLTage:AMPLitude ‘M1.DataOut1’,1.11

In short form:

:SOUR:VOLT:AMPL ‘M1.DataOut1’,1.11

With the optional commands removed:

:VOLT ‘M1.DataOut1’,1.11

The long form is the most descriptive form of programming commands in SCPI.

Sending Commands to the M8020A/M8030A

Sending Commands to the M8020A/M8030A - Concepts

A command is invalid and will be rejected if:

- It contains a syntax error.
- It cannot be identified.
- It has too few or too many parameters.
- A parameter is out of range.
- It is out of context.

Sending Commands using VISA

The following is a list of the available hardware interfaces for sending commands to the M8020A/M8030A firmware:

SCPI Access (HiSLIP): TCPIP0::localhost::hislip0::INSTR (High-Speed LAN Instrument Protocol)

SCPI Access (VXI-11): TCPIP0::localhost::inst0::INSTR (VXI-11 is a TCP/IP instrument protocol defined by the VXIbus Consortium)

NOTE

If you use the VXI-11 (TCP/IP instrument protocol) in your test programs, you must change the resource string to the HiSLIP protocol if the software is running on Windows 8 or Windows 8.1. VXI-11 is not supported at this time on Windows 8 or Windows 8.1.

For example:

“TCPIP0::192.17.34.0::inst0::INSTR” -> “TCPIP0::192.17.34.0::hislip0::INSTR”

SCPI Access (Socket): TCPIP0::localhost::5025::SOCKET (Standard SCPI-over-sockets port)

SCPI Access (Telnet): telnet localhost 5024 (Communication with LAN instrument through SCPI Telnet port)

Command Line Arguments

(See Communication on page 99 for details about /Socket, /Telnet, /Inst, /AutoID, /NoAutoID, /FallBack).

Table 19 Command line arguments

Argument	Description
/AutoId	Start in auto ID mode (this is the default) [optional]
/NoAutoId	Do not start in auto ID mode - use communication parameters from command line [optional]
/FallBack	Use auto ID mode if the communication parameters from the command line don't work [optional]
/Socket <socket port>	Set the socket port for the SCPI communication (only used with /NoAutoId) [optional]
/Telnet <telnet port>	Set the telnet port for the SCPI communication (only used with /NoAutoId) [optional]
/Inst <instrument number>	Set the instrument number for VXI-11.3 and HiSLIP SCPI communication [optional]
/rcl	Recall last used setting [optional]
/rst	Reset to factory default [optional]
/IgnoreAwg	M8070A software don't grab M8195A modules [optional]

Communication

Depending on the command line arguments /Socket, /Telnet, /Inst, /AutoID, /NoAutoID, /FallBack, the M8070A software starts several servers to handle SCPI commands. (Refer to the table above.)

/Socket, /Telnet, /Inst: If -1, don't start the respective servers

“Defaults:

Socket port: 5025 (e.g. TCPIP0::localhost::5025::SOCKET)

Telnet port: 5024

HiSLIP, VXI-11.3: 0 (e.g. TCPIP0::localhost::hislip0::INSTR, TCPIP0::localhost::inst0::INSTR)

/FallBack: If starting a server fails because of a conflict, try using another port or number

"HiSLIP, VXI-11.3: increase the index until a server can be started successfully

"Socket, Telnet: start with port 60000, then increase it until the servers can be started successfully. If neither socket nor telnet is disabled the M8070A software tries to start the servers on two consecutive ports

(socket port = telnet port + 1)

/AutoID: Automatically select ports and number for the connections, which are unique per instrument.

"This is the default behavior; it is not necessary to specify this argument on the command line.

"/Socket, /Telnet, /Inst are ignored (unless they are -1 and a server is disabled)

"If the M8070A software detects more than one AXIe module, use a special mechanism to obtain a number for the HiSLIP and VXI-11.3 servers, which makes sure that the M8070A software always uses the same VISA resource string per module

"The socket and telnet port are then calculated from the HiSLIP index:

telnet port = 60000 + 2 * <HiSLIP index>

socket port = 60000 + 2 * <HiSLIP index> + 1

Note: Ports may already be in use by Windows or other applications, so they are not available for M8195A.

/NoAutoID: Do not automatically select ports and number for the connections, use the values specified with /Socket, /Telnet, /Inst or their respective default values instead.

If both /NoAutoID and /AutoID are specified, /AutoID overrides /NoAutoID.

NOTE

The first port not assigned by IANA is 49152 (IANA, Internet Assigned Numbers Authority, http://www.iana.org).

5 SCPI Command Reference

Subsystems

The SCPI commands are divided into *subsystems*, which reflect the various functionality of the instrument. The following figure shows where the port-related subsystems are located.

Table 20 Subsystems

Subsystem	Description
STATus	The STATus subsystem reports the current system condition, when an event changes and sets transition filter states to report when a passing event occurs.
TRIGger	The TRIGger subsystem selects the following trigger modes: ▪ INTernal selects the 100 MHz reference clock of the AXIe chassis. ▪ REFerence selects a 10 MHz/100 MHz external reference clock. ▪ DIRect selects the clock in the M8041A and can be set manually from 8.1 GHz to 16.2 GHz as the system frequency. ▪ CMULtiplier selects a multiplied or divided clock to specify the system frequency.
SOURce	The SOURce subsystem controls output signals (for example, for setting frequency and levels).
OUTPut	The OUTPut subsystem controls the output ports of the pattern generator.
INPut	The INPut subsystem controls the input ports of the error detector.
FETCh	The FETCh subsystem is used to query the error detector's results.
MMEMory	The MMEMory subsystem is used to store and recall predefined or custom created settings and patterns.
PLUGin	The PLUGin subsystem is a plugin interface for integrating C# assemblies into the M8070A user interface and instrument software.
SYSTem	The SYSTem subsystem is used for general system functions.
DATA	The DATA subsystem is used to select patterns, define symbol parameters, define sequence blocks and loops and synchronize the pattern.

IEEE Commands – Reference

Mandatory Commands

The following mandatory IEEE 488.2 commands are implemented:

Table 21 IEEE commands

Name	Description under
*CLS	*CLS on page 104
*CAL?	*CAL? on page 104
*LRN?	*LRN? on page 104
*ESE[?]	*ESE[?] on page 104
*ESR?	*ESR? on page 104
*IDN?	*IDN? on page 105
*OPC	*OPC on page 105
*OPC?	*OPC? on page 105
*OPT?	*OPT? on page 106
*RCL	*RCL on page 106
*RST	*RST on page 106
*SAV	*SAV on page 106
*SRE[?]	*SRE[?] on page 107
*STB?	*STB? on page 107
*TST?	*TST? on page 107
*WAI	*WAI on page 108

*CLS

Syntax *CLS

Description This command clears all status register structures in a device. These registers include:

- OPERation Status Register structure
- QUEStionable Status Register structure

The corresponding enable registers are unaffected.

*CAL?

Syntax *CAL?

Description This command returns calibration data.

*LRN?

Syntax *LRN?

Description This command gets the device setup query. It return the instrument settings by binary block data.

*ESE[?]

Syntax *ESE <NRf>

Description The Standard Event Status Enable Command (*ESE) sets the Standard Event Enable Register. This register acts like a mask, so that the next time a selected bit goes high, the ESB bit in the status byte is set.

The query (*ESE?) returns the contents of the Standard Event Enable Register.

*ESR?

Syntax *ESR?

Description This query returns the Standard Event Status Register content. The register is cleared after it is read.

*IDN?

Syntax *IDN?

Description The IDeNtification query (*IDN?) response semantics are organized into four fields, separated by commas. The field definitions are as follows:

Table 22 IDN field definitions

Field	Value
Manufacturer	Agilent Technologies
Model	M80xx
Serial number	MYxxxxxxxx
Firmware level	X.x.x.xxx

*OPC

Syntax *OPC

Description The *OPC command sets the Operation Complete (OPC) bit of the Standard Event Status Register (ESR) when the No Operation Pending flag is TRUE (No Operation Pending flag is attached to each overlapped command). Until that time, the controller may continue to parse and execute previous commands. It is good technique, then, to periodically poll the OPC bit to determine if the overlapped command has completed.

*OPC?

Syntax *OPC?

Description The *OPC? query returns the ASCII character "1" in the Output Queue when the No Operation Pending flag is TRUE. At the same time, it also sets the Message Available (MAV) bit in the Status Byte Register. The *OPC? will not allow further execution of commands or queries until the No Operation Pending flag is true, or receipt of a Device Clear (dcas) message, or a power on.

*OPT?

Syntax: *OPT?

Description: The *OPT? query returns the installed options for each installed module. If no options are installed, only the model number(s) of the installed module(s) are returned.

Example: *OPT?
(M1-M8041A c08,c16)(M2-M8061A,004,008)

*RCL

Syntax: *RCL

Description: The *RCL command recalls a predefined read only instrument state setting. This command recalls the same device-specific functions effected by the *RST and *SAV commands.

The range is *RCL 1 to *RCL 10.

*RST

Syntax: *RST

Description: The Reset Command (*RST) sets the device-specific functions to a known state that is independent of the past-use history of the device.

*SAV

Syntax: *SAV

Description: The *SAV command saves a predefined read only instrument state setting. This command saves the same device-specific functions effected by the *RST and *RCL commands.

The range is *SAV 1 to *SAV 10.

*SRE[?]

Syntax *SRE[?] <NRf>

Description The Service Request Enable Command (*SRE) sets the Service Request Enable Register. This acts as a mask on the Status Byte, defining when the instrument can issue a service request. For a service request to be issued, the summary bit in the Status Byte must match the bit in the Service Request Enable Register. More than one bit may be set by the *SRE command.

The query returns the current contents of the Service Request Enable Register.

*STB?

Syntax *STB?

Description The Read Status Byte Query (*STB?) allows the programmer to read the status byte and Master Summary Status bit. When the status byte is read using the *STB command, bit 6 of the status byte is referred to as the Master Summary (MSS) bit. With this query, the status byte is not cleared when the value is read. It always reflects the current status of all the instrument's status registers.

*TST?

Syntax *TST?

Description The self-test query starts all internal self-tests and places a response into the output queue indicating whether or not the device completed the self-tests without any detected errors. It returns a 0 for success; a 1 if a failure was detected.

Upon successful completion of *TST?, the device settings are restored to their values prior to the *TST?

*WAI

Syntax *WAI

Description The *WAI commands allows no further execution of commands or queries until the No Operation Pending flag is true, or receipt of a Device Clear (dcas) message, or a power on.

The *WAI command can be used for overlapped commands. It stops the program execution until any pending overlapped commands have finished. Specifically, it waits until the No Operation Pending flag is TRUE, or receipt of a dcas message, or a power on.

SCPI Standard Commands

The following SCPI standard commands are implemented:

Table 23 SCPI standard commands

Name	Description under
:SYSTem:ERRor[:NEXT]?	:SYSTem:ERRor[:NEXT]? on page 109
:SYSTem:HELP:HEADers?	:SYSTem:HELP:HEADers? on page 109
:SYSTem:VERSion?	:SYSTem:VERSion? on page 109

:SYSTem:ERRor[:NEXT]?

Syntax :SYSTem:ERRor[:NEXT]?

Description Queries and at the same time deletes the oldest entry in the error queue.

:SYSTem:HELP:HEADers?

Syntax :SYSTem:HELP:HEADers?

Description This query returns all SCPI commands, queries and IEEE 488.2 common commands and queries currently implemented in the instrument.

:SYSTem:VERSion?

Syntax :SYSTem:VERSion?

Description This query returns a numeric value corresponding to the SCPI version number for which the instrument complies. The format is YYYY.V where Ys correspond to the year and the V corresponds to an approved version number for that year.

Miscellaneous Commands

The following miscellaneous commands are implemented:

Table 24 Miscellaneous commands

Name	Description under
:TEST:RESults?	:TEST:RESults? on page 110

:TEST:RESults?

Syntax :TEST:RESults?

.Description This command returns the results of self test.

Command Syntax to Find Min/Max Values

The following example illustrates the syntax to find min/max values for any system parameter:

In the below example, “Voltage Amplitude” is the parameter for which min/max values are to be found.

:SOURce:VOLTage:AMPLitude?

Syntax for Min/Max Value	:SOURce:VOLTage:AMPLitude? ‘identifier’, <MIN/MAX>
Example for Min Value	:SOURce:VOLTage:AMPLitude? ‘M1.DataOut’, MIN
Example for Max Value	:SOURce:VOLTage:AMPLitude? ‘M1.DataOut’, MAX

Location and Module Mapping - Reference

The following table lists locations and their respective modules.

Table 25 Location wise mapping of modules

Location	Module
Data Out	M8041A,M8051A, M8061A, M8062A and M8195A
Data In	M8041A,M8051A, M8061A and M8062A
System	M8041A
Clk Gen	M8041A, M8062A and M8195A
Ref Clk Out	M8041A
Clk Out	M8041A
Cln Clk Out	M8062A
Trig Out	M8041A
Sys Out A	M8041A
Sys Out B	M8041A
Ctrl Out A	M8041A and M8051A
Sys In A	M8041A
Sys In B	M8041A
Ctrl In A	M8041A and M8051A
Ctrl In B	M8041A and M8051A
Elect Idle In	M8061A and M8062A

NOTE The “System” locations are only available for few SCPIs. They are not visible on the GUI.

STATus Subsystem

This subsystem has the following SCPI structure:

```
:STATus
├:QUEStionable
│ └ ...
├:OPERation
│ └ ...
└:INSTrument
  └ ...
```

This subsystem has the following commands and subnodes:

Table 26

Name	Description under
Subnodes	
:QUEStionable	:STATus:QUEStionable Subnode on page 114
:OPERation	:STATus:OPERation Subnode on page 123
INSTrument	:STATus:INSTrument Subnode on page 128

:STATus:QUEStionable Subnode

This subnode has the following SCPI structure:

```
:STATus
└:QUEStionable
  ├:CONDition?
  ├:ENABle[?]
  ├[:EVENt]?
  ├:NTRansition[?]
  ├:PTRansition[?]
  ├:INSTrument[?]
  │ ├:CONDition?
  │ ├:ENABle[?]
  │ ├[:EVENt]?
  │ ├:NTRansition[?]
  │ └:PTRansition[?]
  ├:PROTection
  │ ├:CONDition?
  │ ├:ENABle[?]
  │ ├[:EVENt]?
  │ ├:NTRansition[?]
  │ └:PTRansition[?]
  └:SYMBol
    ├:CONDition?
    ├:ENABle[?]
    ├[:EVENt]?
    ├:NTRansition[?]
    └:PTRansition[?]
```

This subnode has the following commands:

Table 27

Name	Description under
:CONDition?	:STATus:QUEStionable:CONDition? on page 116
:ENABle[?]	:STATus:QUEStionable:ENABle[?] on page 116
[:EVENt]?	:STATus:QUEStionable[:EVENt]? on page 116
:NTRansition[?]	:STATus:QUEStionable:NTRansition[?] on page 116

Name	Description under
:PTRansition[?]	:STATus:QUEStionable:PTRansition[?] on page 117
:INSTrument[?]	:STATus:QUEStionable:INSTrument on page 117
:INSTrument:CONDition?	:STATus:QUEStionable:INSTrument:CONDition? on page 118
:INSTrument:ENABle[?]	:STATus:QUEStionable:INSTrument:ENABle[?] on page 118
:INSTrument[:EVENt]?	:STATus:QUEStionable:INSTrument[:EVENt]? on page 118
:INSTrument:NTRansition[?]	:STATus:QUEStionable:INSTrument:NTRansition [?] on page 118
:INSTrument:PTRansition[?]	:STATus:QUEStionable:INSTrument:PTRansition [?] on page 119
:PROTection	:STATus:QUEStionable:PROTection on page 119
:PROTection:CONDition?	:STATus:QUEStionable:PROTection:CONDition? on page 119
:PROTection:ENABle[?]	:STATus:QUEStionable:PROTection:ENABle[?] on page 119
:PROTection[:EVENt]?	:STATus:QUEStionable:PROTection[:EVENt]? on page 120
:PROTection:NTRansition[?]	:STATus:QUEStionable:PROTection:NTRansition [?] on page 120
:PROTection:PTRansition[?]	:STATus:QUEStionable:PROTection:PTRansition [?] on page 120
:SYMBol	:STATus:QUEStionable:SYMBol on page 120
:SYMBol:CONDition?	:STATus:QUEStionable:SYMBol:CONDition? on page 121
:SYMBol:ENABle[?]	:STATus:QUEStionable:SYMBol:ENABle[?] on page 121
:SYMBol[:EVENt]?	:STATus:QUEStionable:SYMBol[:EVENt]? on page 121
:SYMBol:NTRansition[?]	:STATus:QUEStionable:SYMBol:NTRansition[?] on page 121
:SYMBol:PTRansition[?]	:STATus:QUEStionable:SYMBol:PTRansition[?] on page 122

:STATus:QUEStionable:CONDition?

Syntax :STATus:QUEStionable:CONDition?

Description This command query returns the contents of the condition register of the Questionable Status Register structure.

:STATus:QUEStionable:ENABle[?]

Syntax :STATus:QUEStionable:ENABle <NRf>

:STATus:QUEStionable:ENABle?

Input Parameters <NRf>: Set enable mask.

Description This command sets the enable mask in the Questionable Status Register structure, which allows true conditions in the event register to be reported in the summary bit. The query returns the weighted value of the bits that are set in the enable register.

:STATus:QUEStionable[:EVENt]?

Syntax :STATus:QUEStionable:EVENt?

Description This command query returns the contents of the Questionable Status event register.

:STATus:QUEStionable:NTRansition[?]

Syntax :STATus:QUEStionable:NTRansition <NRf>

:STATus:QUEStionable:NTRansition?

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

:STATus:QUEStionable:PTRansition[?]

Syntax :STATus:QUEStionable:PTRansition <NRf>

:STATus:QUEStionable:PTRansition?

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Questionable Status Register structure. This is the default setting of the instrument. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.

:STATus:QUEStionable:INSTrument

The summary bit of this register set is reflected in bit 9 of the Questionable status register.

The :STATus:QUEStionable:INSTrument contains conditions which reflect states of the instrument's normal operation. These bits are summary bits. For example, if a Data Loss is detected by one or more error detectors, the Data Loss condition will be set.

The definition of each of these bits (condition register) is as follows:

Bit 0 – Error

One or more error detectors detect an error.

Bit 1 – Data Loss

This bit is set when the data source is turned off, not connected, or the cables or device is faulty. This bit can also be set when the 0/1 threshold is not within the eye limits of the incoming data signal. In this last case, use Auto Align or select Avg 0/1 Threshold.

Bit 2 – Sync Loss

This bit is set when the error detector pattern does not match the incoming data pattern or the BER of your device is higher than the sync threshold.

Bit 3 – Clock Loss

This bit is set when the pattern generator receives no external clock signal or the error detector receives no clock input signal.

Bit 4 – CDR Unlocked

One or more CDRs of the instrument are unlocked.

:STATus:QUEStionable:INSTrument:CONDition?

Syntax :STATus:QUEStionable:INSTrument:CONDition?

Description This command returns the contents of the condition register of the Questionable Status Register (bit 9), which reflects the state of the instrument.

:STATus:QUEStionable:INSTrument:ENABle[?]

Syntax :STATus:QUEStionable:INSTrument:ENABle <NRf>

:STATus:QUEStionable:INSTrument:ENABle?

Description This command sets the bits in the event enable register that can generate a summary bit used for the instrument state. The query returns the weighted value of the bits that are set in the event enable register.

:STATus:QUEStionable:INSTrument[:EVENt]?

Syntax :STATus:QUEStionable:INSTrument[:EVENt]?

Description This command queries the contents of the Questionable Status event register.

:STATus:QUEStionable:INSTrument:NTRansition[?]

Syntax :STATus:QUEStionable:INSTrument:NTRansition[?] <NRf>

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

:STATus:QUEStionable:INSTrument:PTRansition[?]

Syntax :STATus:QUEStionable:INSTrument:PTRansition[?] <NRf>

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.

:STATus:QUEStionable:PROTection

The summary bit of this register set is reflected in bit 11 of the Questionable Status Register.

The :STATus:QUEStionable:PROTection contains conditions which reflect states of the instrument's normal operation. These are summary bits. For example, if one or more outputs of the instrument detect an overload condition, the OverloadDetection bit will be set.

The definition of each of these bits (condition register) is as follows:

Bit 0 - OverloadDetection
Indicates that an overload condition has been detected.

:STATus:QUEStionable:PROTection:CONDition?

Syntax :STATus:QUEStionable:PROTection:CONDition?

Description This command returns the contents of the condition register of the Questionable Status Register (bit 11) to detect an overload condition.

:STATus:QUEStionable:PROTection:ENABle[?]

Syntax :STATus:QUEStionable:PROTection:ENABle <NRf>

:STATus:QUEStionable:PROTection:ENABle?

Description This command sets the bits in the event enable register that can generate a summary bit used for monitoring a symbol alignment loss. The query returns the weighted value of the bits that are set in the event enable register.

:STATus:QUEStionable:PROTection[:EVENt]?

Syntax :STATus:QUEStionable:PROTection[:EVENt]?

Description This command queries the contents of the Questionable Status event register.

:STATus:QUEStionable:PROTection:NTRansition[?]

Syntax :STATus:QUEStionable:PROTection:NTRansition <NRf>

:STATus:QUEStionable:PROTection:NTRansition?

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

:STATus:QUEStionable:PROTection:PTRansition[?]

Syntax :STATus:QUEStionable:PROTection:PTRansition <NRf>

:STATus:QUEStionable:PROTection:PTRansition?

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.

:STATus:QUEStionable:SYMBol

The summary bit of this register set is reflected in bit 12 of the Questionable Status Register.

The :STATus:QUEStionable:SYMBol contains conditions which reflect states of the instrument’s normal operation. These are summary bits. For example, if an error detector(s) of the instrument detects a symbol alignment loss, the SymbolAlignmentLoss bit will be set.

The definition of each of these bits (condition register) is as follows:

Bit 0 - SymbolAlignmentLoss
Indicates whether the error detector(s) has experienced a symbol alignment loss.

:STATus:QUEStionable:SYMBol:CONDition?

Syntax :STATus:QUEStionable:SYMBol:CONDition?

Description This query returns the contents of the condition register of the Questionable Status Register (bit 12) to detect a symbol alignment loss.

:STATus:QUEStionable:SYMBol:ENABle[?]

Syntax :STATus:QUEStionable:SYMBol:ENABle <NRf>

:STATus:QUEStionable:SYMBol:ENABle?

Description This command sets the bits in the event enable register that can generate a summary bit used for monitoring a symbol alignment loss. The query returns the weighted value of the bits that are set in the event enable register.

:STATus:QUEStionable:SYMBol[:EVENt]?

Syntax :STATus:QUEStionable:SYMBol[:EVENt]?

Description This command queries the contents of the Questionable Status event register.

:STATus:QUEStionable:SYMBol:NTRansition[?]

Syntax :STATus:QUEStionable:SYMBol:NTRansition[?] <NRf>

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

:STATus:QUEStionable:SYMBol:PTRansition[?]

Syntax :STATus:QUEStionable:SYMBol:PTRansition <NRf>

:STATus:QUEStionable:SYMBol:PTRansition?

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.

:STATus:OPERation Subnode

This subnode has the following SCPI structure:

```
:STATus
└:OPERation
  ├:CONDition?
  ├:ENABle[?]
  ├[:EVEN]?
  ├:NTRansition[?]
  ├:PTRansition[?]
  └:RUN
    ├:CONDition?
    ├:ENABle[?]
    ├[:EVEN]?
    ├:NTRansition[?]
    └:PTRansition[?]
```

Table 28

Name	Description under
:CONDition?	:STATus:OPERation:CONDition? on page 124
:ENABle[?]	:STATus:OPERation:ENABle[?] on page 124
[:EVENt]?	:STATus:OPERation[:EVENt]? on page 124
:NTRansition[?]	:STATus:OPERation:NTRansition[?] on page 124
:PTRansition[?]	:STATus:OPERation:PTRansition[?] on page 125
:RUN	:STATus:OPERation:RUN on page 125
:RUN:CONDition?	:STATus:OPERation:RUN:CONDition? on page 126
:RUN:ENABle[?]	:STATus:OPERation:RUN:ENABle[?] on page 126
:RUN[:EVENt]?	:STATus:OPERation:RUN[:EVENt]? on page 126
:RUN:NTRansition[?]	:STATus:OPERation:RUN:NTRansition[?] on page 126
:RUN:PTRansition[?]	:STATus:OPERation:RUN:PTRansition[?] on page 127

:STATus:OPERation:CONDition?

Syntax :STATus:OPERation:CONDition?

Description This query returns the contents of the condition register of the Operation Status Register structure.

:STATus:OPERation:ENABle[?]

Syntax :STATus:OPERation:ENABle <NRf>

:STATus:OPERation:ENABle?

Input Parameters <NRf>: Set enable mask.

Description This command sets the enable mask in the Operation Status Register structure, which allows true conditions in the event register to be reported in the summary bit. The query returns the weighted value of the bits that are set in the enable register.

:STATus:OPERation[:EVENt]?

Syntax :STATus:OPERation[:EVENt]?

Description This query returns the content of Operation Status Event Register.

:STATus:OPERation:NTRansition[?]

Syntax :STATus:OPERation:NTRansition <NRf>

:STATus:OPERation:NTRansition?

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Operation Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

:STATus:OPERation:PTRansition[?]

Syntax	:STATus:OPERation:PTRansition <NRf>
	:STATus:OPERation:PTRansition?
Input Parameters	<NRf>: Set transition filter state.
Description	This command sets the transition filter state in the Operation Status Register structure. This is the default setting of the instrument. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.

:STATus:OPERation:RUN

The summary bit of this register set is reflected in bit 8 of the Operation Status Register.

The :STATus:OPERation:RUN contains conditions which reflect states of the instrument's normal operation. These are summary bits. For example, if the pattern generators, error detectors and clock generator are functioning normally, the Running bit will be set.

The definition of each of these bits (condition register) is as follows:

Bit 0 - Running
Indicates that the instrument is in normal operation mode; generators are working, all error detectors are running and the clock generator is functioning properly.

:STATus:OPERation:RUN:CONDition?

Syntax :STATus:OPERation:RUN:CONDition?

Description This command returns the contents of the condition register of the Questionable Status Register (bit 8) to detect if all components are operating normally.

:STATus:OPERation:RUN:ENABle[?]

Syntax :STATus:OPERation:RUN:ENABle <NRf>

:STATus:OPERation:RUN:ENABle?

Description This command sets the bits in the event enable register that can generate a summary bit used to determine if the hardware is functioning properly. The query returns the weighted value of the bits that are set in the event enable register.

:STATus:OPERation:RUN[:EVENt]?

Syntax :STATus:OPERation:RUN[:EVENt]?

Description This query returns the contents of the Operation Status event register.

:STATus:OPERation:RUN:NTRansition[?]

Syntax :STATus:OPERation:RUN:NTRansition <NRf>

:STATus:OPERation:RUN:NTRansition?

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

:STATus:OPERation:RUN:PTRansition[?]

Syntax :STATus:OPERation:RUN:PTRansition <NRf>

:STATus:OPERation:RUN:PTRansition?

Input Parameters <NRf>: Set transition filter state.

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.

:STATus:INSTrument Subnode

The :STATus:INSTrument commands return the 'current' value/condition of several predefined device states. These commands are a set of SCPI queries which return true (1) or false (0).

This subnode has the following SCPI structure:

```
:STATus
└─:INSTrument
  ├─:CLOSe?
  ├─:DLOSs?
  ├─:SLOSs?
  ├─:SALoss?
  ├─:ERRor?
  ├─:OVERload?
  ├─:RCINput?
  ├─:RUN?
  └─:CDR
    └─:ULOCKed?
```

This subnode has the following commands:

Table 29

Name	Description under
:CLOSs?	:STATus:INSTrument:CLOSs? on page 129
:DLOSs?	:STATus:INSTrument:DLOSs? on page 129
:SLOSs?	:STATus:INSTrument:SLOSs? on page 129
:SALoss?	:STATus:INSTrument:SALoss? on page 130
:ERRor?	:STATus:INSTrument:ERRor? on page 130
:OVERload?	:STATus:INSTrument:OVERload? on page 130
:RCINput?	:STATus:INSTrument:RCINput? on page 131
:RUN?	:STATus:INSTrument:RUN? on page 131
:CDR:ULOCked?	:STATus:INSTrument:CDR:ULOCked? on page 131

:STATus:INSTrument:CLOSs?

Syntax	:STATus:INSTrument:CLOSs? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' or 'M*.DataOut1' or 'M*.DataOut2' or ‘M1.System’
Return Range	0|1
Description	This query indicates if the clock has been lost. A true (1) indicates a clock loss.
Example	:STAT:INST:CLOS? ‘M1.DataIn1’

:STATus:INSTrument:DLOSs?

Syntax	:STATus:INSTrument:DLOSs? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' or ‘M1.System’
Return Range	0|1
Description	This query indicates if the data has been lost. A true (1) indicates data loss.
Example	:STAT:INST:DLOS? ‘M1.DataIn1’

:STATus:INSTrument:SLOSs?

Syntax	:STATus:INSTrument:SLOSs? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' or ‘M1.System’ Return Range 0|1
Description	This query indicates if the sync has been lost. A true (1) indicates a sync loss.
Example	:STAT:INST:SLOS? ‘M1.DataIn1’

:STATus:INSTrument:SALoss?

Syntax	:STATus:INSTrument:SALoss? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' or ‘M1.System’
Return Range	0|1
Description	This query indicates if the symbol alignment has been lost. A true (1) indicates a symbol alignment loss.
Example	:STAT:INST:SAL? ‘M1.DataIn1’

:STATus:INSTrument:ERRor?

Syntax	:STATus:INSTrument:ERRor? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' or ‘M1.System’
Return Range	0|1
Description	This query indicates if an error condition has occurred. A true (1) indicates an error condition.
Example	:STAT:INST:ERR? ‘M1.DataIn1’

:STATus:INSTrument:OVERload?

Syntax	:STATus:INSTrument:OVERload? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' or 'M*.DataOut1' or 'M*.DataOut2' or ‘M1.System’
Return Range	0|1
Description	This query indicates an overload condition exists at an input or output. A true (1) indicates an overload condition.
Example	:STAT:INST:OVER? ‘M1.DataIn1’

:STATus:INSTrument:RCINput?

Syntax	:STATus:INSTrument:RCINput? ‘identifier’
Input Parameters	‘identifier’: ‘M1.ClkGen’
Return Range	0|1
Description	This query indicates the input state of the reference clock.
Example	:STAT:INST:RCIN? ‘M1.ClkGen’

:STATus:INSTrument:RUN?

Syntax	:STATus:INSTrument:RUN? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' or ‘M1.System’
Return Range	0|1
Description	This query indicates if the input is operating properly. A true (1) indicates the input is operating properly.
Example	:STAT:INST:RUN? ‘M1.DataIn1’

:STATus:INSTrument:CDR:ULOCked?

Syntax	:STATus:INSTrument:CDR:ULOCked? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' or ‘M1.System’
Return Range	0|1
Description	This query indicates the lock state of the clock data recovery. A true (1) indicates an unlocked state.
Example	:STAT:INST:CDR:UNLOC? ‘M1.DataIn1’

TRIGger Subsystem

The TRIGger subsystem selects the trigger mode at the TRIG OUT port.

This subsystem has the following SCPI structure:

```
:TRIGger
├[:SOURce][?]
│ └:FREQuency?
│
├:INTernal
│ └[:SOURce][?]
│
├:REFerence
│ └[:FREQuency][?]
│
├:DIRect
│ └[:FREQuency][?]
│   └:DETect
│
└:CMULtiplier
  ├:LBWidth[?]
  └:FREQuency[?]
    ├:DETect
    ├:MULTiplier[?]
    └:DIVider[?]
```

Table 30

Name	Description under
[:SOURce]	:TRIGger[:SOURce][?] on page 133
[:SOURce]:FREQuency?	:TRIGger[:SOURce]:FREQuency? on page 134
:INTernal[:SOURce][?]	:TRIGger:INTernal[:SOURce][?] on page 134
:REFerence[:FREQuency][?]	:TRIGger:REFerence[:FREQuency][?] on page 135
:DIRect[:FREQuency][?]	:TRIGger:DIRect[:FREQuency][?] on page 135
:DIRect[:FREQuency]:DETect	:TRIGger:DIRect[:FREQuency][?] on page 135
:CMULtiplier:LBWidth[?]	:TRIGger:CMULtiplier:LBWidth[?] on page 136

Name	Description under
:CMULtiplier:FREQuency[?]	:TRIGger:CMULtiplier:FREQuency:DETect on page 138
:CMULtiplier:FREQuency:DETect	:TRIGger:CMULtiplier:FREQuency:DETect on page 138
:CMULtiplier:FREQuency:MULTipli er[?]	:TRIGger:CMULtiplier:FREQuency:MULTipli er[?] on page 136
:CMULtiplier:FREQuency:DIVider[?]	:TRIGger:CMULtiplier:FREQuency:DIVider[?] on page 137
:CMULtiplier:FREQuency[?]	:TRIGger:CMULtiplier:FREQuency[?] on page 137
:CMULtiplier:FREQuency:DETect	:TRIGger:CMULtiplier:FREQuency:DETect on page 138

:TRIGger[:SOURce][?]

Syntax :TRIGger[:SOURce] ‘identifier’, INTernal|REFerence|DIRect|CMULtiplier

:TRIGger[:SOURce]? ‘identifier’

Input Parameters ‘identifier’: M1.ClkGen

Return Range INT|REF|DIR|CMUL

Description Selects the different trigger modes. The query returns the current trigger source mode.

The trigger source modes include the following:

- INTernal 100 MHz reference clock or AXIe 100 MHz (M9505A AXIe Chassis)
- REFerence 10 MHz / 100 MHz external reference clock
- DIRect 8.1 GHz to 16.207 GHz clock used as the system frequency directly in the M8041A
- CMULtiplier Multiplied/divided clock used to specify system frequency

This SCPI is only for M8041A and M8062A.

Example :TRIGger:SOURce ‘M1.ClkGen’, INT

:TRIGger:SOURce? ‘M1.ClkGen’

:TRIGger[:SOURce]:FREQuency?

Syntax	:TRIGger[:SOURce]:FREQuency? ‘identifier’
Input Parameters	‘identifier’: ‘M1.TrigOut’
Description	This query returns the effective frequency of the trigger out clock signal. This parameter is only active if the subrate clock operating mode is selected. The mentioned frequency value depends upon the system frequency & divider value. This SCPI is only for M8041A.
Example	:TRIG:FREQ? ‘M1.TrigOut’

:TRIGger:INTernal[:SOURce][?]

Syntax	:TRIGger:INTernal[:SOURce] ‘identifier’, <AXIFrame|INTernal> :TRIGger:INTernal[:SOURce]? ‘identifier’
Input Parameters	‘identifier’: M1.ClkGen <AXIFrame>: Select 100 MHz reference of AXIFrame as reference clock. <INTernal>: Select 100 MHz reference of the M8041A as reference clock.
Return Range	AXIF|INT
Description	In the INTernal mode different reference clock sources are available for the internal oscillator. Select between the 100 MHz reference clock of the M9505A AXIe Chassis or the 100 MHz reference clock of the M8041A module. This SCPI is only for M8041A.
Example	:TRIGger:INTernal:SOURce ‘M1.ClkGen’,INT :TRIGger:INTernal:SOURce? ‘M1.ClkGen’

:TRIGger:REFerence[:FREQuency][?]

Syntax	:TRIGger:REFerence[:FREQuency] 'identifier', <REF10|REF100>
	:TRIGger:REFerence[:FREQuency]? 'identifier'
Input Parameters	'identifier': 'M1.ClkGen'
	<REF10|REF100>: Set external reference clock to 10 MHz or 100 MHz.
Return Range	REF10|REF100
Description	In the REFerence mode two different expected external reference clocks can be specified: 10 MHz / 100 MHz.
	This SCPI is only for M8041A.
Example	:TRIGger:REFerence:FREQuency 'M1.ClkGen', REF100

:TRIGger:DIRect[:FREQuency][?]

Syntax	:TRIGger:DIRect[:FREQuency] 'identifier', <NRf>
	:TRIGger:DIRect[:FREQuency]? 'identifier'
Input Parameters	'identifier': 'M1.ClkGen'
	<NRf>: Enter frequency value at REF CLK IN.
Return Range	8.1 GHz to 16.207 GHz
Description	In the DIRect mode the frequency value expected at the REF CLK IN port can be set manually. Acceptable units include MHz, GHz and exponents (for example, 10E9 is the same as 10 GHz).
	This SCPI is only for M8041A.
Example	:TRIG:DIR:FREQ 'M1.ClkGen',10e9

:TRIGger:DIRect[:FREQuency]:DETect

Syntax	:TRIGger:DIRect[:FREQuency]:DETect 'identifier'
Input Parameters	'identifier': 'M1.ClkGen'
Description	The externally provided clock on the REF CLK IN port is measured once and used as the new system frequency (data rate), as required. This SCPI is only for M8041A.
Example	:TRIG:DIR:DET 'M1.ClkGen'

:TRIGger:CMULtiplier:LBWidth[?]

Syntax	:TRIGger:CMULtiplier:LBWidth 'identifier', <BW100|BW2|BW5> :TRIGger:CMULtiplier:LBWidth? 'identifier'
Input Parameters	'identifier': 'M1.ClkGen' <BW100|BW2|BW5>: Specify loop bandwidth.
Return Range	BW100|BW2|BW5
Description	Different loop bandwidths can be selected. Modulated signals (for example SSC) at the REF CLK IN port can be provided depending on the loop band width. This SCPI is only for M8041A.
Example	:TRIG:CMUL:LBW 'M1.ClkGen', BW2

:TRIGger:CMULtiplier:FREQuency:MULTiplier[?]

Syntax	:TRIGger:CMULtiplier:FREQuency:MULTiplier 'identifier', <NRf> :TRIGger:CMULtiplier:FREQuency:MULTiplier? 'identifier'
Input Parameters	'identifier': 'M1.ClkGen' <NRf>: Specify the multiplier value.
Return Range	2 to 1620; the range is dependent on other system parameters. See Command Syntax to Find Min/Max Values on page 111.

Description This command specifies the external reference clock frequency multiplier.

This SCPI is only for M8041A.

Example :TRIG:CMUL:FREQ:MULT 'M1.ClkGen',3

:TRIGger:CMULtiplier:FREQuency:DIVider[?]

Syntax :TRIGger:CMULtiplier:FREQuency:DIVider 'identifier', <NRf>

:TRIGger:CMULtiplier:FREQuency:DIVider? 'identifier'

Input Parameters 'identifier': 'M1.ClkGen'

<NRf>: Specify divider value.

Return Range 1 to 63; Range is dependent on other system parameters.

Description This command specifies the external reference clock frequency divider.

This SCPI is only for M8041A.

Example :TRIG:CMUL:FREQ:DIV 'M1.ClkGen',1

:TRIGger:CMULtiplier:FREQuency[?]

Syntax :TRIGger:CMULtiplier:FREQuency 'identifier', <NRf>

:TRIGger:CMULtiplier:FREQuency? 'identifier'

Input Parameters 'identifier': 'M1.ClkGen'

<NRf>: Specify reference frequency value.

Return Range 10 MHz to 202.5875 MHz; the range is dependent on other system parameters. See Command Syntax to Find Min/Max Values on page 111.

Description This command is used to set the expected frequency value of the external provided clock at REF CLK IN manually. The sent value will be multiplied / divided for calculating the system frequency (data rate). Acceptable units include kHz, MHz and exponents (for example, 200E6 is the same as 200 MHz).

This SCPI is only for M8041A.

Example :TRIG:CMUL:FREQ 'M1.ClkGen', 200e6

:TRIG:CMUL:FREQ? 'M1.ClkGen'

:TRIGger:CMULtiplier:FREQuency:DETect

Syntax	TRIGger:CMULtiplier:FREQuency:DETect ‘identifier’
Input Parameters	‘identifier’: ‘M1.ClkGen’
Description	The external provided clock at the REF CLK IN port is measured once and will be multiplied/divided for calculating the system frequency (data rate). This SCPI is only for M8041A.
Example	:TRIG:CMUL:FREQ:DET ‘M1.ClkGen’

Source Subsystem

The SOURce subsystem controls output signals (for example, for setting frequency and levels).

This subsystem has the following SCPI structure:

```
[:SOURce]
├─:FREQuecy
├─:PERiod[?]
├─:DRATe
├─:MEMoryCONFigure[:VALue][?]
├─:PULSe
│ └...
├─:VOLTage
│ └...
├─:JITTer
│ ├─ ...
│ ├─ :HFRequency
│ │  ├...
│ │  ├─:PERiodic(*)[?]
│ │  │  └...
│ │  ├─:BUNCorrelate
│ │  │  └...
│ │  ├─:RANDom
│ │  │  └...
│ │  └─:SPECtrally
│ │     └...
│ ├─ :LFRequency
│ │  ├...
│ │  ├─:PERiodic
│ │  │  └...
│ │  └─:RSSClocking
│ │     └...
│ └─ :SWEep
│    └...
├─:SSCLocking
│ └...
└─:INTerference
  └...
```

This subsystem has the following commands and subnodes:

Table 31

Name	Description under
:FREQuency[?]	[:SOURce]:FREQuency[?] on page 140
:PERiod[?]	[:SOURce]:PERiod[?] on page 141
:DRATe[?]	[:SOURce]:DRATe[?] on page 141
:MEMory:CONFigure[:VALue][?]	[:SOURce]:MEMory:CONFigure[:VALue][?] on page 142
Subnodes	
:PULse	[:SOURce]:PULSe Subnode on page 143
:VOLTage	[:SOURce]:VOLTage Subnode on page 146
:JITTer	[:SOURce]:JITTer Subnode on page 150
:JITTer:HFRequency	[:SOURce]:JITTer:HFRequency Subnode on page 152
:JITTer:LFRequency	[:SOURce]:JITTer:LFRequency Subnode on page 169
:JITTer:SWEep	[:SOURce]:JITTer:SWEep Subnode on page 175
:SSCLocking	[:SOURce]:SSCLocking Subnode on page 182
:INTerference	[:SOURce]:INTerference Subnode on page 187

[:SOURce]:FREQuency[?]

Syntax [:SOURce]:FREQuency ‘identifier’, <NRf>

[:SOURce]:FREQuency? ‘identifier’

Input Parameters ‘identifier’: ‘M1.ClkGen’

<NRf>: Set system frequency.

Return Range For M8041A: 256 MHz to 16.207 GHz

For M8195A: 256 MHz to 32.5 GHz

Description	Sets the frequency of the synthesizer in the M8020A/M8030A. Acceptable units include kHz, MHz, GHz and exponents (for example, 10E9 is the same as 10 GHz). This SCPI is only for M8041A, M8195A and M8062A. The SCPI is read only for M8062A.
Example	:FREQ ‘M1.ClkGen’, 10e9

[:SOURce]:PERiod[?]

Syntax	[:SOURce]:PERiod ‘identifier’, <NRf> [:SOURce]:PERiod? ‘identifier’
Input Parameters	‘identifier’: ‘M1.ClkGen’ <NRf>: Set system period.
Return Range	For M8041A: 61.702 ps to 3.906249 ns For M8195A: 30.77 ps to 3.90625 ns
Description	Sets the M8020A/M8030A system period. The period is the reciprocal value of the system frequency. This is provided as a convenience for those who prefer period instead of frequency. Acceptable units include ps, ns and exponents (for example, 61.702E-12 is the same as 61.702 ps). This SCPI is only for M8041A and M8195A.
Example	:PER ‘M1.ClkGen’, 61.702ps

[:SOURce]:DRATe[?]

Syntax	[:SOURce]:DRATe ‘identifier’, <NRf> [:SOURce]:DRATe? ‘identifier’
Input Parameters	‘identifier’: ‘M2.ClkGen’ <NRf>: Set data rate.
Return Range	512 Mb/s to 32.414 Gb/s
Description	This command/query sets/gets the data rate. This SCPI is only for M8062A.
Example	:DRAT 'M2.ClkGen', 10e9

[:SOURce]:MEMory:CONFigure[:VALue][?]

Syntax

[:SOURce]:MEMory:CONFigure ‘identifier’, <SINGle | DUAL | FOUR>

[:SOURce]:MEMory:CONFigure? ‘identifier’

Input Parameters

‘identifier’: ‘M1’

SINGle | DUAL | FOUR

Case 1 - The following description is applicable when the four channel license has been purchased:

- SINGle: DataOut1 will be sourced from extended memory and the other channels will be sourced from module internal memory. The data rate range in this mode is 256 Mb/s ... 32.5 Gb/s.
- DUAL: DataOut1 and DataOut2 will be sourced from extended memory and the other channels will be sourced from module internal memory. The data rate range in this mode is 256 Mb/s ... 16.250 Gb/s.
- FOUR: All four channels will be sourced from extended memory. The data rate range in this mode is 256 Mb/s ... 8.125 Gb/s.

Case 2 - The following description is applicable when the two channel license has been purchased:

- SINGle: DataOut1 will be sourced from extended memory and DataOut4 will be sourced from module internal memory. The data rate range in this mode is 256 Mb/s ... 32.5 Gb/s.
- DUAL: DataOut1 and DataOut4 will be sourced from extended memory. The data rate range in this mode is 256 Mb/s ... 16.250 Gb/s

Description

This command controls the memory configuration of M8195A AWG.

This query returns the current setting.

This SCPI is only for M8195A.

Example

:MEM:CONF 'M1', FOUR

:MEM:CONF? 'M1'

[:SOURce]:PULSe Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:PULSe
  ├:DELay[?]
  └:TRANsition
    ├:FIXed[?]
    └[:LEADing][?]
```

This subnode has the following commands:

Table 32

Name	Description under
:DELay[?]	[:SOURce]:PULSe:DELay[?] on page 143
:TRANsition:FIXed[?]	[:SOURce]:PULSe:TRANsition:FIXed[?] on page 144
:TRANsition[:LEADing][?]	[:SOURce]:PULSe:TRANsition[:LEADing][?] on page 144

[:SOURce]:PULSe:DELay[?]

Syntax

[:SOURce]:PULSe:DELay ‘identifier’, <NRf>

[:SOURce]:PULSe:DELay? ‘identifier’

Input Parameters

‘identifier’: 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'

<NRf>: Set the pulse delay time.

Return Range

For M8041A: 0.0 ps to 100 ns

For M8195A: -1ns to 1ns

For M8062A: 0 ps to 100 ns

Description: This command sets the time from the start of the period to the first edge of the pulse. Acceptable units include ps, ns and exponents (for example, 25E-9 is the same as 25 ns).

This SCPI is for M8041A, M8051A, M8195A and M8062A.

Example: :PULS:DEL ‘M1.DataOut2’, 25ns

[:SOURce]:PULSe:TRANsition:FIXed[?]

Syntax: [:SOURce]:PULSe:TRANsition:FIXed ‘identifier’, <SMOoth | MODerate | STEep>

[:SOURce]:PULSe:TRANsition:FIXed? ‘identifier’

Input Parameters: ‘identifier’: ‘M*.DATAOUT1’, ‘M*.DataOut2’, or ’M*.DataOut’
<SMOoth | MODerate | STEep>

Return Range: SMOoth | MODerate | STEep

Description: Use this command to set/read the transition time of the data output stream.

Following is the expected transition time (tr) for the below

mentioned settings:

1. Smooth: 20 ps typical (20%-80%)

2. Moderate: 17 ps typical (20%-80%)

3. Steep: 12 ps typical (20%-80%)

This SCPI is for M8041A, M8051A and M8062A.

Example: :PULS:TRAN:FIX 'M1.DataOut2', SMO

:PULS:TRAN:FIX? 'M1.DataOut2'

[:SOURce]:PULSe:TRANsition[:LEADing][?]

Syntax: [:SOURce]:PULSe:TRANsition[:LEADing] ‘identifier’,<NRf>

[:SOURce]:PULSe:TRANsition[:LEADing]? ‘identifier’

Input Parameters: ‘identifier’: 'M*.DataOut1’,‘M*.DataOut2’, ‘M*.DataOut3’ and ‘M*.DataOut4’

<NRf>: Specify transition time value.

Description Use this command to set/read the transition time of the data output stream. The fundamental units for transition time is seconds.

This SCPI is for M8195A.

Example :PULS:TRAN 'M1.DataOut2',15e-12

:PULS:TRAN? 'M1.DataOut2'

NOTE It is not possible to define rise and fall times separately. In other words, the rise and fall time will be in the same range.

[:SOURce]:VOLTage Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:VOLTage
  ├[:AMPLitude][?]
  ├:OFFSet[?]
  ├:HIGH[?]
  ├:LOW[?]
  └:RANGe
    ├[:SELect][?]
    └:AUTO[?]
```

This subnode has the following commands:

Table 33

Name	Description under
[:AMPLitude][?]	[:SOURce]:VOLTage[:AMPLitude][?] on page 147
:OFFSet[?]	[:SOURce]:VOLTage:OFFSet[?] on page 147
:HIGH[?]	[:SOURce]:VOLTage:HIGH[?] on page 148
:LOW[?]	[:SOURce]:VOLTage:LOW[?] on page 148
:RANGe[:SELect][?]	[:SOURce]:VOLTage:RANGe[:SELect][?] on page 149
:RANGe:AUTO[?]	[:SOURce]:VOLTage:RANGe:AUTO[?] on page 149

[:SOURce]:VOLTage[:AMPLitude][?]

Syntax	[:SOURce]:VOLTage[:AMPLitude] ‘identifier’, <NRf>
	[:SOURce]:VOLTage[:AMPLitude]? ‘identifier’
Input Parameters	‘identifier’: 'M1.ClkOut', 'M1.TrigOut', 'M1.SysOutA', 'M1.SysOutB', 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', 'M*.DataOut4', ‘M*.DataOut’, or 'M*.CtrlOutA'
	<NRf>: Set peak to peak voltage amplitude.
Return Range	<NRf>
Description	This command sets the peak to peak value of an output signal in Volts addressed by an identifier. Acceptable units include mV, V and exponents (for example, 500E-3 is the same as 500 mV).
	This SCPI is for M8041A, M8051A, M8061A, M8062A and M8195A.
Example	:VOLT ‘M1.DataOut1’, 500mv
	:VOLT? ‘M1.DataOut1’

[:SOURce]:VOLTage:OFFSet[?]

Syntax	[:SOURce]:VOLTage:OFFSet ‘identifier’, <NRf>
	[:SOURce]:VOLTage:OFFSet? ‘identifier’
Input Parameters	‘identifier’: 'M1.ClkOut', 'M1.TrigOut', 'M1.SysOutA', 'M1.SysOutB', 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', 'M*.DataOut4', ‘M*.DataOut’, or 'M*.CtrlOutA'
	<NRf>: Set offset voltage.
Return Range	<NRf>
Description	This command sets the offset value of an output signal in Volts addressed by an identifier. Acceptable units include mV, V and exponents (for example, 500E-3 is the same as 500 mV).
	This SCPI is for M8041A, M8051A, M8061A, M8062A and M8195A.
Example	:VOLT:OFFS ‘M1.DataOut1’, 500mv
	:VOLT:OFFS? ‘M1.DataOut1’

[:SOURce]:VOLTage:HIGH[?]

Syntax [:SOURce]:VOLTage:HIGH ‘identifier’, <NRf>

[:SOURce]:VOLTage:HIGH? ‘identifier’

Input Parameters ‘identifier’: 'M1.ClkOut', 'M1.TrigOut', 'M1.SysOutA', 'M1.SysOutB', 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', 'M*.DataOut4', ‘M*.DataOut’, or 'M*.CtrlOutA'

<NRf>: Set upper voltage level.

Return Range <NRf>

Description This command sets the upper voltage level of an output signal in Volts addressed by an identifier. Acceptable units include mV, V and exponents (for example, 500E-3 is the same as 500 mV).

This SCPI is for M8041A, M8051A, M8061A, M8062A and M8195A.

Example :VOLT:HIGH ‘M1.DataOut1’, 500mv

:VOLT:HIGH? ‘M1.DataOut1’

[:SOURce]:VOLTage:LOW[?]

Syntax [:SOURce]:VOLTage:LOW ‘identifier’, <NRf>

[:SOURce]:VOLTage:LOW? ‘identifier’

Input Parameters ‘identifier’: 'M1.ClkOut', 'M1.TrigOut', 'M1.SysOutA', 'M1.SysOutB', 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', 'M*.DataOut4', ‘M*.DataOut’, or 'M*.CtrlOutA'

<NRf>: Set lower voltage level.

Return Range <NRf>

Description This command sets the lower voltage level of an output signal in Volts addressed by an identifier.

This SCPI is for M8041A, M8051A, M8061A, M8062A and M8195A.

Example :VOLT:LOW ‘M2.DataOut2’,100 mv

:VOLT:LOW? ‘M2.DataOut2’

[:SOURce]:VOLTage:RANGe[:SELect][?]

Syntax	[:SOURce]:VOLTage:RANGE[:SELect] ‘identifier’, <R1|R2|R3|R4|R5|R6|R7|R8>
	[:SOURce]:VOLTage:RANGE[:SELect]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
	<R1|R2|R3|R4|R5|R6|R7|R8>: Select amplitude ranges.
Return Range	R1|R2|R3|R4|R5|R6|R7|R8
Description	Amplitude ranges guarantee a glitch free change of the amplitude value within a specified range. The upper and lower limits of these ranges are specified in the data sheet. Ranges are only specified for the DataOut channel.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:VOLT:RANG ‘M2.DataOut2’, R4

[:SOURce]:VOLTage:RANGe:AUTO[?]

Syntax	[:SOURce]:VOLTage:RANGE:AUTO ‘identifier’, <ON|OFF|1|0>
	[:SOURce]:VOLTage:RANGE:AUTO? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
	<ON|OFF|1|0>: Enable/disable auto tracking.
Return Range	1|0
Description	Disable/enable the automatic tracking of the amplitude ranges.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:VOLT:RANG:AUTO ‘M2.DataOut2’, off

[:SOURce]:JITTer Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
  ├[:GLOBal]
  │└[:STATe][?]
  ├:CONFigure
  │├[:DELay][?]
  │└:HMODe[?]
  ├:HFRequency
  │└…
  └:LFRequency
   └…
```

This subnode has the following commands:

Table 34

Name	Description under
[:GLOBal][:STATe][?]	[:SOURce]:JITTer[:GLOBal][:STATe][?] on page 151
:CONFigure[:DELay][?]	[:SOURce]:JITTer:CONFigure[:DELay][?] on page 151
:CONFigure:HMODe[?]	[:SOURce]:JITTer:CONFigure:HMODe[?] on page 151
:HFRequency	[:SOURce]:JITTer:HFRequency Subnode on page 152
:LFRequency	[:SOURce]:JITTer:LFRequency Subnode on page 169

[:SOURce]:JITTer[:GLOBal][:STATe][?]

Syntax	[:SOURce]:JITTer[:GLOBal][:STATe] ‘identifier’, <ON|OFF|1|0>
	[:SOURce]:JITTer[:GLOBal][:STATe]? ‘identifier’
Input Parameters	‘identifier’: 'M1.System'
	<ON|OFF|1|0>: Enable/disable global state.
Return Range	1|0
Description	This command enables or disables global jitter.
	It is a global SCPI; working for all modules.
Example	:JITT:STAT ‘M1.System’, off

[:SOURce]:JITTer:CONFigure[:DELay][?]

Syntax	[:SOURce]:JITTer:CONFigure[:DELay] ‘identifier’, <NRf>
	[:SOURce]:JITTer:CONFigure[:DELay]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', ‘M1.ClkOut’, or 'M*.DataOut'
	<NRf>: Set jitter profile delay.
Return Range	-40 ns to 40 ns
Description	Sets the delay of the jitter profile of the corresponding output.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:CONF:DEL ‘M1.DataOut2’, 2e-9

[:SOURce]:JITTer:CONFigure:HMODe[?]

Syntax	[:SOURce]:JITTer:CONFigure:HMODe 'Identifier', < ON | OFF | 1 | 0>
	[:SOURce]:JITTer:CONFigure:HMODe? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut'
	< ON | OFF | 1 | 0>
Return Range	1 | 0
Description	Sets the High Jitter Injection Mode of the corresponding output.
	This SCPI is only for M8062A.
Example	JITT:CONF:HMOD 'M2.DataOut', OFF

[:SOURce]:JITTer:HFRequency Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
 └:HFRequency
  ├:UNIT[?]
  ├:EXTernal
  │└[:STATe][?]
  ├:PMMode[?]
  ├:PERiodic(*)[?]
  │└...
  │
  ├:BUNCorrelate
  │└...
  │
  ├:RANDom
  │└...
  │
  └:SPECtrally
   └...
```

This subnode has the following commands:

Table 35

Name	Description under
:UNIT[?]	[:SOURce]:JITTer:HFRequency:UNIT[?] on page 153
:EXTernal[:STATe][?]	[:SOURce]:JITTer:HFRequency:EXTernal[:STATe][?] on page 153
:PMMode[?]	[:SOURce]:JITTer:HFRequency:PMMode[?] on page 154
:PERiodic(*)[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?] on page 156
:BUNCorrelate	[:SOURce]:JITTer:HFRequency:BUNCorrelate Subnode on page 158

Name	Description under
:RANDom	[:SOURce]:JITTer:HFRequency:RANDom Subnode on page 162
:SPECtrally	[:SOURce]:JITTer:HFRequency:SPECtrally Subnode on page 166

[:SOURce]:JITTer:HFRequency:UNIT[?]

Syntax	[:SOURce]:JITTer:HFRequency:UNIT ‘identifier’, <UINTerval|TIME> [:SOURce]:JITTer:HFRequency:UNIT? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut' <UINTerval|TIME>: Specify jitter parameter units.
Return Range	UINT|TIME
Description	Specifies whether the jitter parameters are to be specified/returned in seconds (TIME) or unit intervals (UINTerval). This SCPI is for M8041A, M8051A, M8061A and M8062A. This SCPI is also available on M8195A for DataOut1 to 4.
Example	:JITT:HFR:UNIT ‘M1.DataOut1’, TIME

[:SOURce]:JITTer:HFRequency:EXTernal[:STATe][?]

Syntax	[:SOURce]:JITTer:HFRequency:EXTernal[:STATe] ‘identifier’, <ON|OFF|1|0> [:SOURce]:JITTer:HFRequency:EXTernal[:STATe]? ‘identifier’
Input Parameters	‘identifier’: 'M1.ClkOut', 'M*.DataOut1' or 'M*.DataOut2' <ON|OFF|1|0>: Enable/disable external jitter input.
Return Range	1|0
Description	Enable/disable the external jitter input connector for Data Out1, Data Out2 and Clock Out. The name of the connectors are DATA MOD IN and CLK MOD IN. This SCPI is only for M8041A and M8051A.
Example	:JITT:HFR:EXT:STAT ‘M1.DataOut2’, on

[:SOURce]:JITTer:HFRequency:PMMode[?]

Syntax	[:SOURce]:JITTer:HFRequency:PMMode ‘identifier’, <UI|TIME> [:SOURce]:JITTer:HFRequency:PMMode? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1' or 'M*.DataOut2' <UI|TIME>: Specify unit interval or time.
Return Range	UI|TIME
Description	The internal delay line is optimized for either ‘UI’ (1UI; the maximum of the delay line is not used) or for ‘TIME’ rather than the whole range of the delay line, which can be used independent of the data rate. PMMode stands for Phase Modulation Mode. This SCPI is only for M8041A and M8051A.
Example	:JITT:HFR:PMM ‘M1.DataOut2’, UI

[:SOURce]:JITTer:HFRequency:PERiodic Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
  └:HFRequency
    └:PERiodic(*)
      ├[:STATe][?]
      ├:AMPLitude[?]
      └:FREQuency[?]
```

This subnode has the following commands:

Table 36

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?] on page 156
:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude[?] on page 156
:FREQuency[?]	URce]:JITTer:HFRequency:PERiodic(*):FREQuency[?] on page 157

[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?]

Syntax

[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe] ‘identifier’, <ON|OFF|1|0>

[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe]? ‘identifier’

Input Parameters

‘identifier’: 'M1.ClkOut', 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

<ON|OFF|1|0>: Enable/disable periodic jitter.

Return Range

1|0

Description

This command is used to enable/disable periodic jitter high frequency output at the given location.

The “(*)” suffix specifies which of the two periodic sources (1 or 2) for the specified channel to use. If a suffix is not specified, the suffix 1 is assumed.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

This SCPI is also available on M8195A for DataOut1 to 4.

Example

:JITT:HFR:PER ‘M2.DataOut2’, off

[:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude[?]

Syntax

[:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude ‘identifier’, <NRf>

[:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude? ‘identifier’

Input Parameters

‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

<NRf>: Set periodic jitter amplitude.

Return Range

0 mUI to 1.102 UI; The range is dependent on other system parameters. See Command Syntax to Find Min/Max Values on page 111.

Range for M8195A is 0 to 10 UI.

Description

This command sets the amplitude of the periodic jitter in seconds or unit intervals for the specified output. The units are set using the [:SOURce]:JITTer:HFRequency:UNIT command.

The “(*)” suffix specifies which of the two periodic sources (1 or 2) for the specified channel to use. If a suffix is not specified, the suffix 1 is assumed.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

This SCPI is also available on M8195A for DataOut1 to 4.

Example

:JITT:HFR:PER:AMPL ‘M2.DataOut2’, 100e-12

URce]:JITTer:HFRequency:PERiodic(*):FREQuency[?]

Syntax	[:SOURce]:JITTer:HFRequency:PERiodic(*):FREQuency ‘identifier’, <NRf>
	[:SOURce]:JITTer:HFRequency:PERiodic(*):FREQuency? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1' or 'M*.DataOut2', or 'M*.DataOut'
	<NRf>: Set periodic jitter frequency.
Return Range	1 kHz to 500 MHz
Description	This command sets the frequency of the periodic jitter for the specified output.
	The “(*)” suffix specifies which of the two periodic sources (1 or 2) for the specified channel to use. If a suffix is not specified, the suffix 1 is assumed. Acceptable units include Hz, kHz, MHz and exponents (for example, 1E3 is the same as 1,000 Hz).
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
	This SCPI is also available on M8195A for DataOut1 to 4.
Example	:JITT:HFR:PER:FREQ ‘M1.DataOut2’, 20.0e6

[:SOURce]:JITTer:HFRequency:BUNCorrelate Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
  └:HFRequency
    └:BUNCorrelate
      ├[:STATe][?]
      ├:AMPLitude[?]
      ├:DRATe[?]
      └:FILTer
        └[:SELect][?]
       :PRBSequence
        └[:SELect][?]
```

This subnode has the following commands:

Table 37

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate[:STATe][?] on page 159
:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:AMPLitude[?] on page 159
:DRATe[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:DRATe[?] on page 160
:FILTer[:SELect][?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:FILTer[:SELect][?] on page 160
:PRBSequence[:SELect][?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect][?] on page 161

[:SOURce]:JITTer:HFRequency:BUNCorrelate[:STATe][?]

Syntax	[:SOURce]:JITTer:HFRequency:BUNCorrelate[:STATe] ‘identifier’ <ON|OFF|1|0> [:SOURce]:JITTer:HFRequency:BUNCorrelate[:STATe]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', ‘M1.ClkOut’, or 'M*.DataOut' <ON|OFF|1|0>: Enable/disable bounded uncorrelated jitter.
Return Range	1|0
Description	Enables/disables the generation of bounded uncorrelated jitter at the specified output. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:HFR:BUNC ‘M2.DataOut2’, off

[:SOURce]:JITTer:HFRequency:BUNCorrelate:AMPLitude[?]

Syntax	[:SOURce]:JITTer:HFRequency:BUNCorrelate:AMPLitude ‘identifier’, <NRf> [:SOURce]:JITTer:HFRequency:BUNCorrelate:AMPLitude? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1' or 'M*.DataOut2', or 'M*.DataOut' <NRf>: Set bounded uncorrelated jitter amplitude.
Return Range	NRf
Description	This command sets the bounded uncorrelated jitter amplitude in seconds (TIME) or unit intervals (UINTerval). The units are set using the [:SOURce]:JITTer:HFRequency:UNIT command. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:HFR:BUNC:AMPL ‘M2.DataOut2’, 400e-12

[:SOURce]:JITTer:HFRequency:BUNCorrelate:DRATe[?]

Syntax	[:SOURce]:JITTer:HFRequency:BUNCorrelate:DRATe ‘identifier’ <RATE625|RATE1250|RATE2500> [:SOURce]:JITTer:HFRequency:BUNCorrelate:DRATe? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut' <RATE625|RATE1250|RATE2500>: Set the bounded uncorrelated PRBS data rate.
Return Range	RATE625|RATE1250|RATE2500
Description	This command sets the bounded uncorrelated PRBS data rate to 625 MBps, 1250 MBps, or 2500 MBps. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:HFR:BUNC:DRAT ‘M2.DataOut2’, RATE625

[:SOURce]:JITTer:HFRequency:BUNCorrelate:FILTer[:SELect][?]

Syntax	[:SOURce]:JITTer:HFRequency:BUNCorrelate:FILTer[:SELect] ‘identifier’ <LP50|LP100|LP200> [:SOURce]:JITTer:HFRequency:BUNCorrelate:FILTer[:SELect]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut’ <LP50|LP100|LP200>: Select the low-pass filter.
Return Range	LP50|LP100|LP200
Description	This command selects the low-pass filter for bounded uncorrelated jitter (50, 100, or 200 MHz). This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:HFR:BUNC:FILT ‘M2.DataOut2’, LP50

[:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect][?]

Syntax	[:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect] 'identifier', <PRBS7|PRBS8|PRBS9|PRBS10|PRBS11|PRBS15|PRBS23|PRBS31 [:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect]? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut' <PRBS7|PRBS8|PRBS9|PRBS10|PRBS11|PRBS15|PRBS23|PRBS31>: Select the PRBS polynomial.
Return Range	PRBS7|PRBS8|PRBS9|PRBS10|PRBS11|PRBS15|PRBS23|PRBS31
Description	This command selects polynomial of the PRBS for bounded uncorrelated jitter source. This query returns the current setting. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:HFR:BUNC:PRBS 'M2.DataOut2', PRBS7

[:SOURce]:JITTer:HFRequency:RANDom Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
  └:HFRequency
    └:RANDom
      ├[:STATe][?]
      ├:AMPLitude [?]
      └:FILTer
        ├[:LPASs][?]
        │└:VALue[?]
        └:HPASs[?]
          └:VALue[?]
```

This subnode has the following commands:

Table 38

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:HFRequency:RANDom[:STATe][?] on page 163
:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:RANDom:AMPLitude[?] on page 163
:FILTer[:LPASs][?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs][?] on page 163
:FILTer[:LPASs]:VALue[?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs]:VALue[?] on page 164
:FILTer:HPASs[?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs[?] on page 164
:FILTer:HPASs:VALue[?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs:VALue[?] on page 165

[:SOURce]:JITTer:HFRequency:RANDom[:STATe][?]

Syntax	[:SOURce]:JITTer:HFRequency:RANDom[:STATe] 'identifier' <ON|OFF|1|0>
	[:SOURce]:JITTer:HFRequency:RANDom[:STATe]? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', or 'M*.DataOut'
	<ON|OFF|1|0>: Enable/disable random jitter source.
Return Range	0|1
Description	This command enables/disables the generation of random jitter.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
	This SCPI is also available on M8195A for DataOut1 to 4.
Example	:JITT:HFR:RAND 'M2.DataOut2', on

[:SOURce]:JITTer:HFRequency:RANDom:AMPLitude[?]

Syntax	[:SOURce]:JITTer:HFRequency:RANDom:AMPLitude 'identifier' <NRf>
	[:SOURce]:JITTer:HFRequency:RANDom:AMPLitude? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' or 'M*.DataOut2', or 'M*.DataOut'
	<NRf>: Set random jitter amplitude.
Return Range	0 uUI to 13.08 mUI or 0 ps to 78.57 ps
Description	This command sets the root mean square (rms) RJ amplitude in seconds or unit intervals based on the selected amplitude unit. The units are set using the [:SOURce]:JITTer:HFRequency:UNIT command.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
	This SCPI is also available on M8195A for DataOut1 to 4.
Example	:JITT:HFR:RAND:AMPL 'M2.DataOut2', 0.013

[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs][?]

Syntax	[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs] 'identifier' <OFF|LP100|LP500>
	[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs]? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
	<OFF|LP100|LP500>: Enable/disable low-pass filter.

Return Range OFF|LP100|LP500

Description This command enables/disables the low-pass filter for random jitter. LP100 enables a 100 MHz low-pass filter; LP500 enables a 500 MHz low-pass filter.

This query returns the current settings.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example :JITT:HFR:RAND:FILT ‘M1.DataOut2’, LP100

[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs]:VALue[?]

Syntax [:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs]:VALue <identifier>,<NRf>

[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs]:VALue? <identifier>

Input Parameters ‘identifier’: 'M*.DataOut1', 'M*.DataOut2', M*.DataOut3, M*.DataOut3, or ‘M*.DataOut’

<NRf>

Return Range 1 MHz to 10 GHz

The limit is dynamic, it depends on the high-pass setting.

Description This command controls the low pass filter of the random jitter source.

This query returns the current settings.

This SCPI is for M8195A.

Example :JITT:HFR:RAND:FILT:LPAss:VALue ‘M1.DataOut2’,1e6

[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs[?]

Syntax [:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs ‘identifier’ <OFF|HP10>

[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs? ‘identifier’

Input Parameters ‘identifier’: 'M*.DataOut1' or 'M*.DataOut2', or ‘M*.DataOut’

<OFF|HP10>: Enable/disable high-pass filter.

Return Range OFF|HP10

Description
This command enables/disables the high-pass filter for random jitter. HP10 enables a 10 MHz high-pass filter.

This query returns the current settings.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example
:JITT:HFR:RAND:FILT:HPAS ‘M1.DataOut2’, off

[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs:VALue[?]

Syntax
[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs ‘identifier’ <NRf>

[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs:VALue? ‘identifier’

Input Parameters
‘identifier’: 'M*.DataOut1' or 'M*.DataOut2', or ‘M*.DataOut’

<NRf>

Return Range
1 kHz to 10 MHz

The limit is dynamic, it depends on the low-pass setting.

Description
This command controls the high pass filter value of the random jitter source.

This SCPI is for M8195A.

This query returns the current settings.

Example
:JITT:HFR:RAND:FILT:HPAss:VALue ‘M1.DataOut2’, 1E6.

[:SOURce]:JITTer:HFRequency:SPECtrally Subnode

Spectrally distributed random jitter is composed of two jitter sources: low frequency jitter and high frequency jitter. It is characterized by the amplitudes of the low and the high frequency jitter part.

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
  └:HFRequency
    └:SPECtrally
      ├[:STATe][?]
      ├:AMPLitude(*)[?]
      └:FILTer
        └[:LPASs]
          └[:STATe][?]
```

This subnode has the following commands:

Table 39

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe][?] on page 167
:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*)[?] on page 167
:FILTer[:LPASs][:STATe][?]	[:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs][:STATe][?] on page 168

[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe][?]

Syntax	[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe] ‘identifier’ <ON|OFF|1|0>
	[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', ‘M1.ClkOut’, or ‘M*.DataOut’
	<ON|OFF|1|0>: Enable/disable spectrally distributed jitter.
Return Range	0|1
Description	This command enables/disables the generation of spectrally distributed random jitter.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:HFR:SPEC ‘M2.DataOut2’, off

[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*)[?]

Syntax	[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*) ‘identifier’ <NRf>
	[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*)? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or ‘M*.DataOut’
	<NRf>: Set spectrally distributed jitter amplitude.
Return Range	0 mUI to 78.71 mUI
Description	This command sets the spectrally distributed jitter amplitude in unit intervals or time.
	A suffix attached to the :AMPLitude(*) command is used to address two jitter amplitudes. The :AMPL1 <NR3> command changes the level of the internal RJLF resource and the :AMPL2 <NR3> command changes the level of the internal RJHF resource. Both levels are specified in rms.
	The units are set using the [:SOURce]:JITTer:HFRequency:UNIT command.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:HFR:SPEC:AMPL2 ‘M2.DataOut2’, 0.013

[:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs][:STATe][?]

Syntax	[:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs] [:STATe] ‘identifier’ <ON|OFF|1|0> [:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs] [:STATe]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1' or 'M*.DataOut2' <ON|OFF|1|0>: Enable/disable low-pass filter.
Return Range	1|0
Description	Enables/disables the 100 MHz low-pass filter for spectrally distributed random jitter. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:HFR:SPEC:FILT ‘M2.DataOut2’, off

[:SOURce]:JITTer:LFRequency Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
  └:LFRequency
    ├:UNIT[?]
    ├:PERiodic
    │ └...
    │
    └:RSSClocking
      └...
```

This subnode has the following commands:

Table 40

Name	Description under
:UNIT[?]	[:SOURce]:JITTer:LFRequency:UNIT[?] on page 170
:PERiodic	[:SOURce]:JITTer:LFRequency:PERiodic Subnode on page 171
:RSSClocking	[:SOURce]:JITTer:LFRequency:RSSClocking Subnode on page 173

[:SOURce]:JITTer:LFRequency:UNIT[?]

Syntax	[:SOURce]:JITTer:LFRequency:UNIT ‘identifier’, <UINTerval|TIME>
	[:SOURce]:JITTer:LFRequency:UNIT? ‘identifier’
Input Parameters	‘identifier’: 'M1.ClkOut', 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
	<UINTerval|TIME>: Specify jitter parameter units.
Return Range	UINT|TIME
Description	Specifies whether the jitter parameters are to be specified/returned in seconds (TIME) or unit intervals (UINTerval).
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:LFR:UNIT ‘M1.DataOut2’,TIME

[:SOURce]:JITTer:LFRequency:PERiodic Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
  └:LFRequency
    └:PERiodic
      ├[:STATe][?]
      ├:AMPLitude[?]
      └:FREQuency[?]
```

This subnode has the following commands:

Table 41

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:LFRequency:PERiodic[:STATe][?] on page 171
:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:PERiodic:AMPLitude[?] on page 172
:FREQuency[?]	[:SOURce]:JITTer:LFRequency:PERiodic:FREQuency[?] on page 172

[:SOURce]:JITTer:LFRequency:PERiodic[:STATe][?]

Syntax: [:SOURce]:JITTer:LFRequency:PERiodic[:STATe] ‘identifier’, <ON|OFF|1|0>

[:SOURce]:JITTer:LFRequency:PERiodic[:STATe]? ‘identifier’

Input Parameters: ‘identifier’: 'M1.ClkOut', 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

<ON|OFF|1|0>: Enable/disable the low frequency periodic jitter source.

Return Range: 1|0

Description: This command enables/disables the low frequency periodic jitter source.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example: :JITT:LFR:PER ‘M1.ClkOut’, off

[:SOURce]:JITTer:LFRequency:PERiodic:AMPLitude[?]

Syntax	[:SOURce]:JITTer:LFRequency:PERiodic:AMPLitude ‘identifier’, <NRf>
	[:SOURce]:JITTer:LFRequency:PERiodic:AMPLitude? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', M1.ClkOut’, or ‘M*.DataOut’
	<NRf>: Set periodic low frequency jitter amplitude.
Return Range	NRf
Description	This command sets the amplitude of the periodic low frequency jitter in unit intervals or time for the specified output. The units are set using the [:SOURce]:JITTer:LFRequency:UNIT command.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:LFR:PER:AMPL ‘M2.DataOut2’, 1

[:SOURce]:JITTer:LFRequency:PERiodic:FREQuency[?]

Syntax	[:SOURce]:JITTer:LFRequency:PERiodic:FREQuency ‘identifier’, <NRf>
	[:SOURce]:JITTer:LFRequency:PERiodic:FREQuency? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', M1.ClkOut’, or ‘M*.DataOut’
	<NRf>: Set periodic jitter frequency.
Return Range	NRf
Description	This command sets the frequency of the periodic jitter. Acceptable units include Hz, kHz, MHz and exponents (for example, 1E3 is the same as 1,000 Hz).
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:LFR:PER:FREQ ‘M2.DataOut2’, 1e6

[:SOURce]:JITTer:LFRequency:RSSClocking Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
  └:LFRequency
    └:RSSClocking
      ├[:STATe][?]
      ├:AMPLitude[?]
      └:FREQuency[?]
```

This subnode has the following commands:

Table 42

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:LFRequency:RSSClocking [:STATe][?] on page 173
:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:RSSClocking :AMPLitude[?] on page 174
:FREQuency[?]	[:SOURce]:JITTer:LFRequency:RSSClocking :FREQuency[?] on page 174

[:SOURce]:JITTer:LFRequency:RSSClocking[:STATe][?]

Syntax: [:SOURce]:JITTer:LFRequency:RSSClocking[:STATe] ‘identifier’, <ON|OFF|1|0>

[:SOURce]:JITTer:LFRequency:RSSClocking[:STATe]? ‘identifier’

Input Parameters: ‘identifier’: 'M1.ClkOut', 'M*.DataOut1', 'M*.DataOut2', or ‘M*.DataOut’

<ON|OFF|1|0>: Enable/disable residual spread spectrum clocking.

Return Range: 1|0

Description: This command enables/disables the residual spread spectrum clocking (rSSC).

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example: :JITT:LFR:RSSC ‘M2.DataOut2’, on

[:SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude[?]

Syntax	[:SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude ‘identifier’, <NRf>
	[:SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', M1.ClkOut’, or ‘M*.DataOut’
	<NRf>: Set residual spread spectrum clocking amplitude.
Return Range	NRf
Description	This command sets the amplitude of the residual spread spectrum clocking (rSSC). The units are set using the [:SOURce]:JITTer:LFRequency:UNIT command.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:LFR:RSSC:AMPL ‘M2.DataOut2’, 0.016

[:SOURce]:JITTer:LFRequency:RSSClocking:FREQuency[?]

Syntax	[:SOURce]:JITTer:LFRequency:RSSClocking:FREQuency ‘identifier’, <NRf>
	[:SOURce]:JITTer:LFRequency:RSSClocking:FREQuency? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', ‘M1.ClkOut’, or ‘M*.DataOut’
	<NRf>: Set residual spread spectrum clocking frequency.
Return Range	10 kHz to 100 kHz
Description	This command sets the frequency of the residual spread spectrum clocking (rSSC). Acceptable units include Hz, kHz and exponents (for example, 10E3 is the same as 10,000 Hz).
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:JITT:LFR:RSSC:FREQ ‘M2.DataOut1’, 10khz

[:SOURce]:JITTer:SWEep Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
  └:SWEep
    ├[:STATe][?]
    ├:FREQuency
    │ ├:STARt[?]
    │ └:STOP[?]
    ├:TIME[?]
    ├:STEP
    │ ├:DISTance[?]
    │ └[:VALue]?
    ├:AMPLitude
    │ ├:MODE[?]
    │ └[:VALue]?
    ├:PLOT
    │ ├:INPut?
    │ └[:VALue]?
    └:DATA
      └:FILE[?]
```

This subnode has the following commands:

Table 43

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:SWEep[:STATe][?] on page 177
:FREQuency:STARt[?]	[:SOURce]:JITTer:SWEep:FREQuency:STARt[?] on page 177
:FREQuency:STOP[?]	[:SOURce]:JITTer:SWEep:FREQuency:STOP[?] on page 177
:TIME[?]	[:SOURce]:JITTer:SWEep:TIME[?] on page 178

Name	Description under
:STEP:[VALue][?]	[:SOURce]:JITTer:SWEep:STEP:[VALue][?] on page 178
:STEP:DISTance[?]	[:SOURce]:JITTer:SWEep:STEP:DISTance[?] on page 179
:AMPLitude[:VALue]?	[:SOURce]:JITTer:SWEep:AMPLitude[:VALue]? on page 179
:AMPLitude:MODE[?]	[:SOURce]:JITTer:SWEep:AMPLitude:MODE[?] on page 179
:AMPLitude:[VALue]?	[:SOURce]:JITTer:SWEep:AMPLitude[:VALue]? on page 179
:PLOT[:VALue]?	[:SOURce]:JITTer:SWEep:PLOT[:VALue]? on page 180
:PLOT:INPut?	[:SOURce]:JITTer:SWEep:PLOT:INPut? on page 180
:DATA:FILE[?]	[:SOURce]:JITTer:SWEep:DATA:FILE[?] on page 181

[:SOURce]:JITTer:SWEep[:STATe][?]

Syntax	[:SOURce]:JITTer:SWEep[:STATe] ‘identifier’, <ON|OFF|1|0>
	[:SOURce]:JITTer:SWEep[:STATe]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'
	<ON|OFF|1|0>: Enable/disable periodic jitter sweep.
Return Range	1|0
Description	This command enables/disables the jitter sweep.
	This SCPI is only for M8041A and M8051A.
Example	:JITT:SWE ‘M1.DataOut2’, on

[:SOURce]:JITTer:SWEep:FREQuency:STARt[?]

Syntax	[:SOURce]:JITTer:SWEep:FREQuency:STARt ‘identifier’, <NRf>
	[:SOURce]:JITTer:SWEep:FREQuency:STARt? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'
	<NRf>: Set the start frequency.
Return Range	1 kHz to 500 MHz
Description	This command defines the start frequency of the jitter sweep. The start frequency must be lower than the stop frequency and the range should be in accordance with the waveform. Acceptable units include Hz, kHz, MHz, GHz and exponents (for example, 1E3 is the same as 1,000 Hz).
	This SCPI is only for M8041A and M8051A.
Example	:JITT:SWE:FREQ:STAR ‘M1.DataOut2’, 500kHz

[:SOURce]:JITTer:SWEep:FREQuency:STOP[?]

Syntax	[:SOURce]:JITTer:SWEep:FREQuency:STOP ‘identifier’, <NRf>
	[:SOURce]:JITTer:SWEep:FREQuency:STOP? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'
	<NRf>: Set the stop frequency.
Return Range	1 kHz to 500 MHz

Description: This command defines the stop frequency of the jitter sweep. The stop frequency must be higher than the start frequency and the range should be in accordance with the waveform. Acceptable units include Hz, kHz, MHz, GHz and exponents (for example, 1E3 is the same as 1,000 Hz).

This SCPI is only for M8041A and M8051A.

Example: :JITT:SWE:FREQ:STOP ‘M1.DataOut2’,500MHz

[:SOURce]:JITTer:SWEep:TIME[?]

Syntax: [:SOURce]:JITTer:SWEep:TIME ‘identifier’, <NRf>

[:SOURce]:JITTer:SWEep:TIME? ‘identifier’

Input Parameters: ‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'

<NRf>: Set duration of sweep.

Return Range: 100 ms to 1.2 ks

Description: This command defines the time duration of the jitter sweep. Acceptable units include ms (milliseconds), s (seconds), ks (kiloseconds) and exponents (for example, 1E3 is the same as 1,000 seconds).

This SCPI is only for M8041A and M8051A.

Example: :JITT:SWE:TIME ‘M1.DataOut2’, 100ms

[:SOURce]:JITTer:SWEep:STEP:[VALue][?]

Syntax: [:SOURce]:JITTer:SWEep:STEP:[VALue] ‘identifier’, <NRf>

[:SOURce]:JITTer:SWEep:STEP:[VALue]? ‘identifier’

Input Parameters: ‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'

<NRf>: Set number of steps for a complete sweep.

Return Range: 2 to 100

Description: This command defines the number of steps to fulfill a complete jitter sweep.

This SCPI is only for M8041A and M8051A.

Example: :JITT:SWE:STEP ‘M1.DataOut2’, 50

[:SOURce]:JITTer:SWEep:STEP:DISTance[?]

Syntax: [:SOURce]:JITTer:SWEep:STEP :DISTance ‘identifier’, <EQUidistant|AUTOmatic>

[:SOURce]:JITTer:SWEep:STEP :DISTance? ‘identifier’

Input Parameters: ‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'

<EQUidistan|AUTOmatic>: Specify log equidistant or frequency steps.

Return Range: EQU|AUTO

Description: The command defines whether the frequency steps are log equidistant (EQUidistant) along the periodic jitter curve or a frequency step matches a corner frequency on the jitter curve.

This SCPI is only for M8041A and M8051A.

Example: :JITT:SWE:STEP:DIST ‘M1.DataOut2’, AUTO

[:SOURce]:JITTer:SWEep:AMPLitude[:VALue]?

Syntax: [:SOURce]:JITTer:SWEep:AMPLitude[:VALue] ‘identifier’, <NR3>

[:SOURce]:JITTer:SWEep:AMPLitude[:VALue]? ‘identifier’

Input Parameters: ‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'

<NR3>: Specify constant jitter amplitude of the jitter sweep.

Return Range: 0 mUI to 1.102 UI

Description: This command sets the constant jitter amplitude of the jitter sweep. If CONStant jitter amplitude is selected as the jitter sweep mode, this command is used to define the value of the constant jitter amplitude of the jitter sweep. Acceptable units include mUI, UI and exponents (for example, 100e-3 is the same as 100 mUI).

This SCPI is only for M8041A and M8051A.

Example: :JITT:SWE:AMPL ‘M1.DataOut2’, 1UI

[:SOURce]:JITTer:SWEep:AMPLitude:MODE[?]

Syntax: [:SOURce]:JITTer:SWEep:AMPLitude:MODE ‘identifier’, <CONStant|VARiable>

[:SOURce]:JITTer:SWEep:AMPLitude:MODE? ‘identifier’

Input Parameters	'identifier': 'M*.DataOut1' or 'M*.DataOut2' <CONStant|VARiable>: Specify constant or user defined jitter level.
Return Range	CONS|VAR
Description	The command defines whether a sweep is defined with CONStant jitter level (amplitude) or with the user defined VARiable jitter level (amplitude). This SCPI is only for M8041A and M8051A.
Example	:JITT:SWE:AMPL:MODE 'M1.DataOut2', CONS

[:SOURce]:JITTer:SWEep:PLOT[:VALue]?

Syntax	[:SOURce]:JITTer:SWEep:PLOT[:VALue]? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' or 'M*.DataOut2'
Return Range	<frequency/jitter amplitude value pairs>
Description	This query returns a comma separated list (specified as an expression) of frequency/jitter amplitude value pairs. This list of values represents the 'real' steps performed by the jitter sweep. This SCPI is only for M8041A and M8051A.
Example	:JITT:SWE:PLOT? 'M1.DataOut2'

[:SOURce]:JITTer:SWEep:PLOT:INPut?

Syntax	[:SOURce]:JITTer:SWEep:PLOT :INPut? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' or 'M*.DataOut2'
Description	This query returns a comma separated list (specified as an expression) of frequency/jitter amplitude value pairs. This list of values represents the value pairs of the corner frequencies only. This SCPI is only for M8041A and M8051A.
Example	:JITT:SWE:PLOT:INP? 'M1.DataOut2'

[:SOURce]:JITTer:SWEep:DATA:FILE[?]

Syntax

[:SOURce]:JITTer:SWEep:DATA:FILE ‘identifier’ <“FileName”>

[:SOURce]:JITTer:SWEep:DATA:FILE? ‘identifier’

Input Parameters

‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'

<“FileName”>: Specify path to a data file.

Description

Every point of the periodic jitter sweep curve can be specified by a group of 2 value pairs Frequency, JitterLevel, Frequency, JitterLevel, etc.This command specifies the location of the file to be loaded within the workspace.

Jitter sweep profiles can be stored in three different areas:

1 Local to current setting ("current/")
2 Shared between settings ("shared/")
3 Factory supplied standard sweep profiles ("factory/").
These sweep profiles are read only and cannot be modified.

Below these root nodes there is a folder structure using '/' as separation character. So, a complete file name might be something like "factory/PCIe_8G_CC.jcs", or "shared/subfolder/mySweepProfile.jcs".

This SCPI is only for M8041A and M8051A.

Example

:JITT:SWE:DATA:FILE 'M1.DataOut2', "factory/PCIe_8G_CC.jcs"

[:SOURce]:SSCLocking Subnode

Spread Spectrum Clocking (SSCL) is used for reducing the peak electromagnetic radiation. The clock out signal and data out stream are modulated with a repetitive low-frequency waveform.

This subnode has the following SCPI structure:

```
[:SOURce]
└:SSCLocking
  ├ [:STATe][?]
  ├ :GLOBal[:STATe][?]
  ├ :DEViation[?]
  ├ :TYPE[?]
  ├ :PROFile[?]
  ├ :SHAPe[?]
  └ :FREQuency[?]
```

This subnode has the following commands:

Table 44

Name	Description under
[:STATe][?]	[:SOURce]:SSCLocking[:STATe][?] on page 183
GLOBal[:STATe][?]	:SOURce]:SSCLocking:GLOBal[:STATe][?] on page 184
:DEViation[?]	[:SOURce]:SSCLocking:DEViation[?] on page 184
:TYPE[?]	[:SOURce]:SSCLocking:TYPE[?] on page 185
:PROFile[?]	[:SOURce]:SSCLocking:PROFile[?] on page 185
:SHAPe[?]	[:SOURce]:SSCLocking:SHAPe[?] on page 186
:FREQuency[?]	[:SOURce]:SSCLocking:FREQuency[?] on page 186

[:SOURce]:SSCLocking[:STATe][?]

Syntax	[:SOURce]:SSCLocking[:STATe] 'identifier', <ON|OFF|1|0> [:SOURce]:SSCLocking[:STATe]? 'identifier'
Input Parameters	'identifier': 'M1.ClkGen' <ON|OFF|1|0>: Enable/disable spread spectrum clocking.
Return Range	1|0
Description	This command enables/disables spread spectrum clocking (SSC) state, which turns off the scrambler regardless of clock source. The SSC mode can be only enabled if Global SSC state is activated. This SCPI is only for M8041A and M8195A.
Example	The below mentioned SCPI command turns ON the SSC state: :SSCL 'M1.ClkGen', ON The below mentioned SCPI command queries the SSC: :SSCL? 'M1.ClkGen' 1

:SOURce]:SSCLocking:GLOBal[:STATe][?]

Syntax	[:SOURce]:SSCLocking:GLOBal[:STATe] 'identifier', <ON|OFF|1|0>
	[:SOURce]:SSCLocking:GLOBal[:STATe]? 'identifier'
Input Parameters	'identifier': 'M1.System'
	<ON|OFF|1|0>: Enable/disable spread spectrum clocking.
Return Range	1|0
Description	This command turns the spread spectrum clocking (SSC) mode globally on or off. This means that SSC will be only enable for a specific module if the module specific SSC switch and the global SSC switch are both turned on, otherwise SSC is disabled.
Example	SOUR:SSCL:GLOB 'M1.System',on.

[:SOURce]:SSCLocking:DEViation[?]

Syntax	[:SOURce]:SSCLocking:DEViation 'identifier', <NRf>
	[:SOURce]:SSCLocking:DEViation? 'identifier'
Input Parameters	'identifier': 'M1.ClkGen'
	<NRf>: Set spread spectrum clocking deviation.
Return Range	0% to 1%
Description	This command sets the spread spectrum clocking (SSC) deviation in percent. The effective deviation is ±0.5% (or 1% peak-to-peak).
	This SCPI is only for M8041A and M8195A.
Example	The below mentioned SCPI command sets the spread spectrum clocking (SSC) deviation to '0.2' percent for M8041A:
	SSCL:DEV 'M1.ClkGen', 0.2
	The below mentioned SCPI command queries the spread spectrum clocking (SSC) deviation for M8041A:
	SSCL:DEV? 'M1.ClkGen'
	0.2

[:SOURce]:SSCLocking:TYPE[?]

Syntax

[:SOURce]:SSCLocking:TYPE ‘identifier’, <DOWNspread|UPSPread|CENTerspread>

[:SOURce]:SSCLocking:TYPE? ‘identifier’

Input Parameters

‘identifier’: 'M1.ClkGen'

<DOWNspread|UPSPread|CENTerspread>: Set spread spectrum clocking type.

Return Range

DOWN|UPSP|CENT

Description

This command sets method used to modulate the clock frequency and data. Choose between center spread, down spread, or up spread.

This SCPI is only for M8041A and M8095A.

It is also applicable for M8195A but only in query mode.

Example

The below mentioned SCPI command sets the method used to modulate clock frequency and data to ‘DOWN’ for M8041A:

:SSCL:TYPE ‘M1.ClkGen’, DOWN

The below mentioned SCPI command queries the method used to modulate clock frequency and data for M8041A:

:SSCL:TYPE? ‘M1.ClkGen’

DOWN

[:SOURce]:SSCLocking:PROFile[?]

Syntax

[:SOURce]:SSCLocking:PROFile ‘identifier’, <TRIangular|ARBitrary>

[:SOURce]:SSCLocking:PROFile? ‘identifier’

Input Parameters

‘identifier’: 'M1.ClkGen'

<TRIangular|ARBitrary>: Set spread spectrum clocking profile.

Return Range

TRI|ARB

Description

This command sets up the spread spectrum clocking (SSC) using a triangular or arbitrary profile.

This SCPI is only for M8041A.

It is also applicable for M8195A but only in query mode.

Example

:SSCL:PROF ‘M1.ClkGen’,TRI

[:SOURce]:SSCLocking:SHAPe[?]

Syntax: [:SOURce]:SSCLocking:SHAPe 'identifier', <"Filename">

[:SOURce]:SSCLocking:SHAPe? 'identifier'

Input Parameters: 'identifier': 'M1.ClkGen'

<"Filename">: Set up spread spectrum clocking shape specified in "Filename".

Return Range: "Filename"

Description: This command sets up the spread spectrum clocking (SSC) using the specified shape in "Filename". A simple text file contains the data points which define the arbitrary waveform of the SSC profile.

This SCPI is only for M8041A.

Example: :SSCL:SHAP 'M1.ClkGen',"Factory/Sinusoidal.txt"

[:SOURce]:SSCLocking:FREQuency[?]

Syntax: [:SOURce]:SSCLocking:FREQuency 'identifier', <NRf>

[:SOURce]:SSCLocking:FREQuency? 'identifier'

Input Parameters: 'identifier': 'M1.ClkGen'

<NRf>: Set spread spectrum clocking frequency.

Return Range: 100 Hz to 200 kHz

Description: This command sets the spread spectrum clocking (SSC) frequency in hertz. Acceptable units include Hz, kHz and exponents (for example, 1E3 is the same as 1 kHz).

This SCPI is only for M8041A and M8195A.

Example: :SSCL:FREQ 'M1.ClkGen', 1khz

[:SOURce]:INTerference Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└─:INTerference
   ├─[:LEVel]
   │  ├─:HFRequency[:SOURce]
   │  ├─:LFRequency[:SOURce]
   │  ├─:CMODe
   │  │  ├─[:STATe][?]
   │  │  ├─:AMPLitude[?]
   │  │  ├─:SOURce[?]
   │  │  └─:GAIN[?]
   │  └─:DMODe
   │     ├─[:STATe][?]
   │     ├─:AMPLitude[?]
   │     ├─:SOURce[?]
   │     └─:GAIN[?]
   └─:ISYMbol
      ├─[:STATe][?]
      ├─:MODE[?]
      ├─:FREQuency{1:2}[?]
      ├─:ILOSs{1:2}[?]
      └─:PRESet[?]
```

This subnode has the following commands:

Table 45

Name	Description under
[:LEVel]:HFRequency[:SOURce][?]	[:SOURce]:INTerference[:LEVel]:HFRequency[:SOURce][?] on page 189
[:LEVel]:LFRequency[:SOURce][?]	[:SOURce]:INTerference[:LEVel]:LFRequency[:SOURce][?] on page 189
[:LEVel]:CMODe[:STATe][?]	[:SOURce]:INTerference[:LEVel]:CMODe[:STATe][?] on page 190
[:LEVel]:CMODe:AMPLitude[?]	[:SOURce]:INTerference[:LEVel]:CMODe:AMPLitude[?] on page 190
[:LEVel]:CMODe:SOURce[?]	[:SOURce]:INTerference[:LEVel]:CMODe:SOURce[?] on page 191
[:LEVel]:CMODe:GAIN[?]	[:SOURce]:INTerference[:LEVel]:CMODe:GAIN[?] on page 191
[:LEVel]:DMODe[:STATe][?]	[:SOURce]:INTerference[:LEVel]:DMODe[:STATe][?] on page 192
[:LEVel]:DMODe:AMPLitude[?]	[:SOURce]:INTerference[:LEVel]:DMODe:AMPLitude[?] on page 192
[:LEVel]:DMODe:SOURce[?]	[:SOURce]:INTerference[:LEVel]:DMODe:SOURce[?] on page 193
[:LEVel]:DMODe:GAIN[?]	[:SOURce]:INTerference[:LEVel]:DMODe:GAIN[?] on page 193
:ISYMbol[:STATe][?]	[:SOURce]:INTerference:ISYMbol[:STATe][?] on page 193
:ISYMbol:MODE[?]	[:SOURce]:INTerference:ISYMbol:MODE[?] on page 194
:ISYMbol:FREQuency{1:2}[?]	[:SOURce]:INTerference:ISYMbol:FREQuency{1:2}[?] on page 194
:ISYMbol:ILOSs{1:2}[?]	[:SOURce]:INTerference:ISYMbol:ILOSs{1:2}[?] on page 195
:ISYMbol:PRESet[?]	[:SOURce]:INTerference:ISYMbol:PRESet[?] on page 195

[:SOURce]:INTerference[:LEVel]:HFRequency[:SOURce][?]

Syntax	[:SOURce]:INTerference[:LEVel]:HFRequency[:SOURce] ‘identifier’, <NRf>
	[:SOURce]:INTerference[:LEVel]:HFRequency[:SOURce]? ‘identifier’
Input Parameters	‘identifier’: ‘M*.DATAOUT1’ or ‘M*.DataOut2’
	<NRf>: Set frequency of HF generator.
Return Range	1 GHz to 6 GHz
Description	This command sets the frequency of the high frequency generator. Acceptable units include MHz, GHz and exponents (for example, 2E9 is the same as 2 GHz).
	This SCPI is only for M8041A and M8051A.
Example	:INT:HFR ‘M2.DataOut1’, 2.0e9

[:SOURce]:INTerference[:LEVel]:LFRequency[:SOURce][?]

Syntax	[:SOURce]:INTerference[:LEVel]:LFRequency[:SOURce] ‘identifier’, <NRf>
	[:SOURce]:INTerference[:LEVel]:LFRequency[:SOURce]? ‘identifier’
Input Parameters	‘identifier’: ‘M*.DATAOUT1’ or ‘M*.DataOut2’
	<NRf>: Set frequency of LF generator.
Return Range	100 Hz to 1 GHz
Description	This command sets the frequency of the low frequency generator. Acceptable units include Hz, kHz, MHz, GHz and exponents (for example, 1E9 is the same as 1 GHz).
	This SCPI is only for M8041A and M8051A.
Example	:INT:LFR ‘M2.DataOut1’, 1.0e9

[:SOURce]:INTerference[:LEVel]:CMODe[:STATe][?]

Syntax	[:SOURce]:INTerference[:LEVel]:CMODe[:STATe] 'identifier', <0|1|ON|OFF> [:SOURce]:INTerference[:LEVel]:CMODe[:STATe]? 'identifier'
Input Parameters	'identifier': 'M*.DATAOUT1', 'M*.DataOut2', or 'M*.DataOut' <0|1|ON|OFF>: Turn common mode interference on/off.
Return Range	0|1
Description	This command turns common mode interference on/off. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:INT:CMOD 'M2.DataOut1', on

[:SOURce]:INTerference[:LEVel]:CMODe:AMPLitude[?]

Syntax	[:SOURce]:INTerference[:LEVel]:CMODe:AMPLitude 'identifier', <NRf> [:SOURce]:INTerference[:LEVel]:CMODe:AMPLitude? 'identifier'
Input Parameters	'identifier': 'M*.DATAOUT1' or 'M*.DataOut2' <NRf>: Set common mode interference amplitude.
Return Range	0 mV to 320 mV
Description	This command sets the amplitude of the common mode interference in volts. Acceptable units include mV, V and exponents (for example, 100E-3 is the same as 100 mV). This SCPI is only for M8041A and M8051A.
Example	:INT:CMOD:AMPL 'M2.DataOut1', 100mV

[:SOURce]:INTerference[:LEVel]:CMODe:SOURce[?]

Syntax	[:SOURce]:INTerference[:LEVel]:CMODe:SOURce ‘identifier’, <HFRequency|LFRequency> [:SOURce]:INTerference[:LEVel]:CMODe:SOURce? ‘identifier’
Input Parameters	‘identifier’: ‘M*.DATAOUT1’ or ‘M*.DataOut2’ <HFRequency|LFRequency>: Select HF or LF generator.
Return Range	HFR|LFR
Description	This command selects the high frequency or low frequency generator type for common mode interference. This SCPI is only for M8041A and M8051A.
Example	:INT:CMOD:SOUR ‘M2.DataOut1’, HFR

[:SOURce]:INTerference[:LEVel]:CMODe:GAIN[?]

Syntax	[:SOURce]:INTerference[:LEVel]:CMODe:GAIN ‘identifier’, <NRf> [:SOURce]:INTerference[:LEVel]:CMODe:GAIN? ‘identifier’
Input Parameters	‘identifier’: ‘M*.DATAOUT1’ or ‘M*.DataOut2’ <NRf>: Set common mode interference gain.
Return Range	0.1 to 1.0
Description	This command sets the linear gain of the common mode interference. This command is only for M8061A and M8062A.
Example	:INT:CMOD:GAIN ‘M2.DataOut1’, 0.5

[:SOURce]:INTerference[:LEVel]:DMODe[:STATe][?]

Syntax	[:SOURce]:INTerference[:LEVel]:DMODe[:STATe] 'identifier', <ON|OFF>
	[:SOURce]:INTerference[:LEVel]:DMODe[:STATe]? 'identifier'
Input Parameters	'identifier': 'M*.DATAOUT', 'M*.DATAOUT1', or 'M*.DataOut2'
	<ON|OFF>: Turn differential mode interference on/off.
Return Range	ON|OFF
Description	This command turns differential mode interference on/off.
	This SCPI is only for M8041A, M8051A, M8061A and M8062A.
Example	:INT:DMOD 'M2.DataOut1', on

[:SOURce]:INTerference[:LEVel]:DMODe:AMPLitude[?]

Syntax	[:SOURce]:INTerference[:LEVel]:DMODe:AMPLitude 'identifier', <NRf>
	[:SOURce]:INTerference[:LEVel]:DMODe:AMPLitude? 'identifier'
Input Parameters	'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'
	<NRf>: Set differential mode interference amplitude.
Return Range	0 mV to 360 mV
Description	This command sets the amplitude of the differential mode interference in volts. Acceptable units include mV, V and and exponents (for example, 100E-3 is the same as 100 mV).
	This SCPI is only for M8041A and M8051A.
Example	:INT:DMOD:AMPL 'M2.DataOut1', 100mV

[:SOURce]:INTerference[:LEVel]:DMODe:SOURce[?]

Syntax	[:SOURce]:INTerference[:LEVel]:DMODe:SOURce 'identifier', <HFRequency|LFRequency> [:SOURce]:INTerference[:LEVel]:DMODe:SOURce? 'identifier'
Input Parameters	'identifier': 'M*.DATAOUT1' or 'M*.DataOut2' <HFRequency|LFRequency>: Select HF or LF generator.
Return Range	HFR|LFR
Description	This command selects the high frequency or low frequency generator type for differential mode interference. This SCPI is only for M8041A and M8051A.
Example	:INT:DMOD:SOUR 'M2.DataOut1', LFR

[:SOURce]:INTerference[:LEVel]:DMODe:GAIN[?]

Syntax	[:SOURce]:INTerference[:LEVel]:DMODe:GAIN 'identifier', <NRf> [:SOURce]:INTerference[:LEVel]:DMODe:GAIN? 'identifier'
Input Parameters	'identifier': 'M*.DATAOUT1' or 'M*.DataOut2' <NRf>: Set differential mode interference gain.
Return Range	0.01 to 1.0
Description	This command sets the linear gain of the differential mode interference. This SCPI is for M8061A and M8062A.
Example	:INT:DMOD:GAIN 'M2.DataOut1', 0.5

[:SOURce]:INTerference:ISYMbol[:STATe][?]

Syntax	[:SOURce]:INTerference:ISYMbol[:STATe] 'Identifier', <0|1|ON|OFF> [:SOURce]:INTerference:ISYMbol[:STATe]? 'Identifier'
Input Parameters	Identifier: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
Return Range	0|1

Description: This command enables/disables the inter symbol interference state.

This SCPI is only for M8041A, M8051A and M8062A.

Example: :SOUR:INT:ISYM 'M2.DataOut1', ON

[:SOURce]:INTerference:ISYMbol:MODE[?]

Syntax: [:SOURce]:INTerference:ISYMbol:MODE 'Identifier', <PONE | PTWO>

[:SOURce]:INTerference:ISYMbol:MODE? 'Identifier'

Input Parameters: Identifier: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Description: This command allows the user to adjust inter symbol interface trace via one (no offset) or two point (variable offset).

Inter symbol interference mode must be set to one or two points; then only, the value of frequency and insertion loss parameter can be changed.

This SCPI is only for M8041A, M8051A and M8062A.

Example: :SOUR:INT:ISYM:MODE 'M2.DataOut1', PTWO

[:SOURce]:INTerference:ISYMbol:FREQuency{1:2}[?]

Syntax: [:SOURce]:INTerference:ISYMbol:FREQuency1 'Identifier', <NRf>

[:SOURce]:INTerference:ISYMbol:FREQuency2 'Identifier', <NRf>

[:SOURce]:INTerference:ISYMbol:FREQuency1? 'Identifier'

[:SOURce]:INTerference:ISYMbol:FREQuency2? 'Identifier'

Input Parameters: Identifier: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Description: This command sets one or two discrete user definable frequencies f1 and f2 of the inter symbol interference (ISI) trace. The mode has to be changed to two points in order to use the second frequency.

This SCPI is only for M8041A, M8051A and M8062A.

Range: 1GHz to 16GHz.

Example: :SOUR:INT:ISYM:FREQ1 'M2.DataOut1', 8.0e9

[:SOURce]:INTerference:ISYMbol:ILOSs{1:2}[?]

Syntax	[:SOURce]:INTerference:ISYMbol:ILOSs1 'Identifier', <NRf>
	[:SOURce]:INTerference:ISYMbol:ILOSs2 'Identifier', <NRf>
	[:SOURce]:INTerference:ISYMbol:ILOSs1? 'Identifier'
	[:SOURce]:INTerference:ISYMbol:ILOSs2? 'Identifier'
Input Parameters	Identifier: 'M*.DataOut', 'M*.DataOut1', or 'M*.DataOut2'
Description	This command sets the insertion loss at one or two discrete user definable frequencies, f1 and f2 of the inter symbol interference (ISI) trace.
	This SCPI is only for M8041A, M8051A, or M8062A.
Range	0.0dB to -25.0dB.
Example	:SOUR:INT:ISYM:ILOS1 'M2.DataOut', -9.8

[:SOURce]:INTerference:ISYMbol:PRESet[?]

Syntax	[:SOURce]:INTerference:ISYMbol:PRESet 'Identifier', <CUSTom | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 | P13 | P14 | P15 | P16 | P17>
	[:SOURce]:INTerference:ISYMbol:PRESet? 'Identifier'
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
Description	Selects an inter symbol interference (ISI) preset for the specific application.
	This command should ideally not be sent in one PM (SCPI - Programm Message) with following commands...:MODE; ...:FREQuency; ...:ILOSs.
	This SCPI is only for M8041A, M8051A and M8062A.
	For M8062A the presets are available up to P11 only; please refer to the Table 47 on page -197.

The following table describes the list of presets for selected standards and traces.

Table 46 Preset List for Selected Standards and Traces (For M8041A and M8051A)

			P1		P2	
		SCPI Name	f1/GHz	L1/dB	f2/GHz	L2/dB
PCIe3	Short	P1	1	-3.0	4	-9.6
PCIe3	Long	P2	1	-5.6	4	-17.9
M-PHY G3A	Ch1	P3	0	0.0	4.992	-6.0
M-PHY G3A	Ch2	P4	1.248	-4.1	4.992	-13.9
M-PHY G3B	Ch1	P5	0	0.0	5.83	-5.8
M-PHY G3B	Ch2	P6	1.458	-4.2	5.83	-13.8
M8048A	7.7”	P7	0	0.0	4	-4.4
M8048A	9.4”	P8	0	0.0	4	-6.0
M8048A	11.1”	P9	1	-1.9	4	-6.7
M8048A	12.8”	P10	1	-2.3	4	-7.1
M8048A	14.4”	P11	1	-2.7	4	-8.1
M8048A	16.1”	P12	1	-3.0	4	-9.0
M8048A	24.4	P13	1	-4.5	4	-13.8
SAS-3 12 Gb/s		P14	1	-4.5	4	-13.8
MIPI-Short		P15	0	0.0	5	-4.85
MIPI-Standard		P16	1.25	-3.75	5	-11.8
MIPI-Long		P17	1.25	-6.3	5	-20.0

.

Example :SOUR:INT:ISYM:PRES 'M2.DataOut1',P1

Table 47 Preset List for Selected Standards and Traces (For M8062A)

			P1		P2	
		SCPI Name	f1/GHz	L1/dB	f2/GHz	L2/dB
CEI-28G-VSR	Min	P1	14	-3.5	-	-
CEI-28G-VSR	Max	P2	14	-9	-	-
CEI-28G-VSR	100GbE	P3	12.89	-10.25	-	-
CEI-28G-VSR	32GFC	P4	14.025	-10.25	-	-
M8048A	7.7”	P5	4	-4.4	-	-
M8048A	9.4”	P6	4	-6	-	-
M8048A	11.1”	P7	1	-1.9	4	-6.7
M8048A	12.8”	P8	1	2.3	4	-7.1
M8048A	14.4”	P9	1	-2.7	4	-8.1
M8048A	16.1”	P10	1	-3	4	-9
M8048A	24.4”	P11	1	-4.5	4	-13.8

[:SOURce]:CONFigure Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└─:CONFigure
   └─:MINTegration[?]
```

This subnode has the following command:

[:SOURce]:CONFigure:MINTegration[?]

Syntax

[:SOURce]:CONFigure:MINTegration ‘identifier’,<NONe|MUX|DMUX|BOTH>[<data-location>[<visa-resource>]]

[:SOURce]:CONFigure:MINTegration? ‘identifier’

Input Parameters

‘identifier’: ‘M*.MuxMode’

<NONe|MUX|DMUX|BOTH>: Select mux/demux module.

<data-location>: (Optional) Specify location of data module. If omitted or an empty string is specified, the first suitable module is chosen automatically.

<visa-resource>: (Optional) Specify the address string of the N4877A.

Return Range

NONe|MUX|DMUX|BOTH

Description

This command controls the integration of a multiplexer and/or demultiplexer to a data module. Also, it controls the integration of the N4877A Clock Data Recovery and Demultiplexer.

In M8062A, the parameter values 'MUX', 'DMUX' and 'BOTH' all perform the same action i.e. they enable the 32G mode. 'NONe' disables 32G mode.

This SCPI is for M8061A and M8062A.

Example

:CONF:MINT ‘M2.MuxMode’, BOTH

OUTPut Subsystem

The OUTPut subsystem controls the output ports of the pattern generator.

This subsystem has the following SCPI structure:

```
:OUTPut
├─:GLOBal
│ └─[:STATe][?]
├─[:STATe][?]
├─:DRATe?
├─:POLarity[?]
├─:TVOLtage[?]
├─:TCONfig[?]
├─:COUPling[?]
├─:DIVider[?]
├─:FREQuency
│ └─:DIVider
├─:CALibration
│ └─:SKEW
├─:EIDLe
│ └─[:STATe][?]
├─:DATA
│ ├─:XOVer[?]
│ └─:F2Jitter[?]
├─:DEEMphasis
│ └─…
├─:EINSertion
│ └─…
├─:TRIGger
│ └─…
├─:DEMBedding
│ └─…
└─:EMBedding
  └─…
```

This subsystem has the following commands and subnodes:

Table 48

Name	Description under
:GLOBal[:STATe][?]	:OUTPut:GLOBal[:STATe][?] on page 201
[:STATe][?]	:OUTPut[:STATe][?] on page 201
:DRATe?	:OUTPut:DRATe? on page 201
:POLarity[?]	:OUTPut:POLarity[?] on page 202
:TVOLtage[?]	:OUTPut:TVOLtage[?] on page 202
:TCONfig[?]	:OUTPut:TCONfig[?] on page 203
:COUPling[?]	:OUTPut:COUPling[?] on page 203
:DIVider[?]	:OUTPut:DIVider[?] on page 204
:FREQuency[?]	:OUTPut:FREQuency[?] on page 204
:FREQuency:DIVider	:OUTPut:FREQuency:DIVider[?] on page 204
:CALibration:SKEW[?]	:OUTPut:CALibration:SKEW[?] on page 205
:EIDLe[:STATe][?]	:OUTPut:EIDLe[:STATe][?] on page 205
:DATA:XOVer[?]	:OUTPut:DATA:XOVer[?] on page 206
:DATA:F2Jitter[?]	:OUTPut:DATA:F2Jitter[?] on page 206
Subnodes	
:DEEMphasis	:OUTPut:DEEMphasis Subnode on page 207
:EINSertion	:OUTPut:EINSertion Subnode on page 213
:TRIGger	:OUTPut:TRIGger Subnode on page 217
:DEMBedding	:OUTPut:DEMBedding Subnode on page 219
:EMBedding	:OUTPut:EMBedding Subnode on page 222

:OUTPut:GLOBal[:STATe][?]

Syntax	:OUTPut:GLOBal[:STATe] ‘identifier’, <ON|OFF|1|0>
	:OUTPut:GLOBal][:STATe]? ‘identifier’
Input Parameters	‘identifier’: 'M1.System'
	<ON|OFF|1|0>: Enable/disable global state.
Return Range	1|0
Description	This command sets global output state to ON or OFF.
	This command works for all modules, as it is a Global command.
Example	:OUTP:GLOB ‘M1.System’, off

:OUTPut[:STATe][?]

Syntax	:OUTPut[:STATe] ‘identifier’, <ON|OFF|1|0>
	:OUTPut[:STATe]? ‘identifier’
Input Parameters	‘identifier’: 'M1.ClkOut', 'M1.TrigOut', 'M1.SysOutA' or 'M1.SysOutB', 'M*.DataOut1', 'M*.DataOut2', 'M*.CtrlOut', ‘M*.CtrlOutA’, or ‘M*.DataOut’
	<ON|OFF|1|0>: Switch an output on or off.
Return Range	1|0
Description	This command switches an output on or off.
	This SCPI is for M8041A, M8051A, M8061A, M8062A and M8195A.
Example	:OUTP ‘M1.DataOut1’, off

:OUTPut:DRATe?

Syntax	:OUTPut:DRATe? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut', or ‘M1.ClkOut’
Description	This query returns data rate for the output channel.
	This SCPI is for M8041A, M8051A, M8061A, M8062A and M8195A.
Example	:OUTP:DRAT? 'M1.DataOut1'

:OUTPut:POLarity[?]

Syntax :OUTPut:POLarity ‘identifier’, <NORMal|INVerted>

:OUTPut:POLarity? ‘identifier’

Input Parameters ‘identifier’: 'M1.TrigOut', 'M*.DataOut1', 'M*.DataOut2', 'M*.CtrlOutA', 'M*.SysOutA', 'M*.SysOutB', or 'M*.DataOut'

<NORmal|INVerted>: Set output polarity to normal or inverted.

Return Range NORM|INV

Description This command sets the output polarity to either normal or inverted.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example :OUTP:POL ‘M1.DataOut1’, INV

:OUTPut:TVOLtage[?]

Syntax :OUTPut:TVOLtage ‘identifier’, <NRf>

:OUTPut:TVOLtage? ‘identifier’

Input Parameters ‘identifier’: 'M1.ClkOut', 'M1.TrigOut', 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

<NRf>: Specify external termination voltage.

Return Range <NRf>

Description This command specifies the external termination voltage. The default is 0 V (ground). Acceptable units include mV, V and exponents (for example, 700E-3 is the same as 700 mV).

This SCPI is for M8041A, M8051A, M8061A, M8062A and M8195A.

Example :OUTP:TVOL ‘M1.DataOut1’, 700mV

:OUTPut:TCONfig[?]

Syntax	:OUTPut:TCONfig ‘identifier’, <BALanced|UNBalanced>
	:OUTPut:TCONfig? ‘identifier’
Input Parameters	‘identifier’: 'M1.ClkOut', 'M1.TrigOut', 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
	<BALanced|UNBalanced>: Specify balanced or unbalanced output.
Return Range	BAL|UNB
Description	This command selects the termination model for an output frontend:
	BALanced - balanced or differential output.
	UNBalanced - unbalanced with respect to ground (termination voltage).
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:OUTP:TCON ‘M2.DataOut2’, BAL

:OUTPut:COUPling[?]

Syntax	:OUTPut:COUPling ‘identifier’, <AC|DC>
	:OUTPut:COUPling? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
	<AC|DC>: Select output coupling.
Return Range	AC|DC
Description	This command selects DC or AC output coupling.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:OUTP:COUP ‘M1.DataOut1’, DC

:OUTPut:DIVider[?]

Syntax	:OUTPut:DIVider ‘identifier’, <DIV1...DIV80>
	:OUTPut:DIVider? ‘identifier’
Input Parameters	‘identifier’: 'M1.ClkOut'
	<DIV1...DIV80>: Specify a divider.
Return Range	DIV1 to DIV80
Description	This command selects a specific clock divider from DIV1 through DIV80.
	This SCPI is only for M8041A.
Example	:OUTP:DIV ‘M1.ClkOut’,DIV1

:OUTPut:FREQuency[?]

Syntax	:OUTPut:FREQuency ‘identifier’, <NRf>
	:OUTPut:FREQuency? ‘identifier’
Input Parameters	‘identifier’: ‘M1.RefClkOut’ or ‘M2.ClnClkOut’
Return Range	1e+7 |1e+8
Description	This command can be used to fetch or set the frequency.at RefClockOut.
	This SCPI is only for M8041A and for M8062A it is only available as a query.
Example	:OUTPut:FREQuency ‘m1.refclkout’, 1e+8.

:OUTPut:FREQuency:DIVider[?]

Syntax	:OUTPut:FREQuency:DIVider ‘identifier’, <NRf>
	:OUTPut:FREQuency:DIVider? ‘identifier’
Input Parameters	‘identifier’: 'M*.ClnClkOut’
Return Range	<NRf>
Description	This command divides the frequency at clean clock out (Cln Clk Out) on an M8062A module. This SCPI is only for M8062A.
Example	:OUTPut:FREQuency:DIVider ‘M2.ClnClkout’, 2

:OUTPut:CALibration:SKEW[?]

Syntax :OUTPut:CALibration:SKEW ‘identifier’, <NRf>

:OUTPut:CALibration:SKEW? ‘identifier’

Input Parameters ‘identifier’: 'M*.DataOut1 or M*.DataOut2'

Return Range -2ns to 2ns

Description This command is used as offset to Data Delay and Jitter Delay.

This SCPI is only for M8041A and M8051A.

Example :OUTPut:CALibration:SKEW ‘M1.DataOut1’, 0

:OUTPut:EIDLe[:STATe][?]

Syntax :OUTPut:EIDLe[:STATe] ‘identifier’, <OFF|IDLE|EXTernal>

:OUTPut:EIDLe[:STATe]? ‘identifier’

Input Parameters ‘identifier’: 'M*.DataOut'

<OFF>: Electrical idle feature is disabled.

<IDLE>: Electrical idle feature is enabled (software controlled, power saving mode).

<EXTernal>: Electrical idle input can be used to provide an electrical idle signal at the Data Output. An external generated signal, for example by a pulse generator at the electrical idle input, controls the output stream at the Data Output.

Return Range OFF|IDLE|EXT

Description This command enables/disables the electrical idle feature for the Data Output. The state can be OFF, IDLE or EXTernal.

This SCPI is only for M8061A and M8062A.

Example :OUTP:EIDL ‘M2.DataOut’, EXT

:OUTPut:DATA:XOVer[?]

Syntax	:OUTPut:DATA:XOVer ‘identifier’, <NRf> :OUTPut:DATA:XOVer? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut' <NRf>: Specify an eye crossover.
Return Range	For M8041A and M8051A: 10% to 90% For M8061A and M8062A: 30% to 70%
Description	This command sets the eye crossover of the transmitter signal at the DATA OUT/-DATA OUT connectors. Crossover is defined for NRZ signals. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:OUTP:DATA:XOV ‘M1.DataOut2’, 25

:OUTPut:DATA:F2Jitter[?]

Syntax	:OUTPut:DATA:F2Jitter ‘identifier’, <NRf> :OUTPut:DATA:F2Jitter? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut' <NRf>: Set the f/2 jitter.
Return Range	-20 ps to +20 ps
Description	This command sets the f/2 jitter at the output in seconds. Acceptable units include ps and exponents (for example, -10e-12 is the same as -10 ps). This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:OUTP:DATA:F2J ‘M1.DataOut2’, -10e-12

:OUTPut:DEEMphasis Subnode

The M8070A provides an eight tap de-emphasis signal. Seven of these taps can be specified: two pre-cursors and five post-cursors. The resulting main cursor is calculated.

The taps can be programmed remotely in two different ways. Firstly, all the tap values can be set by one command. In this case all seven tap values have to be specified at once in a comma separated list of numbers. Total 31 preset registers are available. Secondly, a preset register can be determined as the current preset and the pre- and post-cursor commands work on this selected preset register. Only absolute values are accepted in both cases.

The sign of a tap value is defined by a separate command. If, for example, the sign of pre-cursor 2 is set to negative (-1), it will change the sign of every pre-cursor 2 in the 31 preset registers.

This subnode has the following SCPI structure:

```
:OUTPut
└─:DEEMphasis
  ├─:UNIT[?]
  ├─:PRESet[?]
  ├─:SELect
  │ └─:PRESet[?]
  ├─:PRECursor{1:2}[?]
  ├─:POSTcursor{1:5}[?]
  └─:CONFigure
    └─:SIGN[?]
```

This subnode has the following commands:

Table 49

Name	Description under
:UNIT[?]	:OUTPut:DEEMphasis:UNIT[?] on page 208
:PRESet[?]	:OUTPut:DEEMphasis:PRESet[?] on page 209
:SELect:PRESet[?]	:OUTPut:DEEMphasis:SELect:PRESet[?] on page 210
:PRECursor{1:2}[?]	:OUTPut:DEEMphasis:PRECursor(*)[?] on page 211
:POSTcursor{1:5}[?]	:OUTPut:DEEMphasis:POSTcursor(*)[?] on page 211
:CONFigure:SIGN[?]	:OUTPut:DEEMphasis:CONFigure:SIGN(*)[?] on page 212

:OUTPut:DEEMphasis:UNIT[?]

Syntax

:OUTPut:DEEMphasis:UNIT ‘identifier’, <DB|PCT>

:OUTPut:DEEMphasis:UNIT? ‘identifier’

Input Parameters

‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

<DB|PCT>: Select dB or percent as the unit.

Return Range

DB|PCT

Description

This command defines the tap value unit: dB or PCT (PerCenT). This command should not be sent in one PM (SCPI - Program Message) with the following de-emphasis commands:

- :PRESet
- :PRECursor
- :POSTcursor

In other words, the unit should be set before sending the values with the above de-emphasis commands.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example

:OUTP:DEEM:UNIT ‘M1.DataOut2’, dB

:OUTPut:DEEMphasis:PRESet[?]

Syntax

:OUTPut:DEEMphasis:PRESet ‘identifier’,
<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>

:OUTPut:DEEMphasis:PRESet? ‘identifier’, <Preset Register No.>

Input Parameters

‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'

<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>:

Set register index and tap values.

<Preset Register No.>: Specify preset register number.

Return Range

<Pre2>,<Pre1>,<Post1>,<Post2>,< Post3>,< Post4>,

< Post5>

Description

This command is used to enter a comma separated list representing a register index and tap values. The first number in the list specifies the index of the preset register. There are 31 preset registers available addressed by a register index of 0 up to 30. The values in the list specify the register index and taps in the following order:

<RegNr><Pre2>,<Pre1>,<Post1>,<Post2>,< Post3>,< Post4>,< Post5>

Supported preset values for M8041A and M8051A are 0 - 30 and for M8061A and M8062A, 0 is the only preset value.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example

:OUTP:DEEM:PRES ‘M1.DataOut2’,2,1.1,2.2,4.3,6.7,5.3,6,3.3

:OUTPut:DEEMphasis:SELect:PRESet[?]

Syntax :OUTPut:DEEMphasis:SELect:PRESet ‘identifier’, <NRf>

:OUTPut:DEEMphasis:SELect:PRESet? ‘identifier’

Input Parameters ‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'

<NRf>: Set current preset register.

Return Range 0 to 30

Description This command sets the current preset register. One in 31 registers can be defined as the ‘current’ preset register. The tap values of the ‘current’ preset register are shown at the output. Preset register 0 is the default register. The :PRECursor and :POSTCursor commands change the tap values only in that preselected preset register. There are 31 preset registers available addressed by a register index of 0 up to 30.

This command should not be sent in one PM (SCPI - Program Message) with the following de-emphasis commands:

- :PRESet
- :PRECursor
- :POSTcursor

In other words, the preset register should be selected before sending the values with the above de-emphasis commands.

Supported preset values for M8041A and M8051A are 0 - 30 and for M8061A and M8062A, 0 is the only preset value.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example :OUTP:DEEM:SEL:PRES ‘M1.DataOut2’, 4

:OUTPut:DEEMphasis:PRECursor(*)[?]

Syntax	:OUTPut:DEEMphasis:PRECursor(*) ‘identifier’, <NRf>
	:OUTPut:DEEMphasis:PRECursor(*)? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'
	<NRf>: Set selected pre-cursor value.
Return Range	PRECursor1: -12.04 dB to 12.04 dB | 25% to 400%
	PRECursor2: -6.02 dB to 6.02 dB | 50% to 200%
Description	This command programs the pre-cursor value of the currently selected pre-cursor (pre-cursor 1 or pre-cursor 2).
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:OUTP:DEEM:PREC2 ‘M1.DataOut2’, 3.1

:OUTPut:DEEMphasis:POSTcursor(*)[?]

Syntax	:OUTPut:DEEMphasis:POSTcursor(*) ‘identifier’, <NRf>
	:OUTPut:DEEMphasis:POSTcursor(*)? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'
	<NRf>: Set selected post-cursor value.
Return Range	POSTcursor1: -20 dB to 20 dB | 10% to 1000%
	POSTcursor2, POSTcursor3: -12.04 dB to 12.04 dB | 25% to 400%
	POSTcursor4, POSTcursor5: -6.02 dB to 6.02 dB | 50% to 200%
Description	This command programs the post-cursor value of the currently selected post-cursor (post-cursor 1 through post-cursor 5).
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:OUTP:DEEM:POST4 ‘M1.DataOut2’, 2.3

:OUTPut:DEEMphasis:CONFigure:SIGN(*)[?]

Syntax

:OUTPut:DEEMphasis:CONFigure:SIGN(*) 'identifier', <enum>, <enum>, <enum>, <enum>, <enum>, <enum>, <enum>

:OUTPut:DEEMphasis:CONFigure:SIGN(*)? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2'

<enum>: Specify positive/negative tap value sign for each tap.

Return Range

POS|NEG (returned for each tap)

Description

A list of "POSitive"s and "NEGative"s specify the sign of the corresponding tap.

The following list illustrates the meaning of the comma separated list:

<Pre2>,<Pre1>,<Post1>,<Post2>,<Post3>,<-Post4>,<Post5>

The order of the values corresponds to the order in the :PRESet command argument list. So sending the following comma separated list "pos,neg,pos,neg,neg,neg,pos" determines that the pre-cursor 2 is a POSitive value, pre-cursor 1 is a NEGative value, post cursor 1 is a POSitive value and so on. This command influences all presets at once. For example, it is not possible to have a post cursor 2 with different signs in other preset registers.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example

:OUTP:DEEM:CONF:SIGN 'M1.DataOut2',pos,pos,neg,neg,pos,pos,pos

NOTE

The sign (+ve or -ve) for de-emphasis values (pre-cursor or post-cursor) is set using the command ":OUTPut:DEEMphasis:CONFigure:SIGN(*)". However, the same can now be directly programmed by using signed values with the commands ":OUTPut:DEEMphasis:PRECursor(*)", or ":OUTPut:DEEMphasis:POSTcursor(*)".

This SCPI is only supported by M8041A/M8051A (Hardware Revision 1) and M8061A modules.

:OUTPut:EINSertion Subnode

This subnode has the following SCPI structure:

```
:OUTPut
└:EINSertion
  ├[:STATe][?]
  ├:RATio[?]
  ├:MODE[?]
  ├:ONCE
  └:DISParity
```

This subnode has the following commands:

Table 50

Name	Description under
[:STATe][?]	:OUTPut:EINSertion[:STATe][?] on page 214
:RATio[?]	:OUTPut:EINSertion:RATio[?] on page 214
:MODE[?]	:OUTPut:EINSertion:MODE[?] on page 215
:ONCE	:OUTPut:EINSertion:ONCE on page 215
:DISParity	:OUTPut:EINSertion:DISParity on page 216

:OUTPut:EINSertion[:STATe][?]

Syntax	:OUTPut:EINSertion[:STATe] ‘identifier’, <ON|OFF|1|0> :OUTPut:EINSertion[:STATe]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut' <ON|OFF|1|0>: Switch error insertion output on or off.
Return Range	1|0
Description	This command switches the error insertion feature on or off for the specified output. This SCPI is only for M8041A and M8051A. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:OUTP:EINS ‘M1.DataOut1’, on

:OUTPut:EINSertion:RATio[?]

Syntax	:OUTPut:EINSertion:RATio ‘identifier’, <RM1|...|RM12> :OUTPut:EINSertion:RATio? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut' <RM1|...|RM12>: Set the error insertion ratio.
Return Range	<RM1|...|RM12>
Description	This command controls the ratio of the internal fixed error insertion ratio. The range is 1E-12 (RM12) to 1E-1 (RM1). This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:OUTP:EINS:RAT ‘M1.DataOut1’, RM6

:OUTPut:EINSertion:MODE[?]

Syntax :OUTPut:EINSertion:MODE ‘identifier’, <ERFSpacing|...|SINB>

:OUTPut:EINSertion:MODE? ‘identifier’

Input Parameters ‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

<ERFSpacing|...|SINB>: Specify the error insertion mode.

Return Range <ERFSpacing|...|SINB>

Description This command is used to specify the error insertion mode as follows:

ERFSpacing: Generates errors with a predetermined bit error rate (fixed spacing).

ERATio: Generates errors with a predetermined bit error rate. Erroneous bits are randomly distributed.

CINA: Specifies CTRL IN A as a trigger.

CINB: Specifies CTRL IN B as a trigger.

BRK: Specifies software break command as a trigger.

SINA: Specifies SYS IN A as a trigger source.

SINB: Specifies SYS IN B as a trigger source.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example :OUTP:EINS:MODE ‘M1.DataOut1’, ERFSpacing

:OUTPut:EINSertion:ONCE

Syntax :OUTPut:EINSertion:ONCE ‘identifier’

Input Parameters ‘identifier’: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Description This command generates a single error (single bit error) on the specified output.

This SCPI is for M8041A, M8051A, M8061 and M8062A.

Example :OUTP:EINS:ONCE ‘M1.DataOut1’

:OUTPut:EINSertion:DISParity

Syntax	:OUTPut:EINSertion:DISParity ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut1' or 'M*.DataOut2'
Description	This command generates a single disparity error (only valid for 8b / 10b operating mode) on the specified output. This SCPI is only for M8041A and M8051A.
Example	:OUTP:EINS:DISP ‘M1.DataOut2’

:OUTPut:TRIGger Subnode

This subnode has the following SCPI structure:

```
:OUTPut
└:TRIGger
  ├:MODE[?]
  └:DIVider[?]
```

This subnode has the following commands:

Table 51

Name	Description under
:MODE[?]	:OUTPut:TRIGger:MODE[?] on page 218
:DIVider[?]	:OUTPut:TRIGger:DIVider[?] on page 218

:OUTPut:TRIGger:MODE[?]

Syntax	:OUTPut:TRIGger:MODE ‘identifier’, <CLOCk|SEQuencer> :OUTPut:TRIGger:MODE? ‘identifier’
Input Parameters	‘identifier’: 'M1.TrigOut' <CLOCk|SEQuencer>: Set the operating mode of the trigger out.
Return Range	CLOC|SEQ
Description	This command sets the operating mode of the trigger out (TRIG OUT) either to subrate clock or sequencer controlled mode. This SCPI is only for M8041A.
Example	:OUTP:TRIG:MODE ‘M1.TrigOut’, CLOC

:OUTPut:TRIGger:DIVider[?]

Syntax	:OUTPut:TRIGger:DIVider ‘identifier’, <NRf> :OUTPut:TRIGger:DIVider? ‘identifier’
Input Parameters	‘identifier’: 'M1.TrigOut' <NRf>: Set the divider factor for the trigger out.
Return Range	2 to 65535
Description	This command sets the divider factor of the trigger out when the subrate clock operating mode is selected. This SCPI is only for M8041A.
Example	:OUTP:TRIG:DIV ‘M1.TrigOut’, 8

:OUTPut:DEMBedding Subnode

This subnode has the following SCPI structure:

```
:OUTPut
└:DEMBedding
 ├[:SCABle]
 │ └[:STATe][?]
 └:SPARameter
   ├ [:STATe][?]
   ├ :PROFile[?]
   ├ :IPORt[?]
   └ :OPORt[?]
```

This subnode has the following commands:

Table 52

Name	Description under
:SCABle:STATe[?]	:OUTPut:DEMBedding:SCABle:STATe[?] on page 220
SPARameter:STATe[?]	:OUTPut:DEMBedding:SPARameter:STATe[?] on page 220
:SPARameter:PROFile[?]	:OUTPut:DEMBedding:SPARameter:PROFile[?] on page 220
:SPARameter:IPORt[?]	:OUTPut:DEMBedding:SPARameter:IPORt[?] on page 221
:SPARameter:OPORt[?]	:OUTPut:DEMBedding:SPARameter:OPORt[?] on page 221

:OUTPut:DEMBedding:SCABle:STATe[?]

Syntax :OUTPut:DEMBedding[:SCABle][:STATe] ‘identifier’, <OFF | ON | 0 | 1>

:OUTPut:DEMBedding[:SCABle][:STATe]? ‘identifier’

Input Parameters ‘identifier’:'M*.DataOut1’,‘M*.DataOut2’,‘M*.DataOut3’,or ‘M*.DataOut4’
OFF | ON | 0 | 1: Enables/disables standard cable compensation.

Return Range 0|1

Description This command switches on/off the standard cable compensation on the specified output.

The query return the present setting.

This SCPI is for M8195A.

Example :OUTP:DEMB:SCAB:STAT ‘M1.DataOut1’, ON

:OUTPut:DEMBedding:SPARameter:STATe[?]

Syntax :OUTPut:DEMBedding:SPARameter[:STATe] ‘identifier’, <OFF | ON | 0 | 1>

:OUTPut:DEMBedding:SPARameter[:STATe]? ‘identifier’

Input Parameters ‘identifier’:'M*.DataOut1’,‘M*.DataOut2’,‘M*.DataOut3’,or ‘M*.DataOut4’
<OFF | ON | 0 | 1>

Description This command switches on/off the compensation via S-parameter file on the specified output.

The query return the present setting.

This SCPI is for M8195A.

Example :OUTP:DEMB:SPAR:STAT ‘M1.DataOut1’, ON

:OUTPut:DEMBedding:SPARameter:PROFile[?]

Syntax :OUTPut:DEMBedding:SPARameter:PROFile ‘identifier’, <”Filename”>

:OUTPut:DEMBedding:SPARameter:PROFile? ‘identifier’

Input Parameters ‘identifier’:'M*.DataOut1’,‘M*.DataOut2’,‘M*.DataOut3’,or ‘M*.DataOut4’

Description This command selects the S-Parameter profile. S-parameter is defined by frequency/real/imaginary tuples for 1 to 9 port components.

The query return the present setting.

This SCPI is for M8195A.

Example :OUTP:DEMB:SPAR:PROF ‘M1.DataOut1’, ”factory/huber_suhner_cable.s2p”

:OUTPut:DEMBedding:SPARameter:IPORt[?]

Syntax :OUTPut:DEMBedding:SPARameter:IPORt ‘identifier’, <NRf>

:OUTPut:DEMBedding:SPARameter:IPORt? ‘identifier’

Input Parameters ‘identifier’:'M*.DataOut1’,‘M*.DataOut2’,‘M*.DataOut3’,or ‘M*.DataOut4’ 1|2|3|4:

Return Range 1|2|3|4; depends upon the selected S parameter file.

Description This command selects the input port in the S-Parameter profile.

The query return the present setting.

This SCPI is for M8195A.

Example :OUTP:DEMB:SPAR:IPOR ‘M1.DataOut1’, 1

:OUTPut:DEMBedding:SPARameter:OPORt[?]

Syntax :OUTPut:DEMBedding:SPARameter:OPORt ‘identifier’, <NRf>

:OUTPut:DEMBedding:SPARameter:OPORt? ‘identifier’

Input Parameters ‘identifier’:'M*.DataOut1’,‘M*.DataOut2’,‘M*.DataOut3’,or ‘M*.DataOut4’ 1|2|3|4:

Return Range 1|2|3|4; depends upon the selected S-parameter file.

Description This command selects the output port in the S-Parameter profile.

The query return the present setting.

This SCPI is for M8195A.

Example :OUTP:DEMB:SPAR:OPOR ‘M1.DataOut1’, 2

:OUTPut:EMBedding Subnode

This subnode has the following SCPI structure:

```
:OUTPut
└:EMBedding
  └:SPARameter
    ├ [:STATe][?]
    ├ :PROFile[?]
    ├ :IPORt[?]
    └ :OPORt[?]
```

This subnode has the following commands:

Table 53

Name	Description under
:SPARameter[:STATe][?]	:OUTPut:EMBedding:SPARameter[:STATe][?] on page 223
:SPARameter:PROFile[?]	:OUTPut:EMBedding:SPARameter:PROFile[?] on page 223
:SPARameter:IPORt[?]	:OUTPut:EMBedding:SPARameter:IPORt[?] on page 223
:SPARameter:OPORt[?]	:OUTPut:EMBedding:SPARameter:OPORt[?] on page 224

:OUTPut:EMBedding:SPARameter[:STATe][?]

Syntax	:OUTPut:EMBedding:SPARameter[:STATe] ‘identifier’, <OFF | ON | 0 | 1>
	:OUTPut:EMBedding:SPARameter[:STATe]? ‘identifier’
Input Parameters	‘identifier’:'M*.DataOut1’,‘M*.DataOut2’,‘M*.DataOut3’,or ‘M*.DataOut4’ OFF | ON | 0 | 1
Return Range	0|1
Description	This command Switches on/off the compensation via S-parameter file on the specified output.
	The query returns the present setting.
	This SCPI is for M8195A.
Example	:OUTP:EMB:SPAR:STAT ‘M1.DataOut1’, ON

:OUTPut:EMBedding:SPARameter:PROFile[?]

Syntax	:OUTPut:EMBedding:SPARameter:PROFile ‘identifier’, <”Filename”>
	:OUTPut:EMBedding:SPARameter:PROFile? ‘identifier’
Input Parameters	‘identifier’:'M*.DataOut1’,‘M*.DataOut2’,‘M*.DataOut3’,or ‘M*.DataOut4’
Description	This command selects the S-Parameter profile. S-parameter is defined by frequency/real/imaginary tuples for 1 to 9 port components.
	The query returns the present setting.
	This SCPI is for M8195A.
Example	OUTP:EMB:SPAR:PROF ‘M1.DataOut1’, ”factory/huber_suhner_cable.s2p”

:OUTPut:EMBedding:SPARameter:IPORt[?]

Syntax	:OUTPut:EMBedding:SPARameter:IPORt ‘identifier’, <NRf>
	:OUTPut:EMBedding:SPARameter:IPORt? ‘identifier’
Input Parameters	‘identifier’:'M*.DataOut1’,‘M*.DataOut2’,‘M*.DataOut3’,or ‘M*.DataOut4’
Description	This command selects the input port in the S-Parameter profile.
	The query returns the present setting.

This SCPI is for M8195A.

Example :OUTP:EMB:SPAR:IPOR ‘M1.DataOut1’, 1

:OUTPut:EMBedding:SPARameter:OPORt[?]

Syntax :OUTPut:EMBedding:SPARameter:OPORt ‘identifier’, <NRf>

:OUTPut:EMBedding:SPARameter:OPORt ‘identifier’

Input Parameters ‘identifier’:'M*.DataOut1’,‘M*.DataOut2’,‘M*.DataOut3’,or ‘M*.DataOut4’

Description This command selects the output port in the S-Parameter profile.

The query returns the present setting.

This SCPI is for M8195A.

Example OUTP:EMB:SPAR:OPOR ‘M1.DataOut1’, 2

INPut Subsystem

The INPut subsystem controls the input ports of the analyzer.

This subsystem has the following SCPI structure:

```
:INPut
├─:BRM[?]
├─:DRATe?
├─:DELay[?]
├─:EQUalization
│ └─[:PRESet][?]
├─:POLarity[?]
├─:CMVoltage[?]
│ └─:DETect
├─:THReshold[?]
├─:TVOLtage[?]
├─:TCONfig[?]
├─:CMODe[?]
├─:COUPling[?]
├─:SENSitivity[?]
├─:CDR
│ └─…
└─:ALIGnment
  └─...
```

The subsystem has the following commands and subnodes:

Table 54

Name	Description under
:BRM[?]	:INPut:DATA:BRMode[?] on page 226
:DRATe?	:INPut:DRATe? on page 228
:DELay	:INPut:DELay[?] on page 228
:EQUalization[:PRESet]	:INPut:EQUalization[:PREset][?] on page 228
:POLarity	:INPut:POLarity[?] on page 230
:CMVoltage[?]	:INPut:CMVoltage[?] on page 230
:CMVoltage:DETect	:INPut:CMVoltage:DETect on page 230
:THReshold	:INPut:THReshold[?] on page 231
:TVOLtage	:INPut:TVOLtage[?] on page 231
:TCONfig[?]	:INPut:TCONfig[?] on page 232
:CMODe[?]	:INPut:CMODe[?] on page 232
COUPling[?]	:INPut:COUPling[?] on page 233
SENSitivity[?]	:INPut:SENSitivity[?] on page 233
Subnodes	
:CDR	:INPut:CDR Subnode on page 234
:ALIGnment	:INPut:ALIGnment Subnode on page 242

:INPut:DATA:BRMode[?]

Syntax :INPut:DATA:BRMode ‘identifier’, < 0 | 1 | ON | OFF>

:INPut:DATA:BRMode? ‘identifier’

Input Parameters ‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'

Range 0 | 1 | ON | OFF

Description This command enables/disables the bit recovery mode. In BRM, the error detector does not expect any specific data. This mode can be used if the incoming data is completely unknown.

NOTE **The bit recovery mode is only available with CDR.**

Example :INP:DATA:BRM 'M1.DataIn1',ON

:INPut:DRATe?

Syntax	:INPut:DRATe? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1', 'M*.DataIn2' or 'M*.DataIn'
Description	This query returns the data rate for the input channel. This SCPI is for M8061A and M8062A.
Example	:INP:DRAT? 'M1.DataIn1'

:INPut:DELay[?]

Syntax	:INPut:DELay ‘identifier’, <NRf> :INPut:DELay? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' <NRf>: Specify sampling edge delay value.
Return Range	For M8041A and M8051A: -6.7 ns to +6.7 ns For M8061A: 0 ps to 60 ps For M8062A: -10 ns to 10 ns
Description	This command specifies the delay of the sampling edge. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:INP:DEL ‘M1.DataIn1’, 1e-9

:INPut:EQUalization[:PREset][?]

Syntax	:INPut:EQUalization[:PREset] 'Identifier', <OFF | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10> :INPut:EQUalization[:PREset]? 'Identifier',
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' <OFF | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10>
Return Range	OFF | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10

Description

This command is used to select from the following equalization presets:

For M8041A and M8051A:

- OFF – Disables equalization
- P1 – PCIe 8Gb/s -6 dB
- P2 – PCIe 8Gb/s -9 dB
- P3 – PCIe 8Gb/s -12 dB
- P4 – USB 5Gb/s -3.5 dB
- P5 – PCIe 16Gb/s -6 dB
- P6 – PCIe 16Gb/s -9 dB
- P7 – PCIe 16Gb/s -12 dB
- P8 – USB 10Gb/s 0 dB
- P9 – USB 10Gb/s -3 dB
- P10 – USB 10Gb/s -6 dB

For M8061A:

- OFF – off
- P1 – PCIe 8Gb/s -6 dB
- P2 – PCIe 8Gb/s -9 dB
- P3 – PCIe 8Gb/s -12 dB
- P4 – USB 5Gb/s -3.5 dB

For M8062A:

- P1 – Low
- P2 – Medium
- P3 – High

The availability of individual presets depends on the type of module being controlled and the calibration status of the respective module.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

For M8062A, the equalizer is available only for 20 Gb/s and higher. For other values, if a command is sent, it will return OFF and will notify that the value is read-only.

Example

:INP:EQU ‘M1.DataIn1’,P1

:INPut:POLarity[?]

Syntax	:INPut:POLarity ‘identifier’, <NORMal|INVerted> :INPut:POLarity? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1', 'M*.DataIn2', or 'M*.DataIn' <NORMal|INVerted>: Set input polarity.
Return Range	NORM|INV
Description	This command sets the input polarity of the specified input to either NORMal or INVerted. This SCPI is only for M8041A, M8051A, M8061A and M8062A.
Example	:INP:POL ‘M1.DataIn1’, INV

:INPut:CMVoltage[?]

Syntax	:INPut:CMVoltage ‘identifier’, <NRf> :INPut:CMVoltage? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' <NRf>: Set common mode voltage.
Return Range	-1.0 V to 2.6 V; The range is dependent on other system parameters. See Command Syntax to Find Min/Max Values on page 111.
Description	This command is used to measure the common mode voltage of the selected data input by the module itself. Acceptable units are mV, V and exponents (for example, 300E-3 is the same as 300 mV). This SCPI is only for M8041A and M8051A.
Example	:INP:CMV ‘M1.DataIn1’, 300mV

:INPut:CMVoltage:DETect

Syntax	:INPut:CMVoltage:DETect ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
Description	This command measures the common mode voltage at the selected data input and applies it to the comparator. This SCPI is only for M8041A and M8051A.

Example :INP:CMV:DET ‘M1.DataIn1’

:INPut:THReshold[?]

Syntax	:INPut:THReshold ‘identifier’, <NRf>
	:INPut:THReshold? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1', 'M*.DataIn2', or 'M2.ElectIdleIn'
	<NRf>: Set receiver input threshold voltage.
Return Range	For M8041A and M8051A: -1 V to 1 V
	For M8062A: -800 mV to 800 mV
Description	This command sets the threshold voltage of the selected receiver input connector (decision threshold). Acceptable units are mV, V and exponents (for example, 300E-3 is the same as 300 mV).
	This SCPI is only for M8041A, M8051A and M8062A.
Example	:INP:THR ‘M1.DataIn1’, 300 mV

:INPut:TVOLtage[?]

Syntax	:INPut:TVOLtage ‘identifier’, <NRf>
	:INPut:TVOLtage? ‘identifier’
Input Parameters	‘identifier’:'M*.DataIn1', 'M*.DataIn2', or 'M2.ElectIdleIn'
	<NRf>: Specify receiver input termination voltage.
Return Range	-1 V to 2.9 V; The range is dependent on other system parameters. See Command Syntax to Find Min/Max Values on page 111.
Description	This command specifies the termination voltage of the selected receiver input connector. Acceptable units are mV, V and exponents (for example, 300E-3 is the same as 300 mV).
	This SCPI is only for M8041A, M8051A and M8062A.
Example	:INP:TVOL ‘M1.DataIn1’, 300mV

:INPut:TCONfig[?]

Syntax	:INPut:TCONfig 'identifier', <BALanced|UNBalanced> :INPut:TCONfig? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1' or 'M*.DataIn2' <BALanced|UNBalanced>: Select input termination configuration.
Return Range	BAL|UNB
Description	This command selects the termination configuration for an input front end. The following options are available: **BALanced**: balanced or differential **UNBalanced**: unbalanced with respect to ground (termination voltage) This SCPI is only for M8041A and M8051A.
Example	:INP:TCON 'M1.DataIn1', BAL

:INPut:CMODe[?]

Syntax	:INPut:CMODe 'identifier', <SENormal|SEComplement|DIFFerential> :INPut:CMODe? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1' or 'M*.DataIn2' <SENormal|SEComplement|DIFFerential>: Select compare mode.
Return Range	SEN|SEC|DIFF
Description	This command defines the Compare MODe. This command defines which input (DATA or DATA\) is active. The following options are available: **SENormal**: Data stream is compared at Normal input; complement is terminated. **SEComplement**: Data stream is compared at Complement input; Normal is terminated. **DIFFerential**: If differential both input ports need to receive a signal; actual data signal is measured as the voltage difference between the two incoming signals. This SCPI is only for M8041A, M8051A and M8061A.
Example	:INP:CMOD 'M1.DataIn1', SEC

:INPut:COUPling[?]

Syntax	:INPut:COUPling ‘identifier’, <AC|DC>
	:INPut:COUPling? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
	<AC|DC>: Select input coupling.
Return Range	AC|DC
Description	This command selects DC or AC input coupling.
	This SCPI is only for M8041A and M8051A.
Example	:INP:COUP ‘M1.DataIn1’, DC

:INPut:SENSitivity[?]

Syntax	:INPut:SENSitivity ‘identifier’, <HIGH|NORMal>
	:INPut:SENSitivity? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
	<HIGH|NORMal>: Select input coupling.
Return Range	HIGH|NORM
Description	This command sets the error detector input sensitivity to high or normal.
	This SCPI is only for M8041A, M8051A and M8062A.
Example	:INP:SENS ‘M1.DataIn1’, HIGH

:INPut:CDR Subnode

This subnode has the following SCPI structure:

```
:INPut
└:CDR
  ├:LORDer[?]
  ├:CTRL[?]
  ├[:STATe][?]
  ├:AUTO[?]
  ├:RELOck
  ├:HIGHTD[?]
  ├:OPTimize
  ├:FIRSt
  │ ├:TDENsity[?]
  │ └:LBANdwidth[?]
  └:SECond
    ├:TDENsity[?]
    ├:LBANdwidth[?]
    └:PEAKing[?]
```

This subnode has the following commands:

Table 55

Name	Description under
:LORDer[?]	:INPut:CDR:LORDer[?] on page 236
:CTRL[?]	:INPut:CDR:CTRL[?] on page 236
[:STATe][?]	:INPut:CDR[:STATe][?] on page 237
:AUTO[?]	:INPut:CDR:AUTO[?] on page 237
:RELOck	:INPut:CDR:RELOck on page 238
:HIGHTD[?]	:INPut:CDR:HIGHTD[?] on page 238
OPTimize	:INPut:CDR:OPTimize on page 238
:FIRSt:TDENsity[?]	:INPut:CDR:FIRSt:TDENsity[?] on page 239
:FIRSt:LBANdwidth[?]	:INPut:CDR:FIRSt:LBANdwidth[?] on page 239

Name	Description under
:SECond:TDENsity[?]	:INPut:CDR:SECond:TDENsity[?] on page 240
:SECond:LBANdwidth[?]	:INPut:CDR:SECond:LBANdwidth[?] on page 240
:SECond:PEAKing[?]	:INPut:CDR:SECond:PEAKing[?] on page 241

:INPut:CDR:LORDer[?]

Syntax: :INPut:CDR:LORDer ‘identifier’, <FIRSt|SECond>

:INPut:CDR[:STATe]? ‘identifier’

Input Parameters: ‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'

<FIRSt|SECond>: Set the loop order of the CDR.

Return Range: FIRS|SEC

Description: This command determines the loop order of the CDR. Depending on this setting, defining the behavior of the CDR is done by different parameters. For first order only, transition density and bandwidth fully define its characteristics while for the second order this definition needs at least one more parameter i.e. peaking.

This SCPI is only for M8041A and M8051A.

Example: :INP:CDR:LORD ‘M1.DataIn1’, FIRS

:INPut:CDR:CTRL[?]

Syntax: :INPut:CDR:CTRL ‘identifier’, <MANual|SEQuence>

:INPut:CDR:CTRL? ‘identifier’

Input Parameters: ‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'

<MANual|SEQuence>: Set the CDR control method.

Return Range: MAN|SEQ

Description: This command selects whether the CDR is controlled manually or by the sequencer.

This SCPI is only for M8041A and M8051A.

Example: :INP:CDR:CTRL ‘M1.DataIn1’, SEQ

:INPut:CDR[:STATe][?]

Syntax	:INPut:CDR[:STATe] ‘identifier’, <ON|OFF|1|0>
	:INPut:CDR[:STATe]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
	<ON|OFF|1|0>: Enable/disable the CDR.
Return Range	1|0
Description	This command enables or disables CDR.
	This SCPI is only for M8041A and M8051A.
Example	:INP:CDR ‘M1.DataIn1’, ON

:INPut:CDR:AUTO[?]

Syntax	:INPut:CDR:AUTO ‘identifier’, <ON|OFF|1|0>
	:INPut:CDR:AUTO? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn'
	<ON|OFF|1|0>: Enable/disable the auto relock.
Return Range	1|0
Description	This command enables or disables the auto relock.
	This SCPI is only for M8062A.
Example	The following SCPI command sets the ‘auto relock’ to 1.
	:INPut:CDR:AUTO ‘M2.DataIn1’, 1
	The following SCPI command queries the ‘auto relock’ state.
	:INPut:CDR:AUTO? ‘M2.DataIn1’
	1

:INPut:CDR:RELOck

Syntax	:INPut:CDR:RELOck ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn'
Description	This command execute a full CDR relock. This SCPI is only for M8062A.
Example	:INPut:CDR:RELOck ‘M2.DataIn’

:INPut:CDR:HIGHTD[?]

Syntax	:INPut:CDR:HIGHTD ‘identifier’, <ON|OFF|1|0> :INPut:CDR:HIGHTD? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn' <ON|OFF|1|0>: Set the High Transition Density state to ON or OFF.
Return Range	0|1
Description	This command sets the High Transition Density state to ON or OFF. This SCPI is only for M8062A.
Example	:INPut:CDR:HIGHTD ‘M2.DataIn’, 1

:INPut:CDR:OPTimize

Syntax	:INPut:CDR:OPTimize ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn'
Description	This command invokes the M8062A CDR optimization at the current data rate and optionally applied jitter. This SCPI is only for M8062A.
Example	:INPut:CDR:OPTimize ‘M2.DataIn’

:INPut:CDR:FIRSt:TDENsity[?]

Syntax :INPut:CDR:FIRSt:TDENsity ‘identifier’, <NRf>

:INPut:CDR:FIRSt:TDENsity? ‘identifier’

Input Parameters ‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'

<NRf>: Set first order transition density.

Return Range 25% to 100%

Description This command is used to set the first order transition density of the incoming signal from 25% to 100%. The default is 50%.

This SCPI is only for M8041A and M8051A.

Example :INP:CDR:FIRS:TDEN ‘M1.DataIn1’, 25

:INPut:CDR:FIRSt:LBANdwidth[?]

Syntax :INPut:CDR:FIRSt:LBANdwidth ‘identifier’, <NRf>

:INPut:CDR:FIRSt:LBANdwidth? ‘identifier’

Input Parameters ‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'

<NRf>: Set first order loop bandwidth of the CDR.

Return Range 102 kHz to 20 MHz

Description This command is used to set the first order loop bandwidth of the CDR. The default is 2 MHz. Acceptable units are Hz, kHz, MHz and exponents (for example, 4E6 is the same as 4 MHz).

This SCPI is only for M8041A and M8051A.

Example :INP:CDR:FIRS:LBAN ‘M1.DataIn1’, 4MHz

:INPut:CDR:SECond:TDENsity[?]

Syntax	:INPut:CDR:SECond:TDENsity 'identifier', <NRf>
	:INPut:CDR:SECond:TDENsity? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1' or 'M*.DataIn2'
	<NRf>: Set second order transition density.
Return Range	25% to 100%
Description	This command is used to set the second order transition density of the incoming signal from 10% to 100%. The default is 50%.
	This SCPI is only for M8041A and M8051A.
Example	:INP:CDR:SEC:TDEN 'M1.DataIn1', 25

:INPut:CDR:SECond:LBANdwidth[?]

Syntax	:INPut:CDR:SECond:LBANdwidth 'identifier', <NRf>
	:INPut:CDR:SECond:LBANdwidth? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1' or 'M*.DataIn2'
	<NRf>: Set second order loop bandwidth of the CDR.
Return Range	102 kHz to 20 MHz
Description	This command is used to set the second order loop bandwidth of the CDR. The default is 2 MHz. Acceptable units are Hz, kHz, MHz and exponents (for example, 4E6 is the same as 4 MHz).
	This SCPI is only for M8041A and M8051A.
Example	:INP:CDR:SEC:LBAN 'M1.DataIn1', 4MHz

:INPut:CDR:SECond:PEAKing[?]

Syntax	:INPut:CDR:SECond:PEAKing ‘identifier’, <NRf>
	:INPut:CDR:SECond:PEAKing? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
	<NRf>: Adjust second order peaking of the CDR.
Return Range	0 dB to 3 dB
Description	This command adjusts the second order peaking of the CDR which is valid for the jitter transfer function. The default is 1 dB.
	This SCPI is only for M8041A and M8051A.
Example	:INP:CDR:SEC:PEAK ‘M1.DataIn1’, 0.5

:INPut:ALIGnment Subnode

This subnode has the following SCPI structure:

```
:INPut
└:ALIGnment
  └:EYE
    ├[:AUTO]
    ├:ACENter
    ├:TCENter
    ├:ABORt
    ├:THReshold[?]
    └:RESult
      ├[:HEIGth]?
      ├:WIDTh?
      ├:THReshold?
      ├:DELay?
      └:POLarity?
```

This subnode has the following commands:

Table 56

Name	Description under
:EYE[:AUTO]	:INPut:ALIGnment:EYE[:AUTO] on page 243
:EYE:ACENter	:INPut:ALIGnment:EYE:ACEN on page 243
:EYE:TCENter	:INPut:ALIGnment:EYE:TCEN on page 244
:EYE:ABORt	:INPut:ALIGnment:EYE:ABORt on page 244
:EYE:THReshold[?]	:INPut:ALIGnment:EYE:THReshold[?] on page 244
:EYE:RESult[HEIGht]?	:INPut:ALIGnment:EYE:RESult[:HEIGht]? on page 245
:EYE:RESult:WIDTh?	:INPut:ALIGnment:EYE:RESult:WIDTh? on page 245

Name	Description under
:EYE:RESult:THReshold?	:INPut:ALIGnment:EYE:RESult:THReshold? on page 245
:EYE:RESult:DELay?	:INPut:ALIGnment:EYE:RESult:DELay? on page 246
:EYE:RESult:POLarity?	:INPut:ALIGnment:EYE:RESult:POLarity? on page 246

:INPut:ALIGnment:EYE[:AUTO]

Syntax :INPut:ALIGnment:EYE[:AUTO] ‘identifier’

Input Parameters ‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'

Description This command starts an auto alignment.

This SCPI is only for M8041A, M8051A and M8062A.

Example :INP:ALIG:EYE ‘M1.DataIn1’

:INPut:ALIGnment:EYE:ACEN

Syntax :INPut:ALIGnment:EYE:ACEN ‘identifier’

Input Parameters ‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'

Description This command initiates a search for the 0/1 threshold voltage midway between the two 0/1 threshold voltages with a measured BER just in excess of the BER configured by the :EYE:THReshold command. If successful, the command leaves the 0/1 threshold at this value.

This SCPI is only for M8041A and M8051A.

Example :INP:ALIG:EYE:ACEN ‘M1.DataIn1’

:INPut:ALIGnment:EYE:TCEN

Syntax	:INPut:ALIGnment:EYE:TCEN ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
Description	This command initiates a search for the value of data/clock delay that puts the active clock edge in the center of the data eye, midway between the two relative delay points with a measured BER just in excess of the BER configured by the :EYE:THReshold command. If successful, the command leaves the data/clock delay at this value. This SCPI is only for M8041A and M8051A.
Example	:INP:ALIG:EYE:TCEN ‘M1.DataIn1’

:INPut:ALIGnment:EYE:ABORt

Syntax	:INPut:ALIGnment:EYE:ABORt ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
Description	This command interrupts and aborts an alignment. This SCPI is only for M8041A, M8051A and M8062A.
Example	:INP:ALIG:EYE:ABOR ‘M1.DataIn1’

:INPut:ALIGnment:EYE:THReshold[?]

Syntax	:INPut:ALIGnment:EYE:THReshold ‘identifier’, <NRf> :INPut:ALIGnment:EYE:THReshold? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2' <NRf>: Set BER threshold.
Return Range	1E-9 to 1E-1
Description	The command sets the BER threshold to be used in the determination of the edges of the eye. The query returns the current BER threshold value. This SCPI is only for M8041A, M8051A and M8062A.
Example	:INP:ALIG:EYE:THR ‘M1.DataIn1’, 1E-3

:INPut:ALIGnment:EYE:RESult[:HEIGht]?

Syntax	:INPut:ALIGnment:EYE:RESult[:HEIGht]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
Description	This is a query command that searches for the value of data amplitude that puts the 0/1 threshold level midway between the upper and lower bounds at which the error ratio exceeds the threshold value set by the :INPut:ALIGnment:EYE:THReshold command. This SCPI is only for M8041A, M8051A and M8062A.
Example	:INP:ALIG:EYE:RES? ‘M1.DataIn1’

:INPut:ALIGnment:EYE:RESult:WIDTh?

Syntax	:INPut:ALIGnment:EYE:RESult:WIDTh? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
Description	This is a query command that interrogates the eye width found by the most recent search for the value of data/clock delay that put the active edge in the center of the data eye. This SCPI is only for M8041A, M8051A and M8062A.
Example	:INP:ALIG:EYE:RES:WIDT? ‘M1.DataIn1’

:INPut:ALIGnment:EYE:RESult:THReshold?

Syntax	:INPut:ALIGnment:EYE:RESult:THReshold? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
Description	Returns the threshold detected at the last alignment run. This SCPI is only for M8041A, M8051A and M8062A.
Example	:INP:ALIG:EYE:RES:THR? ‘M1.DataIn1’

:INPut:ALIGnment:EYE:RESult:DELay?

Syntax	:INPut:ALIGnment:EYE:RESult:DELay? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
Description	Returns the delay detected at the last alignment run. This SCPI is only for M8041A, M8051A and M8062A.
Example	:INP:ALIG:EYE:RES:DEL? ‘M1.DataIn1’

:INPut:ALIGnment:EYE:RESult:POLarity?

Syntax	:INPut:ALIGnment:EYE:RESult:POLarity? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'
Return Range	NOR|INV
Description	Returns the polarity of the signal detected at the last alignment run. This SCPI is only for M8041A, M8051A and M8062A.
Example	:INP:ALIG:EYE:RES:POL? ‘M1.DataIn1’

CLOCk Subsystem

The CLOCk subsystem is used to detect clock source or to set the clock frequency applied to input or output channel.

NOTE

M8061A has only DataOut with Clock functional block, so the commands are only supported in the DataOut location and not in the DataIn location.

This subsystem has the following SCPI structure:

```
:CLOCk
├[:SOURce]?
└:FREQuency[?]
```

This subsystem has the following commands:

Table 57

Name	Description under
[:SOURce][?]	:CLOCk[:SOURce][?] on page 248
:FREQuency [?]	:CLOCk:FREQuency [?] on page 248

:CLOCk[:SOURce][?]

Syntax	:CLOCk[:SOURce] ‘identifier’, <NRf>
	:CLOCk[:SOURce]? ‘identifier’
Input Parameters	‘identifier’: 'M*.DataOut' and ‘M*.DataIn’
	<NRf>: CDR | CLK | SYS | AUXCLK
Return Range	CDR | CLK | SYS | AUXCLK
Description	This command sets the source of the clock signal.
	For Data Out:
	CLK -> Front panel ‘CLK IN’ connector
	AUX CLK -> Front panel ‘AUX CLK IN’ connector
	For Data In:
	CDR -> Clock data recovery on analyzer
	SYS -> Internal clock generation
	CLK -> Front panel ‘CLK IN’ connector
	M*.DataIn location is only valid for M8062A.
	This SCPI is only for M8061A and M8062A.
Example	:CLOCk:SOURce 'M2.DataOut', CLK

:CLOCk:FREQuency [?]

Syntax	CLOCk:FREQuency ‘Identifier', <NRf>
	CLOCk:FREQuency? ‘Identifier'
Input Parameters	‘identifier’: 'M*.DataOut'
Return Range	NRf
Description	This command sets the clock frequency applied to the output channel.
	This SCPI is only for M8061A and M8062A.
Example	:CLOCK:FREQ 'M3.DataOut', 8.0e9

FETCh Subsystem

The FETCh subsystem is used to query the error detector's results. Commands for starting, stopping, resetting, etc. are found in the :PLUGin:ERATio subsystem.

This subsystem has the following SCPI structure:

```
:FETCh
├:BCOunter?
└:IBERate?
```

This subsystem has the following commands:

Table 58

Name	Description under
:BCOunter?	:FETCh:BCOunter? on page 249
:IBERate?	:FETCh:IBERate? on page 251

:FETCh:BCOunter?

Syntax :FETCh:BCOunter? ‘identifier’

Input Parameters ‘identifier’: 'M*.DataIn1', 'M*.DataIn2', or 'M*.DataIn'

Description This query returns the accumulated bit counts since the start of the accumulation period. The results are combined to an expression. An expression is comparable to a struc in a programming language. It's a set of parameters of different types separated be a comma and enclosed by parenthesis.

Commands for starting, stopping, resetting, etc. are found in the :PLUGin:ERATio subsystem.

The returned expression contains a timestamp, the location and a set of bit counter values:

(TimeStamp, “Location”, Counted1s, Counted0s, Erroneous1s, Erroneous0s)

Timestamp: As timestamp the Unix epoch time is used; it is defined as elapsed seconds, milliseconds etc., since 01.01.1970 UTC.

Location: String refers to the place the bit counter is located. This parameter is important if a group name is used as 'identifier'.

Counted1s: The number of received ones since the start of the accumulation period

Counted0s: The number of received zeros since the start of the accumulation period.

Erroneous1s: The number of received erroneous ones since the start of the accumulation period.

Erroneous0s: The number of received erroneous zeros since the start of the accumulation period.

Additional counters for 8b10b coding:

- ComparedSymbols
- ErroredSymbols
- IllegalSymbols
- WrongDisparity
- FillerSymbols
- ReceivedSymbols
- Frames
- ErroredFrames

Additional counters for 128/130 coding:

- Blocks
- ErroredBlocks
- IllegalSyncHeaders
- FillerSymbols
- ModifiedFillerSymbols
- Frames
- ErroredFrames

If this query is sent to a 'group' of inputs the response will be expanded by additional sets of counted values. The expression contains a separate set of values starting with the location for every input belonging to a group.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example :FETCh:BCO? 'M1.DataIn1'

(1.366014336e9,“M1.DataIn1“,1.2222e10,1.112345e10,5.1e1,7.2e2)

:FETCh:IBERate?

Syntax :FETCh:IBERate? ‘identifier’

Input Parameters ‘identifier’: 'M*.DataIn1' or 'M*.DataIn2'

Description This query returns the instantaneous bit error rate measured by the addressed error detector. The response contains a timestamp, the location and the measured BER. For Timestamp definition, see the :FETC:BCO? SCPI command. As ‘identifier’ a group name is also allowed. The response will be expanded by additional sets of values per error detector included in the group.

Commands for starting, stopping, resetting, etc. are found in the :PLUGin:ERATio subsystem.

The returned expression contains a timestamp, the location and a set of bit counter values:

(TimeStamp, “Location”, BER)

Timestamp: As timestamp the Unix epoch time is used; it is defined as elapsed seconds, milliseconds etc., since 01.01.1970 UTC.

Location: String refers to the place the bit counter is located. This parameter is important if a group name is used as ‘identifier’.

BER: The instantaneous bit error ratio.

Additional counters for 8b10b coding:

- SymbolErrorRatio
- IllegalSymbolRatio
- DisparityErrorRatio
- FillerSymbolRatio
- FrameErrorRatio

Additional counters for 128/130 coding:

- BlockErrorRatio
- FillerSymbolRatio
- FrameErrorRatio
- ErrorRatio
- ComparedBits
- ErroredBits
- Blocks
- ErroredBlocks

- IllegalSyncHeader
- FillerSymbols
- ModifiedFillerSymbols
- TotalWords
- FER
- Frames
- ErroredFrames

Additional counters for 128/132 coding:

- CorrectedSyncHeaders

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example :FETC:IBER? ‘M1.DataIn1’
(1.266014336e9, “M1.DataIn1“, 1.2345e-1)

MMEMory Subsystem

All instrument states of the M8020A/M8030A instrument are stored in a “workspace”. There are two defined areas in a workspace. One area is named “Factory”. In this area you can find read only predefined settings and patterns. The other area is named “User”. This area stores customer created settings or patterns.

The “Path” argument is a simple string that can be used to address a setting. This is a simple name that follows the Windows specification for allowed characters of a file name. Additionally, group or folder names can be specified for grouping settings. For example, “SATA/Test1” or “SATA/InitialTests/TestAtTemperature80” and “SATA/InitialTests/TestAtTemperature32” or “PCIeStartUpParameter”.

This subsystem has the following SCPI structure:

```
:MMEMory
└:WORKspace
  ├:SETTing
  │ ├:USER
  │ │ ├:CATalog?
  │ │ ├:STORe
  │ │ ├:RECall
  │ │ └:DELete
  │ └:FACTory
  │   ├:CATalog?
  │   └:RECall
  └:IMPort
    └[:USER]
      ├:CATalog?
      └:DELete
```

This subsystem has the following commands:

Table 59

Name	Description under
:WORKspace:SETTings:USER:CATalog?	:MMEMory:WORKspace:SETTings[:USER]:CATalog? on page 255
:WORKspace:SETTings:USER:STORe	:MMEMory:WORKspace:SETTings[:USER]:STORe on page 255
:WORKspace:SETTings:USER:RECall	:MMEMory:WORKspace:SETTings[:USER]:RECall on page 255
:WORKspace:SETTings:USER:DELete	:MMEMory:WORKspace:SETTings[:USER]:DELete on page 256
:WORKspace:SETTings:FACTory:CATalog?	:MMEMory:WORKspace:SETTings:FACTory:CATalog? on page 256
:WORKspace:SETTings:FACTory:RECall	:MMEMory:WORKspace:SETTings:FACTory:RECall on page 256
:WORKspace:IMPort[:USER]	:MMEMory:WORKspace:IMPort[:USER] on page 257
:WORKspace:IMPort[:USER]:CATalog?	:MMEMory:WORKspace:IMPort[:USER]:CATalog? on page 258
:WORKspace:IMPort[:USER]:DELete	:MMEMory:WORKspace:IMPort[:USER]:DELete on page 258

:MMEMory:WORKspace:SETTings[:USER]:CATalog?

Syntax :MMEMory:WORKspace:SETTings[:USER]:CATalog?

Description This query returns a list of names currently stored in the “User” area.

Example :MMEM:WORK:SETT:CAT?

"PCI3_InitialTestSetting,DefaultSetting_1,DefaultSetting_2, DefaultSetting_3,DefaultSetting_4,PCI3/InitialTestA, PCI3/InitialTestB, PCI3_InitialTestSetting"

:MMEMory:WORKspace:SETTings[:USER]:STORe

Syntax :MMEMory:WORKspace:SETTings[:USER]:STORe “Path”

Input Parameters “Path”: Specify name/path to store instrument state.

Description This command stores the instrument state to the specified location. The “Path” argument of this command can be a simple string like “TestConfigurationPCIe3”; the instrument state will then be stored into a location named “TestConfigurationPCIe3”. In addition, it can specify a group or folder name. This command is meant for grouping settings into one folder (for example, “PCIe1/TestConfiguration1”).

Example :MMEM:WORK:SETT:STOR “TestConfigurationSATA_1”
:MMEM:WORK:SETT:STOR “SATA/TestConfiguration_1”
:MMEM:WORK:SETT:STOR “SATA/TestConfiguration_2”
:MMEM:WORK:SETT:STOR “SATA/TestConfiguration_3”

:MMEMory:WORKspace:SETTings[:USER]:RECall

Syntax :MMEMory:WORKspace:SETTings[:USER]:RECall “Path”

Input Parameters “Path”: Specify location of stored instrument state.

Description This command recalls/restores the instrument state from the specified location.

Example :MMEM:WORK:SETT:REC “TestConfigurationSATA_1”
:MMEM:WORK:SETT:REC “SATA/TestConfiguration_1”
:MMEM:WORK:SETT:REC “SATA/TestConfiguration_2”

:MMEMory:WORKspace:SETTings[:USER]:DELete

Syntax	:MMEMory:WORKspace:SETTings[:USER]:DELete “Path”
Input Parameters	“Path”: Specify location of stored instrument state.
Description	This command deletes the instrument state from the specified location.
Example	:MMEM:WORK:SETT:DEL “TestConfigurationSATA_1” :MMEM:WORK:SETT:DEL “SATA/TestConfiguration_1” :MMEM:WORK:SETT:DEL “SATA/TestConfiguration_2”

:MMEMory:WORKspace:SETTings:FACTory:CATalog?

Syntax	:MMEMory:WORKspace:SETTings:FACTory:CATalog?
Description	This command creates a list of “predefined” setting names located in the “Factory” area of the workspace.
Example	:MMEM:WORK:SETT:FACT:CAT? “PCIe3_InitialTestSetting”

:MMEMory:WORKspace:SETTings:FACTory:RECall

Syntax	:MMEMory:WORKspace:SETTings:FACTory:RECall “Path”
Input Parameters	“Path”: Specify location of stored instrument state.
Description	This command recalls/restores the instrument state from a “factory predefined” addressed location.
Example	:MMEMory:WORKspace:SETTings:FACTory:RECall "PCIe1/PCIe1_compliance_ASIC"

:MMEMory:WORKspace:IMPort[:USER]

Syntax
MMEMory:WORKspace:IMPort[:USER]
<ProfileType>,<"SourcePath">,<"RelativeTargetPath">

Input Parameters
<ProfileType>: Specify jitter sweep or spread spectrum clocking profile type or SPProfile.

<“SourcePath”>: Specify the absolute source path of file to be imported.

<“RelativeTargetPath”>: Specify a current setting or a shared (common) storage location.

Description
An arbitrary profile waveform file can be imported for generating jitter sweep or spread spectrum clocking. Three arguments have to be specified:

The first one is the profile type JSPRofile | SSCProfile | SPProfile,

second one is the absolute source path of the file to be imported,

third one is a relative target path "current:..." addresses the current setting and "shared: ..." addresses a location in the storage that is common for all user settings.

For jitter sweep files, the file extension "...jcs" has to be used and for spread spectrum clocking the extension "...txt".

S-parameter profiles containing1- to 9-port components are supported. The file extension "...s1p...s9p" can be used.

Example
:MMEM:WORK:IMP
jspr,“c:/temp/Jittersweep.jcs”,“shared:Jittersweep.jcs”

Example
:MMEM:WORK:IMP
jspr,“c:/temp/Jittersweep.jcs”,“current:Jittersweep.jcs”

Example
:MMEM:WORK:IMP
jspr,“c:/temp/Jittersweep.jcs”,“current:test/PCIe3.0/Jittersweep.jcs”

Example
:MMEM:WORK:IMP
ssp, "c:/temp/INCH7_7_88.s4p","shared:INCH7_7_88.s4p"

Example
:MMEM:WORK:SSC
ssc, "c:/temp/INCH7_7_88.s4p","shared:INCH7_7_88.s4p"

:MMEMory:WORKspace:IMPort[:USER]:CATalog?

Syntax :MMEMory:WORKspace:IMPort[:USER]:CATalog? <ProfileType>

Input Parameters <ProfileType>: Specify jitter sweep, spread spectrum clocking and spread spectrum profile type i.e. <JSPRofile | SSCProfile | SPProfile>.

Description This query generates a list of imported files based on the profile type.

Example :MMEM:WORK:IMP:CAT? jspr

Returns:
“current:Jittersweep.jcs”,“shared:Jittersweep.jcs”,
“shared:Sweep.jcs”,“shared:Test42.jcs”

:MMEMory:WORKspace:IMPort[:USER]:DELete

Syntax :MMEMory:WORKspace:IMPort[:USER]:DELete
<ProfileType>,<“RelativeTargetPath”>

<ProfileType>: Specify jitter sweep, spread spectrum clocking and spread spectrum profile type i.e.<JSPRofile | SSCProfile |SPProfile>.

<“RelativeTargetPath”>: Specify a current setting or a shared (common) storage location.

Description Deletes an imported profile waveform file from the specified storage location. First argument is the profile type and the second argument is the relative target path of the file to be deleted.

Example :MMEM:WORK:IMP:DEL jspr,“shared:Jittersweep.jcs”
Example :MMEM:WORK:IMP:DEL jspr,“current:Jittersweep.jcs”
Example :MMEM:WORK:IMP:DEL sscp,“current:deviation.txt”
Example :MMEM:WORK:IMP:DEL sscp,“shared:deviation.txt”
Example :MMEM:WORK:IMP:DEL spp,"shared: INCH7_7_88.s4p"

PLUGin Subsystem

The M8020A/M8030A platform supports a plugin interface used to implement certain interfaces recognized by the software and integrated into the M8070A GUI and instrument software. Error Ratio, Output Timing, Jitter Tolerance, Output Level, Eye Diagram and Pattern Capture measurements are examples of the plugin concept.

For more information about the plugin interface, refer to PLUGin Subsystem on page 29.

This subsystem has the following SCPI structure:

```
:PLUGin
├:CATalog?
├:RNODes
│└[:LIST]?
├:ERATio
│└...
├:OTIMing
│└...
├:OLEVel
│└...
├:JTOLerance
│└...
├:EDIagram
│└...
└:CCAPture
 └...
```

This subsystem has the following subnodes:

Table 60

Name	Description under
:CATalog?	:PLUGin:CATalog? on page 260
:RNODes[:LIST]?	
:ERATio	:PLUGin:ERATio Subnode on page 261
:OTIMing	:PLUGin:OTIMing Subnode on page 279
:OLEVel	:PLUGin:OLEVel Subnode on page 307
:JTOLerance	:PLUGin:JTOLerance Subnode on page 340
:EDIagram	:PLUGin:EDIagram Subnode on page 364
:CCAPture	:PLUGin:CCAPture Subnode on page 388

:PLUGin:CATalog?

Syntax :PLUGin:CATalog?

Description This query returns the names of all the plug-ins.

Example :PLUGin:CATalog?

"Pattern Capture","Script Editor","DUT Control Interface","Error Ratio","Jitter Tolerance","Jitter Tolerance Template Editor","Output Level","Output Timing","C-Phy Frame Generator","D-Phy Frame Generator"

:PLUGin:ERATio Subnode

This is the basic measurement for the bit error ratio, symbol error ratio and frame error ratio and is integrated with the M8070A software during the startup. The measurement comes with the set of acquisition and evaluation parameters which are used to configure the measurement run. The acquisition parameters can only be configured before the measurement starts (i.e., you cannot change the acquisition parameter number while the measurement is running); however, the evaluation parameters can be configured at any time, even while the measurement is running.
Results can be retrieved using the :PLUGin:ERATio:FETCh:DATA? or :FETCh subsystem commands.

This subnode has the following SCPI structure:

```
:PLUGin
└:ERATio
  ├:ACQuisition
  │ ├:ALOCation[?]
  │ ├:AEND[?]
  │ ├:DURation[?]
  │ ├:HISTory[?]
  │ ├:INTerval[?]
  │ ├:TCLevel[?]
  │ ├:TERatio[?]
  │ └:TIME[?]
  ├:EVALuation
  │ ├:CERatio[?]
  │ └:RVMode?
  ├:FETCh
  │ ├:DATA?
  │ └[:DURation]?
  │   └:PFResult?
  ├:CATalog?
  ├:DELete
  ├:NEW
  ├:RESet
  ├:STARt
  ├:STOP
  ├:SHOW:SGLegends
  └:RUN
    ├:HISTory
    │ ├[:STATe][?]
    │ └:CLEar
    ├:LOG?
    ├:MESSage?
    ├:PROGress?
    └[:STATus]?
```

This subnode has the following commands:

Table 61

Name	Description under
:ACQuisition:ALOCation[?]	:PLUGin:ERATio:ACQuisition:ALOCation[?] on page 264
:ACQuisition:AEND[?]	:PLUGin:ERATio:ACQuisition:AEND[?] on page 265
:ACQuisition:DURation[?]	:PLUGin:ERATio:ACQuisition:DURation[?] on page 265
:ACQuisition:HISTory[?]	:PLUGin:ERATio:ACQuisition:HISTory[?] on page 266
:ACQuisition:INTerval[?]	:PLUGin:ERATio:ACQuisition:INTerval[?] on page 266
:ACQuisition:TCLevel[?]	:PLUGin:ERATio:ACQuisition:TCLevel[?] on page 267
:ACQuisition:TERatio[?]	:PLUGin:ERATio:ACQuisition:TERatio[?] on page 267
:ACQuisition:TIME[?]	:PLUGin:ERATio:ACQuisition:TIME[?] on page 268
:EVALuation:CERatio[?]	:PLUGin:ERATio:EVALuation:CERatio[?] on page 268
:EVALuation:RVMode?	:PLUGin:ERATio:EVALuation:RVMode[?] on page 269
:FETCh:DATA?	:PLUGin:ERATio:FETCh:DATA? on page 270
:FETCh[:DURation]?	:PLUGin:ERATio:FETCh[:DURation]? on page 271
:FETCh[:DURation]:PFResult?	:PLUGin:ERATio:FETCh[:DURation]:PFResult? on page 272
:CATalog?	:PLUGin:ERATio:CATalog? on page 273
:DELete	:PLUGin:ERATio:DELete on page 274
:NEW	:PLUGin:ERATio:NEW on page 273
:RESet	:PLUGin:ERATio:RESet on page 274
:STARt	:PLUGin:ERATio:STARt on page 274

Name	Description under
:STOP	:PLUGin:ERATio:STOP on page 275
:SHOW:SGLegends	:PLUGin:ERATio:SHOW:SGLegends on page 275
:RUN:HISTory[:STATe]	:PLUGin:ERATio:RUN:HISTory[:STATe][?] on page 276
:RUN:HISTory:CLEar	:PLUGin:ERATio:RUN:HISTory:CLEar on page 276
:RUN:LOG?	:PLUGin:ERATio:RUN:LOG? on page 277
:RUN:MESSage?	:PLUGin:ERATio:RUN:MESSage? on page 277
:RUN:PROGress?	:PLUGin:ERATio:RUN:PROGress? on page 278
:RUN[:STATus]?	:PLUGin:ERATio:RUN[:STATus]? on page 278

:PLUGin:ERATio:ACQuisition:ALOCation[?]

Syntax :PLUGin:ERATio:ACQuisition:ALOCation ‘identifier’, <location-string>

:PLUGin:ERATio:ACQuisition:ALOCation? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

<location-string>: Specify the location identifier or group name identifier.

Return Range <location or group name>

Description This command sets the location identifier (or group name identifier) against which the data acquisition is performed for the specified measurement name identifier. The query returns the current location or group name for the specified measurement name.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example :PLUG:ERAT:ACQ:ALOC ‘MyMeasurement’,“M1.DataIn1”

:PLUGin:ERATio:ACQuisition:AEND[?]

Syntax	:PLUGin:ERATio:ACQuisition:AEND 'identifier', <PFA|FDUR>
	:PLUGin:ERATio:ACQuisition:AEND? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
	<PFA|FDUR>: Set accumulation end mode.
Return Range	PFA|FDUR
Description	This command sets the accumulation to end when a pass/fail (PFA) condition has been met or end after the full duration (FDUR) of the measurement for the addressed measurement name identifier. The query returns the current accumulation end value.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:ERAT:ACQ:AEND 'MyMeasurement', PFA

:PLUGin:ERATio:ACQuisition:DURation[?]

Syntax	:PLUGin:ERATio:ACQuisition:DURation 'identifier', <FTIM|IND>
	:PLUGin:ERATio:ACQuisition:DURation? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
	<FTIM|IND>: Set accumulation duration mode.
Return Range	FTIM|IND
Description	This command sets the accumulation duration mode to either a fixed time (FTIM) or indefinitely (IND) for the addressed measurement name identifier. If FTIM is selected, use the :PLUGin:ERATio:ACQuisition:TIME command to set the duration time. The query returns the current accumulation duration value.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:ERAT:ACQ:DUR 'MyMeasurement', FTIM

:PLUGin:ERATio:ACQuisition:HISTory[?]

Syntax
:PLUGin:ERATio:ACQuisition:HISTory 'identifier', <NRf>
:PLUGin:ERATio:ACQuisition:HISTory? 'identifier'

Input Parameters
'identifier': Specify the measurement name.
<NRf>: Set size of accumulation values.

Return Range
1 to 100000

Description
This command sets the size of accumulation values to be kept in memory for the addressed measurement name identifier. The query returns the current history size.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example
:PLUG:ERAT:ACQ:HIST 'MyMeasurement', 100000

:PLUGin:ERATio:ACQuisition:INTerval[?]

Syntax
:PLUGin:ERATio:ACQuisition:INTerval 'identifier', <NRf>
:PLUGin:ERATio:ACQuisition:INTerval? 'identifier'

Input Parameters
'identifier': Specify the measurement name.
<NRf>: Set accumulation interval.

Return Range
100E-3 to 2E6

Description
This command sets how often to take error ratio samples (from 0.1 to 2000000 seconds) for the addressed measurement name identifier. The query returns the current accumulation interval.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

NOTE **Do not use the "Ms" unit with a value. Doing so will cause an error. Instead, use "mas" for the unit. For example, 2E6 is the same as 2 mas.**

Example
:PLUG:ERAT:ACQ:INT 'MyMeasurement',0.3

:PLUGin:ERATio:ACQuisition:TCLevel[?]

Syntax	:PLUGin:ERATio:ACQuisition:TCLevel 'identifier', <NRf> :PLUGin:ERATio:ACQuisition:TCLevel? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <NRf>: Set target confidence level.
Return Range	0.1% to 99.9%
Description	This command sets the target confidence level when accumulation end is set to pass/fail (in percent) for the addressed measurement name identifier. The query returns the current target confidence level. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:ERAT:ACQ:TCL 'MyMeasurement', 95

:PLUGin:ERATio:ACQuisition:TERatio[?]

Syntax	:PLUGin:ERATio:ACQuisition:TERatio 'identifier', <NRf> :PLUGin:ERATio:ACQuisition:TERatio? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <NRf>: Set target error ratio.
Return Range	1E-18 to 1E-3
Description	This command sets the target error ratio for the addressed measurement name identifier. The query returns the current target error ratio value. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:ERAT:ACQ:TER 'MyMeasurement', 1e-09

:PLUGin:ERATio:ACQuisition:TIME[?]

Syntax
:PLUGin:ERATio:ACQuisition:TIME ‘identifier’, <NRf>

:PLUGin:ERATio:ACQuisition:TIME? ‘identifier’

Input Parameters
‘identifier’: Specify the measurement name.

<NRf>: Set accumulation fixed time.

Return Range
1 s to 31.5E6 s

Description
This command sets the accumulation fixed time, when accumulation duration is set to FTIM (fixed time), for the addressed measurement name identifier. The query returns the current accumulation fixed time value in seconds.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

NOTE
Do not use the “Ms” unit with a value. Doing so will cause an error. Instead, use “mas” for the unit. For example, 20E6 is the same as 20 mas.

Example
:PLUG:ERAT:ACQ:TIME ‘MyMeasurement’,1000

:PLUGin:ERATio:EVALuation:CERatio[?]

Syntax
:PLUGin:ERATio:EVALuation:CERatio ‘identifier’, <ERAT|EZR|EOR>

:PLUGin:ERATio:EVALuation:CERatio? ‘identifier’

Input Parameters
‘identifier’: Specify the measurement name.

<ERAT|EZR|EOR>: Set error ratio analysis mode.

Return Range
ERAT|EZR|EOR

Description
This command sets the error ratio analysis mode to error ratio of all accumulated bits (ERAT), errored zero and counted zero bits (EZR), or errored ones and counted ones bits (EOR) for the addressed measurement name identifier. The query returns the current setting error ratio analysis mode.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example
:PLUG:ERAT:EVAL:CER ‘MyMeasurement’,EOR

:PLUGin:ERATio:EVALuation:RVMode[?]

Syntax	:PLUGin:ERATio:EVALuation:RVMode ‘identifier’, <DET|SUMM> :PLUGin:ERATio:EVALuation:RVMode? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name. <DET|SUMM>: Select measurement results view mode.
Return Range	DET|SUMM
Description	This command sets the measurement results view mode to detailed or summary when viewing the results using the :PLUGin:ERATio:FETCh:DATA? command. This mode can be changed during a measurement or after a measurement has completed. The default is summary mode. DET (detailed) sets the mode to return the error ratio, compared bits, errored bits, errored zero ratio, compared zeroes, errored zeroes, errored one ratio, compared ones, errored ones, etc. SUMM (summary) sets the mode to return the error ratio, compared bits, errored bits, etc. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:ERAT:EVAL:RVM ‘MyMeasurement’, DET

:PLUGin:ERATio:FETCh:DATA?

Syntax :PLUGin:ERATio:FETCh:DATA? ‘identifier’[,‘Location’]

Input Parameters ‘identifier’: Specify the measurement name.

[,‘Location’]: ‘M*.DATAOUT1’ or ‘M*.DataOut2’ (optional).

Return Range ‘M*.DATAOUT1’ or ‘M*.DataOut2’ (data results)

Description This command returns the raw data of the error ratio measurement for the addressed measurement name identifier and location identifier. A measurement run on location identifier ‘M1.DataIn1’, for example, would return results specific to ‘M1.DataIn1’ only. If the location identifier is omitted and measurement data exists from multiple locations, results will be returned for every location.

The first item in the comma separated list is the location identifier name (for example, “M1.DataIn1”). The successive values are dependent on the coding, shown in the order below and are repeated in the same order for subsequent measured points.

Bit Error Counter

TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, ...

Counter for 8b/10b Bit Coding

TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, ComparedSymbols, ErroredSymbols, IllegalSymbols, WrongDisparity, FillerSymbols, ReceivedSymbols, Frames, ErroredFrames, ...

Counter for 128/130 Bit Coding

TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, Blocks, ErroredBlocks, IllegalSynHeaders, FillerSymbols, ModifiedFillerSymbols, Frames, ErroredFrames, ...

Counter for 128/132 Bit Coding

TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, Blocks, ErroredBlocks, IllegalSynHeaders, CorrectedSyncHeaders, FillerSymbols, ModifiedFillerSymbols, Frames, ErroredFrames, ...

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example shows the results returned for 128/130 Bit Coding.

:PLUG:ERAT:FETC:DATA? ‘Error Ratio 1’,‘M1.DataIn1’

Return:
("M1.DataIn1",1420795894.76206,499999811,500000189,0,0,7692307,0,
0,0,0,0,0, ...)

If the location is omitted from the above example e.g. if the query is :PLUG:ERAT:FETC:DATA? ‘Error Ratio 1’ then the query returns the comma separated list of all the analyzer location group participating in the measurement.

Return:
("M1.DataIn1",1420795894.76206,499999811,500000189,0,0,7692307,0,
0,0,0,0,0, ...),
("M1.DataIn2",1420795894.76206,499999811,500000189,0,0,7692307,0,
0,0,0,0,0, ...)

:PLUGin:ERATio:FETCh[:DURation]?

Syntax	:PLUGin:ERATio:FETCh[:DURation]? ‘identifier’[,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: Specify the location identifier (optional).
Return Range	<NR3>
Description	This command returns the bit error ratio duration, nOneBits, nZeroBits, nOneBitErrs and nZeroBitErrs of the specified measurement name and location identifier. A measurement run on location ‘M1.DataIn1’, for example, would return results specific to ‘M1.DataIn1’ only. If the location identifier is omitted and measurement data exists from multiple locations, results will be returned for every location. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:ERAT:FETC? ‘MyMeasurement’,‘M1.DataIn1’

:PLUGin:ERATio:FETCh[:DURation]:PFResult?

Syntax: :PLUGin:ERATio:FETCh[:DURation]:PFResult? ‘identifier’[,‘Location’]

Input Parameters: ‘identifier’: Specify the measurement name.

[,‘Location’]: Specify the location identifier (optional).

Return Range: <NR3>

Description: This command returns the results from the :PLUGin:ERATio:FETCh[:DURation]? query as well as four additional values of the specified measurement name and location identifier:

- ConfidenceLevelAtTargetErrorRatio
- MinimumErrorRatioAtTargetConfidenceLevel
- MaximumErrorRatioAtTargetConfidenceLevel
- PassFailedResult (“UNKNOWN”, “PASS”, or “FAIL” if the AEND parameter is set to “PassFail”)

If the AEND parameter is set to “FullDuration”, the result is the same as PLUGin:ERATio:FETCh[:DURation]? except that the four additional values are set to “Unknown” (not applicable).

A measurement run on location ‘M1.DataIn1’, for example, would return results specific to ‘M1.DataIn1’ only. If the location identifier is omitted and measurement data exists from multiple locations, results will be returned for every location.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example: :PLUG:ERAT:FETC:PFR? ‘MyMeasurement’,‘M1.DataIn1’

:PLUGin:ERATio:CATalog?

Syntax :PLUGin:ERATio:CATalog?

Description This command returns a list of all created error ratio measurement names currently available for measuring.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example Assume the following is a list of created error ratio measurement names:

:PLUG:ERAT:NEW 'ERAT_1'

:PLUG:ERAT:NEW 'ERAT_2'

:PLUG:ERAT:NEW 'ERAT_3'

The command and returned list would look like the following:

:PLUG:ERAT:CAT?

"ERAT_1,ERAT_2,ERAT_3"

:PLUGin:ERATio:NEW

Syntax :PLUGin:ERATio:NEW 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command creates a new error ratio measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example creates an error ratio measurement name identifier called 'MyMeasurement':

:PLUG:ERAT:NEW 'MyMeasurement'

NOTE

Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use "0" as a parameter input in this command.

Example- :PLUG:ERAT:NEW 'MyMeasurement',0

:PLUGin:ERATio:DELete

Syntax :PLUGin:ERATio:DELete ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to delete.

Description This command deletes a previously created error ratio measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example deletes an error ratio measurement addressed by the measurement name identifier called ‘MyMeasurement’:

:PLUG:ERAT:DEL ‘MyMeasurement’

:PLUGin:ERATio:RESet

Syntax :PLUGin:ERATio:RESet ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to reset.

Description This command resets an error ratio measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example resets an error ratio measurement addressed with the measurement name identifier called ‘MyMeasurement’:

:PLUG:ERAT:RES ‘MyMeasurement’

:PLUGin:ERATio:STARt

Syntax :PLUGin:ERATio:STARt ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to start.

Description This command starts an error ratio measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example starts an error ratio measurement addressed with the measurement name identifier called ‘MyMeasurement’:

:PLUG:ERAT:STAR ‘MyMeasurement’

:PLUGin:ERATio:STOP

Syntax	:PLUGin:ERATio:STOP ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name to stop.
Description	This command stops an error ratio measurement addressed by the measurement name identifier. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	The following example stops an error ratio measurement with the measurement name identifier called ‘MyMeasurement’: :PLUG:ERAT:STOP ‘MyMeasurement’

:PLUGin:ERATio:SHOW:SGLegends

Syntax	:PLUGin:ERATio:SHOW:SGLegends 'Identifier', <0|1|OFF|ON> :PLUGin:ERATio:SHOW:SGLegends? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement' <0|1|OFF|ON>: Specify the show/hide state for graph legends.
Return Range	0|1
Description	This command shows/hides the graph legends. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUGin:ERATio:SHOW:SGLegends 'MyMeasurement', 1

:PLUGin:ERATio:RUN:HISTory[:STATe][?]

Syntax :PLUGin:ERATio:RUN:HISTory[:STATe] ‘identifier’ <0|1|ON|OFF>

:PLUGin:ERATio:RUN:HISTory[:STATe]? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Return Range 0|1|ON|OFF

Description This command enables/disables the storage of error ratio measurement results addressed by the measurement name identifier.

This query returns the current setting.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example enables storage of error ratio measurement results with the measurement name identifier called ‘MyMeasurement’:

:PLUG:ERAT:RUN:HIST ‘MyMeasurement’, 1

:PLUGin:ERATio:RUN:HISTory:CLEar

Syntax :PLUGin:ERATio:RUN:HISTory:CLEar ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command deletes the storage of error ratio measurement history addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example deletes error ratio measurement history with the measurement name identifier called ‘MyMeasurement’:

:PLUGin:ERATio:RUN:HISTory:CLEar ‘MyMeasurement’

:PLUGin:ERATio:RUN:LOG?

Syntax :PLUGin:ERATio:RUN:LOG? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns logs for the addressed measurement.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example returns logs for the measurement name identifier called 'MyMeasurement':

:PLUGin:ERATio:RUN:LOG? 'MyMeasurement'

Output:

#284 07/20/2015 18:26:56,Measurement.Error Ratio.Error Ratio 1,Info,"Start Measurement"

:PLUGin:ERATio:RUN:MESSage?

Syntax :PLUGin:ERATio:RUN:MESSage? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This query returns a string describing the state of an error ratio measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Finished, Error, or Stopped.

This SCPI is for M8041A, M8051A, M8061 and M8062A.

Example The following example returns the state of an error ratio measurement with the measurement name identifier called 'MyMeasurement':

:PLUG:ERAT:RUN:MESS? 'MyMeasurement'
Running

:PLUGin:ERATio:RUN:PROGress?

Syntax :PLUGin:ERATio:RUN:PROGress? ‘identifier’

Input Parameters ‘identifier’: Specify measurement name.

Return Range 0.0 to 1.0

Description This query returns a number in the range of 0.0 to 1.0 to indicate the progress of an error ratio measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example returns the progress of an error ratio measurement with the measurement name identifier called ‘MyMeasurement’:

:PLUG:ERAT:RUN:PROG? ‘MyMeasurement’
0.51

:PLUGin:ERATio:RUN[:STATus]?

Syntax :PLUGin:ERATio:RUN[:STATus]? ‘identifier’

Input Parameters ‘identifier’: Specify measurement name.

Return Range 0|1

Description This command returns the running status of an error ratio measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example returns the running status of an error ratio measurement with the measurement name identifier called ‘MyMeasurement’:

:PLUG:ERAT:RUN ‘MyMeasurement’, 1

:PLUGin:OTIMing Subnode

This is the basic measurement for BER versus the sampling point delay, which is displayed as a “Bathtub” curve and is integrated with the M8070A software during the startup. The measurement comes with the set of acquisition and evaluation parameters which are used to configure the measurement run. The acquisition parameters can only be configured before the measurement starts (i.e., you cannot change the value of the acquisition parameter while the measurement is running); however, the evaluation parameters can be configured at any time even while the measurement is running.

This subnode has the following SCPI structure:

```
:PLUGin
└:OTIMing
  ├:ACQuisition
  │ ├:ALOCation[?]
  │ ├:DRESolution[?]
  │ ├:FTJBer[?]
  │ ├:OPTimization[?]
  │ ├:COMPared[?]
  │ ├:EBEnabled[?]
  │ └:ERRored[?]
  ├:EVALuation
  │ ├:BERThresh[?]
  │ ├:MBJPartition[?]
  │ ├:ETJBer[?]
  │ └:ERRType[?]
  ├:SHOW
  │ ├:SMPoints
  │ ├:UNIT
  │ ├:VSCale
  │ └:SGLegends
  ├:CATalog?
  ├:DELete
  ├:NEW
  ├:RESet
  ├:STARt
  ├:STOP
  ├:BREak
  ├:STEP
  ├:CONTinue
  ├:RUN
  │ ├:HISTory
  │ │ ├[:STATe][?]
  │ │ └:CLEar
  │ ├:LOG?
  │ ├:MESSage?
  │ ├:PROGress?
  │ └[:STATus]?
  └:FETCh
    └...
```

This subnode has the following commands:

Table 62

Name	Description under
:ACQuisition:ALOCation[?]	:PLUGin:OTIMing:ACQuisition:ALOCation[?] on page 282
:ACQuisition:DRESolution[?]	:PLUGin:OTIMing:ACQuisition:DRESolution[?] on page 283
:ACQuisition:FTJBer[?]	:PLUGin:OTIMing:ACQuisition:FTJBer[?] on page 283
:ACQuisition:OPTimization[?]	:PLUGin:OTIMing:ACQuisition:OPTimization[?] on page 284
:ACQuisition:COMPared[?]	:PLUGin:OTIMing:ACQuisition:COMPared[?] on page 284
:ACQuisition:EBEnabled[?]	:PLUGin:OTIMing:ACQuisition:EBEnabled[?] on page 285
:ACQuisition:ERRored[?]	:PLUGin:OTIMing:ACQuisition:ERRored[?] on page 285
:EVALuation:BERThresh[?]	:PLUGin:OTIMing:EVALuation:BERThresh[?] on page 286
:EVALuation:MBJPartition[?]	:PLUGin:OTIMing:EVALuation:MBJPartition[?] on page 286
:EVALuation:ETJBer[?]	:PLUGin:OTIMing:EVALuation:ETJBer[?] on page 287
:EVALuation:ERRType[?]	:PLUGin:OTIMing:EVALuation:ERRType[?] on page 287
:SHOW:SMPoints[?]	:PLUGin:OTIMing:SHOW:SMPoints[?] on page 288
:SHOW:UNIT[?]	:PLUGin:OTIMing:SHOW:UNIT[?] on page 288
:SHOW:VSCale[?]	:PLUGin:OTIMing:SHOW:VSCale[?] on page 289
:SHOW:SGLegends[?]	:PLUGin:OTIMing:SHOW:SGLegends on page 289
:CATalog?	:PLUGin:OTIMing:CATalog? on page 289

Name	Description under
:DELete	:PLUGin:OTIMing:DELete on page 290
:NEW	:PLUGin:OTIMing:NEW on page 290
:RESet	:PLUGin:OTIMing:RESet on page 291
:STARt	:PLUGin:OTIMing:STARt on page 291
:STOP	:PLUGin:OTIMing:STOP on page 291
:BREak	:PLUGin:OTIMing:BREak on page 292
:STEP	:PLUGin:OTIMing:STEP on page 292
:CONTinue	:PLUGin:OTIMing:CONTinue on page 293
:RUN:HISTory[:STATe]	:PLUGin:OTIMing:RUN:HISTory[:STATe][?] on page 293
:RUN:HISTory:CLEar	:PLUGin:OTIMing:RUN:HISTory:CLEar on page 293
:RUN:LOG?	:PLUGin:OTIMing:RUN:LOG? on page 294
:RUN:MESSage?	:PLUGin:OTIMing:RUN:MESSage? on page 294
:RUN:PROGress?	:PLUGin:OTIMing:RUN:PROGress? on page 295
:RUN[:STATus]?	:PLUGin:OTIMing:RUN[:STATus]? on page 295
:FETCh	:PLUGin:OTIMing:FETCh Subnode on page 296

:PLUGin:OTIMing:ACQuisition:ALOCation[?]

Syntax

:PLUGin:ERATio:ACQuisition:ALOCation 'identifier', <location-string>

:PLUGin:ERATio:ACQuisition:ALOCation? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<location-string>: Specify location or group name identifier.

Return Range

<location or group name>

Description

This command associates the location or group name identifier with a measurement name identifier against which the data acquisition is performed. The group name identifier is defined using the :SYSTem:INSTrument:GROup:DEFine command.

The query returns the current location or group name identifier against which the measurement name identifier is configured for the data acquisition.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OTIM:ACQ:ALOC ‘MyMeasurement’,“M1.DataIn1”

:PLUGin:OTIMing:ACQuisition:DRESolution[?]

Syntax :PLUGin:OTIMing:ACQuisition:DRESolution ‘identifier’, <NRf>

:PLUGin:OTIMing:ACQuisition:DRESolution? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

<NRf>: Set the sample delay resolution.

Return Range 1 mUI to 250 mUI

Description This command sets the sample delay resolution in terms of UI and associates it with a specific measurement name identifier. The query returns the current sample delay resolution value in UI.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OTIM:ACQ:DRES ‘MyMeasurement’, 0.25

:PLUGin:OTIMing:ACQuisition:FTJBer[?]

Syntax :PLUGin:OTIMing:ACQuisition:FTJBer ‘identifier’, <NRf>

:PLUGin:OTIMing:ACQuisition:FTJBer? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

<NRf>: Specify the BER value for the fast total jitter measurement.

Return Range 1E-15 to 1E-9

Description This command specifies the BER value for the fast total jitter measurement and associates it with a specific measurement name identifier. The query returns the current value at which the fast total jitter measurement was performed.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OTIM:ACQ:FTJB ‘MyMeasurement’, 1E-10

:PLUGin:OTIMing:ACQuisition:OPTimization[?]

Syntax	:PLUGin:OTIMing:ACQuisition:OPTimization ‘identifier’, <FTJ|NONE> :PLUGin:OTIMing:ACQuisition:OPTimization? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name. <FTJ|NONE>: Specify the sample delay optimization technique.
Return Range	FTJ|NONE
Description	This command specifies the sample delay optimization technique and associates it with a specific measurement name identifier. If NONE is selected then no optimization is performed. If FTJ is selected then the fast total jitter measurement is performed. The query returns the current value. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OTIM:ACQ:OPT ‘MyMeasurement’, FTJ

:PLUGin:OTIMing:ACQuisition:COMPared[?]

Syntax	:PLUGin:OTIMing:ACQuisition:COMPared ‘identifier’, <NRf> :PLUGin:OTIMing:ACQuisition:COMPared? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name. <NRf>: Specify the number of compared bits.
Return Range	1E+0 to 1E+18
Description	This command specifies the number of compared bits used as criteria for moving to the next measurement point when the sample delay optimization is set to NONE. This command also associates this parameter with a specific measurement name identifier. The query returns the current value. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OTIM:ACQ:COMP ‘MyMeasurement’, 1E+7

:PLUGin:OTIMing:ACQuisition:EBEnabled[?]

Syntax	:PLUGin:OTIMing:ACQuisition:EBEnabled 'identifier', <OFF|ON|0|1> :PLUGin:OTIMing:ACQuisition:EBEnabled? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <OFF|ON|0|1>: Disable/enable errored bits as criteria to move to the next measurement point.
Return Range	0|1
Description	This command enables/disables the number of errored bits. This command also associates this parameter with a specific measurement name identifier. If enabled (1 or ON), then errored bits will be used along with the number of compared bits as the criteria for moving to the next measurement point when the sample delay optimization is set to NONE. If disabled (0 or OFF), then only the number of compared bits will be used as the criteria for moving to the next measurement point while running the measurement. The query returns the current value. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OTIM:ACQ:EBE 'MyMeasurement', 1

:PLUGin:OTIMing:ACQuisition:ERRored[?]

Syntax	:PLUGin:OTIMing:ACQuisition:ERRored 'identifier', <NRf> :PLUGin:OTIMing:ACQuisition:ERRored? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <NRf>: Specify the number of errored bits.
Return Range	1E+0 to 1E+18
Description	This command specifies the number of errored bits. This command also associates this parameter with a specific measurement name identifier. This command can only be used when the :PLUGin:OTIMing:ACQuisition:EBEnabled command is set to 1 (enabled) and the sample delay optimization is set to NONE. This is the criteria for moving to the next measurement point while running the measurement. The query returns the current value. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OTIM:ACQ:ERR 'MyMeasurement', 1E+4

:PLUGin:OTIMing:EVALuation:BERThresh[?]

Syntax	:PLUGin:OTIMing:EVALuation:BERThresh 'identifier', <NRf>
	:PLUGin:OTIMing:EVALuation:BERThresh? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
	<NRf>: Specify the BER threshold value.
Return Range	1E-15 to 1E-1
Description	This command specifies the BER threshold value used to calculate the measurement parameters and associates it with a specific measurement name identifier. The query returns the current BER threshold value.
	This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OTIM:EVAL:BERT 'MyMeasurement', 1E-10

:PLUGin:OTIMing:EVALuation:MBJPartition[?]

Syntax	:PLUGin:OTIMing:EVALuation:MBJPartition 'identifier', <NRf>
	:PLUGin:OTIMing:EVALuation:MBJPartition? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
	<NRf>: Specify the minimum BER for separating random and deterministic jitter components.
Return Range	1E-18 to 1E+0
Description	This command specifies the minimum BER value for separating the random and deterministic jitter components and associates it with a specific measurement name identifier. In order to separate the random and deterministic jitter from the bathtub, the specified BER threshold value must be greater than the value specified by this command. The range is 1E-18 to 1E+0. The query returns the minimum value specified to separate the random and deterministic jitter components. The default is 1E-9.
	This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OTIM:EVAL:MBJP 'MyMeasurement', 1E-7

:PLUGin:OTIMing:EVALuation:ETJBer[?]

Syntax

:PLUGin:OTIMing:EVALuation:ETJBer ‘identifier’, <EM6|...|EM12>

:PLUGin:OTIMing:EVALuation:ETJBer? ‘identifier’

Input Parameters

‘identifier’: Specify the measurement name.

<EM6|...|EM12>: Specify the residual BER for total jitter estimation calculation.

Return Range

EM6 to EM12

Description

This command specifies the residual BER for total jitter estimation calculation and associates it with a specific measurement name identifier. The range is 1E-12 (EM12) to 1E-6 (EM6). The query returns the current residual BER for total jitter estimation calculation. The default is 1E-12.

This SCPI is for M8041A, M8051A and M8062A.

Example

:PLUG:OTIM:EVAL:ETJB ‘MyMeasurement’, EM7

:PLUGin:OTIMing:EVALuation:ERRType[?]

Syntax

:PLUGin:OTIMing:EVALuation:ERRType ‘identifier’, <AERR|ZERR|OERR>

:PLUGin:OTIMing:EVALuation:ERRType? ‘identifier’

Input Parameters

‘identifier’: Specify the measurement name.

<AERR|ZERR|OERR>: Specify the error type to evaluate.

Return Range

AERR|ZERR|OERR

Description

This command specifies the error type to evaluate and associates it with a specific measurement name identifier. AERR allows evaluation of all errors. ZERR allows evaluation of errored zeroes and OERR allows evaluation of errored ones. The query returns the current value.

This SCPI is for M8041A, M8051A and M8062A.

Example

:PLUG:OTIM:EVAL:ERRT ‘MyMeasurement’, ZERR

:PLUGin:OTIMing:SHOW:SMPoints[?]

Syntax :PLUGin:OTIMing:SHOW:SMPoints ‘identifier’, <OFF|ON|0|1>

:PLUGin:OTIMing:SHOW:SMPoints? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

<OFF|ON|0|1>: Disable/enable viewing the measurement points.

Return Range 0|1

Description This command disables/enables viewing of the measurement points in the GUI and associates it with a specific measurement name identifier. A 0 or OFF value disables viewing and a 1 or ON value enables viewing of the measurement points. The query returns the current value.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OTIM:SHOW:SMP ‘MyMeasurement’, 1

:PLUGin:OTIMing:SHOW:UNIT[?]

Syntax :PLUGin:OTIMing:SHOW:UNIT ‘identifier’, <UINT|TIME>

:PLUGin:OTIMing:SHOW:UNIT? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

<UINT|TIME>: Specify the timing units.

Return Range UINT|TIME

Description This command specifies the timing units and associates it with a specific measurement name identifier. If UINT is specified then the timing units are in UI. If TIME is specified then the timing units are in seconds. The query returns the current value.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OTIM:SHOW:UNIT ‘MyMeasurement’, unit

:PLUGin:OTIMing:SHOW:VSCale[?]

Syntax
:PLUGin:OTIMing:SHOW:VSCale ‘identifier’, <LOG|LIN>

:PLUGin:OTIMing:SHOW:VSCale? ‘identifier’

Input Parameters
‘identifier’: Specify the measurement name.

<LOG|LIN>: Specify the vertical axis scale.

Return Range
LOG|LIN

Description
This command specifies the vertical BER axis scale and associates it with a specific measurement name identifier. If LOG is specified then the BER values are in terms of a logarithmic scale. If LIN is specified then the BER values are in terms of a linear scale. The query returns the current value.

This SCPI is for M8041A, M8051A and M8062A.

Example
:PLUG:OTIM:SHOW:VSC ‘MyMeasurement’, LOG

:PLUGin:OTIMing:SHOW:SGLegends

Syntax
:PLUGin:OTIMing:SHOW:SGLegends 'Identifier', <0|1|OFF|ON>

:PLUGin:OTIMing:SHOW:SGLegends? 'Identifier'

Input Parameters
Identifier: 'MyMeasurement'

<0|1|OFF|ON>: Specify the show/hide state for graph legends.

Return Range
0|1

Description
This command shows/hides the graph legends.

This SCPI is for M8041A, M8051A and M8062A.

Example
:PLUGin:OTIMing:SHOW:SGLegends 'MyMeasurement', 1

:PLUGin:OTIMing:CATalog?

Syntax
:PLUGin:OTIMing:CATalog?

Description
This command returns a list of all created output timing measurement names currently available for measuring.

This SCPI is for M8041A, M8051A and M8062A.

Example
Assume the following is a list of created output timing measurement names:

:PLUG:OTIM:NEW ‘OTIM_1’

:PLUG:OTIM:NEW ‘OTIM_2’

:PLUG:OTIM:NEW ‘OTIM_3’

The command and returned list would look like the following:

:PLUG:OTIM:CAT?

“OTIM_1”,“OTIM_2”,“OTIM_3”

:PLUGin:OTIMing:NEW

Syntax	:PLUGin:OTIMing:NEW ‘identifier’
Input Parameters	‘identifier’: Specify a new measurement name.
Description	This command creates a new output timing measurement name identifier. This SCPI is for M8041A, M8051A and M8062A.
Example	The following example creates an output timing measurement with the measurement name identifier called ‘MyOutputTiming’: :PLUG:OTIM:NEW ‘MyOutputTiming’

NOTE

Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use “0” as a parameter input in this command.

Example- :PLUG:OTIM:NEW ‘MyOutputTiming’,0

:PLUGin:OTIMing:DELete

Syntax	:PLUGin:OTIMing:DELete ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name to delete.
Description	This command deletes a previously created output timing measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A and M8062A.

Example — The following example deletes an output timing measurement with the measurement name identifier called ‘MyOutputTiming’:

:PLUG:OTIM:DEL ‘MyOutputTiming’

:PLUGin:OTIMing:RESet

Syntax — :PLUGin:OTIMing:RESet ‘identifier’

Input Parameters — ‘identifier’: Specify the measurement name to reset.

Description — This command resets an output timing measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A and M8062A.

Example — The following example resets an output timing measurement addressed with the measurement name identifier called ‘MyOutputTiming’:

:PLUG:OTIM:RES ‘MyOutputTiming’

:PLUGin:OTIMing:STARt

Syntax — :PLUGin:OTIMing:STARt ‘identifier’

Input Parameters — ‘identifier’: Specify the measurement name to start.

Description — This command starts an output timing measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A and M8062A.

Example — The following example starts an output timing measurement with the measurement name identifier called ‘MyOutputTiming’:

:PLUG:OTIM:STAR ‘MyOutputTiming’

:PLUGin:OTIMing:STOP

Syntax — :PLUGin:OTIMing:STOP ‘identifier’

Input Parameters — ‘identifier’: Specify the measurement name to stop.

Description This command stops an output timing measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A and M8062A.

Example The following example stops an output timing measurement with the measurement name identifier called ‘MyOutputTiming’:

:PLUG:OTIM:STOP ‘MyOutputTiming’

:PLUGin:OTIMing:BREak

Syntax :PLUGin:OTIMing:BREak ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to break/pause.

Description This command breaks/pauses a measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A and M8062A.

Example The following example breaks an output timing measurement with the measurement name identifier called ‘MyOutputTiming’:

:PLUG:OTIM:BRE ‘MyOutputTiming’

:PLUGin:OTIMing:STEP

Syntax :PLUGin:OTIMing:STEP ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to step.

Description This command allows you to step through an output timing measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is for M8041A, M8051A and M8062A.

Example The following example triggers a single output timing measurement step of the measurement addressed with the measurement name identifier called ‘MyOutputTiming’:

:PLUG:OTIM:STEP ‘MyOutputTiming’

:PLUGin:OTIMing:CONTinue

Syntax :PLUGin:OTIMing:CONTinue ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to continue.

Description This command allows you to continue an output timing measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is for M8041A, M8051A and M8062A.

Example The following example continues a halted output timing measurement addressed with the measurement name identifier called ‘MyOutputTiming’:

:PLUG:OTIM:CONT ‘MyOutputTiming’

:PLUGin:OTIMing:RUN:HISTory[:STATe][?]

Syntax :PLUGin:OTIMing:RUN:HISTory[:STATe] ‘identifier’,<0|1|OFF|ON>

:PLUGin:OTIMing:RUN:HISTory[:STATe]? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Return Range 0|1

Description This command enables/disables the storage of output timing measurement results addressed by the measurement name identifier.

This query returns the current setting.

This SCPI is for M8041A, M8051A and M8062A.

Example The following example enables storage of output timing measurement results with the measurement name identifier called ‘MyMeasurement’:

:PLUG:OTIM:RUN:HIST ‘MyMeasurement’, 1

:PLUGin:OTIMing:RUN:HISTory:CLEar

Syntax :PLUGin:OTIMing:RUN:HISTory:CLEar ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command deletes the output timing measurement history addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A and M8062A.

Example The following example deletes the output timing measurement history with the measurement name identifier called ‘MyMeasurement’:

:PLUG:OTIM:RUN:HIST:CLE ‘MyMeasurement’

:PLUGin:OTIMing:RUN:LOG?

Syntax :PLUGin:OTIMing:RUN:LOG? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns logs for the addressed measurement.

This SCPI is for M8041A, M8051A and M8062A.

Example This command returns logs for the measurement ‘MyMeasurement’.

:PLUG:OTIM:RUN:LOG? ‘MyMeasurement’

Output:

#287 07/20/2015 18:25:52,Measurement.Output Timing.Output Timing 1,Info,"Stop Measurement"

:PLUGin:OTIMing:RUN:MESSage?

Syntax :PLUGin:OTIMing:RUN:MESSage? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns a string describing the state of an output timing measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Suspended, Finished, Error, or Stopped.

This SCPI is for M8041A, M8051A and M8062A.

Example The following example returns the state of an output timing measurement with the measurement name identifier called ‘MyMeasurement’:

:PLUG:OTIM:RUN:MESS? ‘MyMeasurement’
Running

:PLUGin:OTIMing:RUN:PROGress?

Syntax :PLUGin:OTIMing:RUN:PROGress? ‘identifier’

Input Parameters ‘identifier’: Specify measurement name.

Return Range 0.0 to 1.0

Description This command returns a number in the range of 0.0 to 1.0 to indicate the progress of an output timing measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is for M8041A, M8051A and M8062A.

Example The following example returns the progress of an output timing measurement with the measurement name identifier called ‘MyMeasurement’:

:PLUG:OTIM:RUN:PROG? ‘MyMeasurement’

0.51

:PLUGin:OTIMing:RUN[:STATus]?

Syntax :PLUGin:OTIMing:RUN[:STATus]? ‘identifier’

Input Parameters ‘identifier’: Specify measurement name.

Return Range 0|1

Description This command returns the running status of an output timing measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is for M8041A, M8051A and M8062A.

Example The following example returns the running status of an output timing measurement with the measurement name identifier called ‘MyMeasurement’:

:PLUG:OTIM:RUN? ‘MyMeasurement’
1

:PLUGin:OTIMing:FETCh Subnode

This subnode has the following SCPI structure:

```
:PLUGin
└:OTIMing
  └:FETCh
    ├:EDGE
    │ ├[:LEFT]
    │ │ ├[:POINts]?
    │ │ └:RSQuare?
    │ └:RIGHt
    │   ├[:POINts]?
    │   └:RSQuare?
    ├[:JITTer]
    │ └[:TOTal]
    │   ├:DJITter?
    │   ├:ESTimated?
    │   ├:MEAN?
    │   ├:PTPeak?
    │   ├[:RMSquare]?
    │   ├:RANDom
    │   │ └[:RMSquare]?
    │   ├:OSDelay?
    │   ├:PMARgin?
    │   └:UNCertainty?
    ├:DATA?
    ├:DELay?
    └:FREQuency?
```

This subnode has the following commands:

Table 63

Name	Description under
:EDGE[:LEFT][:POINts]?	:PLUGin:OTIMing:FETCh:EDGE[:LEFT][:POINts]? on page 298
:EDGE[:LEFT]:RSQuare?	:PLUGin:OTIMing:FETCh:EDGE[:LEFT]:RSQuare? on page 298
:EDGE:RIGHt[:POINts]?	:PLUGin:OTIMing:FETCh:EDGE:RIGHt[:POINts]? on page 299
:EDGE:RIGHt:RSQuare?	:PLUGin:OTIMing:FETCh:EDGE:RIGHt:RSQuare? on page 299
[:JITTer][:TOTal]:DJITTer?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:DJITTer? on page 300
[:JITTer][:TOTal]:ESTimated?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:ESTimated? on page 300
[:JITTer][:TOTal]:MEAN?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:MEAN? on page 301
[:JITTer][:TOTal]:PTPeak?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PTPeak? on page 301
[:JITTer][:TOTal][:RMSquare]?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal][:RMSquare]? on page 302
[:JITTer][:TOTal]:RANDom [:RMSquare]?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:RANDom[:RMSquare]? on page 302
[:JITTer][:TOTal]:OSDelay?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:OSDelay? on page 303
[:JITTer][:TOTal]:PMARgin?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PMARgin? on page 303
[:JITTer][:TOTal]:UNCertainty?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:UNCertainty? on page 304
:DATA?	:PLUGin:OTIMing:FETCh:DATA? on page 304
:DELay?	:PLUGin:OTIMing:FETCh:DELay? on page 305
:FREQuency?	:PLUGin:OTIMing:FETCh:FREQuency? on page 306

:PLUGin:OTIMing:FETCh:EDGE[:LEFT][:POINts]?

Syntax :PLUGin:OTIMing:FETCh:EDGE[:LEFT][:POINts]? ‘identifier’[,‘Location’]

Input Parameters ‘identifier’: Specify the measurement name.

[‘Location’]: Specify location (optional).

Return Range <NR3>

Description This command returns the number of points that have been measured between the BER Threshold and the Min. BER for RJ/DJ separation threshold associated with a specific measurement name identifier. It is calculated for left slope (left edge). This number has to be greater than 2 for the RJ, DJ and estimated TJ values to be applicable.

This SCPI is for M8041A, M8051A and M8062A.

A measurement run on a location identifier (for example, ‘M1.DataIn1’) will return following results.

Example :PLUG:OTIM:FETC:EDGE? ‘MyOutputTiming’,‘M1.DataIn1’

:PLUGin:OTIMing:FETCh:EDGE[:LEFT]:RSQuare?

Syntax :PLUGin:OTIMing:FETCh:EDGE[:LEFT]:RSQuare? ‘identifier’[,‘Location’]

Input Parameters ‘identifier’: Specify the measurement name.

[‘Location’]: Specify location (optional).

Return Range <NR3>

Description This command returns the R^2 values calculated for left slope of the bathtub curve (left edge) associated with a specific measurement name identifier. They are a measure of how well the transformed points between BER Threshold and Min. BER for RJ/DJ Separation fit to the linear regression. They have to be greater than 0.75 for the RJ, DJ and estimated TJ values to be applicable.

This SCPI is for M8041A, M8051A and M8062A.

A measurement run on a location identifier (for example, ‘M1.DataIn1’) will return following results.

Example :PLUG:OTIM:FETC:EDGE:RSQ? ‘MyOutputTiming’,‘M1.DataIn1’

:PLUGin:OTIMing:FETCh:EDGE:RIGHt[:POINts]?

Syntax	:PLUGin:OTIMing:FETCh:EDGE:RIGHt[:POINts]? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. ['Location']: Specify location (optional).
Return Range	<NR3>
Description	This command returns the number of points that have been measured between the BER Threshold and the Min. BER for RJ/DJ separation threshold associated with a specific measurement name identifier. It is calculated for right slope (right edge). This number has to be greater than 2 for the RJ, DJ and estimated TJ values to be applicable. This SCPI is for M8041A, M8051A and M8062A. A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.
Example	:PLUG:OTIM:FETC:EDGE:RIGH? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh:EDGE:RIGHt:RSQuare?

Syntax	:PLUGin:OTIMing:FETCh:EDGE:RIGHt:RSQuare? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. ['Location']: Specify location (optional).
Return Range	<NR3>
Description	This command returns the R^2 values calculated for right slope of the bathtub curve (right edge) associated with a specific measurement name identifier. They are a measure of how well the transformed points between BER Threshold and Min. BER for RJ/DJ Separation fit to the linear regression. They have to be greater than 0.75 for the RJ, DJ and estimated TJ values to be applicable. This SCPI is for M8041A, M8051A and M8062A. A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.
Example	:PLUG:OTIM:FETC:EDGE:RIGH:RSQ? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:DJITter?

Syntax	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:DJITter? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. ['Location']: Specify location (optional).
Return Range	<NR3>
Description	This command returns the total jitter component with non-Gaussian distribution (deterministic jitter) associated with a specific measurement name identifier. After transforming a contiguous range of measured points into Q space and performing a linear regression, it is calculated as the period minus the difference between the means of the two straight lines. This SCPI is for M8041A, M8051A and M8062A. A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.
Example	:PLUG:OTIM:FETC:DJIT? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:ESTimated?

Syntax	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:ESTimated? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. ['Location']: Specify location (optional).
Return Range	<NR3>
Description	This command returns an estimate of the expected jitter for very low bit error rates associated with a specific measurement name identifier. After extrapolating the measured BER curves, it is calculated as the period minus the expected width of the eye opening. This SCPI is for M8041A, M8051A and M8062A. A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.
Example	:PLUG:OTIM:FETC:EST? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:MEAN?

Syntax	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:MEAN? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. ['Location']: Specify location (optional).
Return Range	<NR3>
Description	This command returns the mean value for total jitter associated with a specific measurement name identifier. This value is calculated as the weighted average of the left edge jitter histogram. This command also associates this parameter with a specific measurement name identifier. This SCPI is for M8041A, M8051A and M8062A. A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.
Example	:PLUG:OTIM:FETC:MEAN? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PTPeak?

Syntax	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PTPeak? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. ['Location']: Specify location (optional).
Return Range	<NR3>
Description	This command returns the peak to peak value for total jitter associated with a specific measurement name identifier. This value is calculated as the pulse period (unit interval) minus the phase margin. This SCPI is for M8041A, M8051A and M8062A. A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.
Example	:PLUG:OTIM:FETC:PTP? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal][:RMSquare]?

Syntax :PLUGin:OTIMing:FETCh[:JITTer][:TOTal][:RMSquare]? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the average of the left and right jitter histogram root mean squared values associated with a specific measurement name identifier.

This SCPI is for M8041A, M8051A and M8062A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example :PLUG:OTIM:FETC? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:RANDom[:RMSquare]?

Syntax :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:RANDom[:RMSquare]? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the total jitter component with Gaussian distribution associated with a specific measurement name identifier. After transforming a contiguous range of measured points into Q space and performing a linear regression, it is calculated as the mean of the sigmas of the two straight lines. The contiguous range is limited by the BER threshold and the minimum BER for RJ/DJ separation threshold.

This SCPI is for M8041A, M8051A and M8062A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example :PLUG:OTIM:FETC:RAND? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:OSDelay?

Syntax	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:OSDelay? ‘identifier’[,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [‘Location’]: Specify location (optional).
Return Range	<NR3>
Description	This command returns the average of the left (Ai) and right (Bi) bathtub/BER threshold intersections (optimum sample point delay) associated with a specific measurement name identifier. This SCPI is for M8041A, M8051A and M8062A. A measurement run on a location identifier (for example, ‘M1.DataIn1’) will return following result.
Example	:PLUG:OTIM:FETC:OSD? ‘MyOutputTiming’,‘M1.DataIn1’

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PMARgin?

Syntax	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PMARgin? ‘identifier’[,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [‘Location’]: Specify location (optional).
Return Range	<NR3>
Description	This command returns the period of time where the bit error rate is lower than the BER threshold (phase margin) associated with a specific measurement name identifier. This SCPI is for M8041A, M8051A and M8062A. A measurement run on a location identifier (for example, ‘M1.DataIn1’) will return following result.
Example	:PLUG:OTIM:FETC:PMAR? ‘MyOutputTiming’,‘M1.DataIn1’

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:UNCertainty?

Syntax: :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:UNCertainty? ‘identifier’[,‘Location’]

Input Parameters: ‘identifier’: Specify the measurement name.

[‘Location’]: Specify location (optional).

Return Range: <NR3>

Description: This command returns the total jitter uncertainty associated with a specific measurement name identifier. The returned value is only valid if the “Sample Delay Optimization” is set to “Fast Total Jitter” (FTJ) using the :PLUG:OTIM:ACQ:OPT command.

This SCPI is for M8041A, M8051A and M8062A.

A measurement run on a location identifier (for example, ‘M1.DataIn1’) will return following result.

Example: :PLUG:OTIM:FETC:UNC? ‘MyOutputTiming’,‘M1.DataIn1’

:PLUGin:OTIMing:FETCh:DATA?

Syntax: :PLUGin:OTIMing:FETCh:DATA? ‘identifier’[,‘Location’]

Input Parameters: ‘identifier’: Specify the measurement name.

[‘Location’]: Specify location (optional).

Description: This command returns the raw data of the output timing measurement associated with a specific measurement name identifier.

A measurement run on a location identifier (for example, ‘M1.DataIn1’) will return following results.

This SCPI is for M8041A, M8051A and and M8062A.

Example: :PLUG:OTIM:FETC:DATA? ‘Output Timing 1’,‘M1.DataIn1’

"M1.DataIn1",1420788572.96819,499999011,500000989,249999011,250000000,-0.6035,"UNKNOWN", ...

The first item in the comma separated list is the name of the location for which the data is being queried. Rest of the values in the comma separated list for each measurement point are Time Stamp, Compared

One Bits, Compared Zero Bits, Errored One Bits, Errored Zero Bits, Sample Delay and Pass/Fail Results and are repeated in the same order for subsequent measured points.

In the case of Sample Delay Optimization as "None" the pass/fail result for each measurement point is returned as "UNKNOWN". However if the Fast Total Jitter Measurement is performed then the Pass/Fail results could be either "PASS", "FAIL" or "UNKNOWN"

If the location is omitted from the above example e.g. if the query is :PLUG:OTIM:FETC:DATA? ‘Output Timing 1’ then the query returns the comma separated list of all the analyzer location group participating in the measurement

(("M1.DataIn1",1420788572.96819,499999011,500000989,249999011,250000000,-0.6035,"UNKNOWN", ...), ("M1.DataIn2",1420788572.96819,499999011,500000989,249999011,250000000,-0.6035,"UNKNOWN",))

:PLUGin:OTIMing:FETCh:DELay?

Syntax :PLUGin:OTIMing:FETCh:DELay? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns the currently set delay sweep in seconds. The delay sweep is set using the :INPut:DELay command. After setting the value, a data center alignment is executed followed by an output timing measurement.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OTIM:FETC:DEL? ‘Output Timing 1’
(“M1.DataIn1”,4.879243000000001E-009)

:PLUGin:OTIMing:FETCh:FREQuency?

Syntax :PLUGin:OTIMing:FETCh:FREQuency? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns the currently set frequency of the measurement. The frequency is set using the :SOURce:FREQuency command.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OTIM:FETC:FREQ? ‘Output Timing 1’
("M1.DataIn1",5.000000000000000E+009)

:PLUGin:OLEVel Subnode

This is the basic measurement for BER versus sample threshold and is integrated with the M8070A software during the startup. The measurement comes with the set of Acquisition and Evaluation Parameters which are used to configure the measurement run. The Acquisition parameters can only be configured before the measurement starts (i.e. during the measurement run you cannot change the value of Acquisition parameter); however the Evaluation parameters can be configured any time even after the measurement has been run.

This subnode has the following SCPI structure:

```
:OLEVel
├─:BREak
├─:CATalog?
├─:CONTinue
├─:DELete
├─:NEW
├─:RESet
├─:STARt
├─:STEP
├─:STOP
├─:ACQuisition
│ └...
├─:EVALuation
│ └...
├─:FETCh
│ └...
├─:SHOW
│ └...
└─:RUN
  └...
```

This subnode has the following commands and subnodes:

Table 64

Name	Description under
:BREak[?]	:PLUGin:OLEVel:BREak on page 309
:CATalog?	:PLUGin:OLEVel:CATalog? on page 309
:CONTinue[?]	:PLUGin:OLEVel:CONTinue on page 309
:DELete[?]	:PLUGin:OLEVel:DELete on page 310
:NEW[?]	:PLUGin:OLEVel:NEW on page 310
:RESet[?]	:PLUGin:OLEVel:RESet on page 311
:STARt[?]	:PLUGin:OLEVel:STARt on page 311
:STEP[?]	:PLUGin:OLEVel:STEP on page 311
:STOP[?]	:PLUGin:OLEVel:STOP on page 312
Subnodes	
:ACQuisition:	:PLUGin:OLEVel:ACQuisition Subnode on page 313
:EVALuation	:PLUGin:OLEVel:EVALuation Subnode on page 318
:FETCh	:PLUGin:OLEVel:FETCh Subnode on page 320
:SHOW	:PLUGin:OLEVel:SHOW Subnode on page 333
:RUN	:PLUGin:OLEVel:RUN Subnode on page 336

:PLUGin:OLEVel:BREak

Syntax	:PLUGin:OLEVel:BREak ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name to break/pause.
Description	This command breaks/pauses a measurement addressed by the measurement name identifier. This SCPI is for M8041A, M8051A and M8062A.
Example	The following example breaks an output level measurement with the measurement name identifier called ‘MyMeasurement’: :PLUG:OLEV:BRE ‘MyMeasurement’

:PLUGin:OLEVel:CATalog?

Syntax	:PLUGin:OLEVel:CATalog?
Description	This command returns a list of all created output level measurement names currently available for measuring. This SCPI is for M8041A, M8051A and M8062A.
Example	Assume three output level measurements named as OLEVel_1, OLEVel_2 and OLEVel_3 are created using NEW command: :PLUG:OLEVel:NEW ‘OLEVel_1’ :PLUG:OLEVel:NEW ‘OLEVel_2’ :PLUG:OLEVel:NEW ‘OLEVel_3’ The command and returned list would look like the following: :PLUG:OLEVel:CAT? “OLEVel_1,OLEVel_2,OLEVel_3”

:PLUGin:OLEVel:CONTinue

Syntax	:PLUGin:OLEVel:CONTinue ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name to continue.
Description	This command allows you to continue an output level measurement, addressed by the measurement name identifier, which was stopped using the BREak command. This SCPI is for M8041A, M8051A and M8062A.
Example	The following example continues a halted output level measurement addressed with the measurement name identifier called ‘MyMeasurement’: :PLUG:OLEV:CONT ‘MyMeasurement’

:PLUGin:OLEVel:DELete

Syntax	:PLUGin:OLEVel:DELete ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name to delete.
Description	This command deletes a previously created output level measurement addressed by the measurement name identifier. This SCPI is for M8041A, M8051A and M8062A.
Example	The following example deletes an output level measurement addressed by the measurement name identifier called ‘MyMeasurement’: :PLUG:OLEV:DEL ‘MyMeasurement’

:PLUGin:OLEVel:NEW

Syntax	:PLUGin:OLEVel:NEW ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name.
Description	This command creates a new output level measurement name identifier. This SCPI is for M8041A, M8051A and M8062A.
Example	The following example creates an output level measurement name identifier called ‘MyMeasurement’: :PLUG:OLEV:NEW ‘MyMeasurement’

NOTE

Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use “0” as a parameter input in this command.

Example- :PLUG:OLEV:NEW ‘MyMeasurement’,0

:PLUGin:OLEVel:RESet

Syntax :PLUGin:OLEVel:RESet ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to reset.

Description This command resets an output level measurement addressed by the measurement name identifier.
This SCPI is for M8041A, M8051A and M8062A.

Example The following example resets an output level measurement addressed with the measurement name identifier called ‘MyMeasurement’:
:PLUG:OLEV:RES ‘MyMeasurement’

:PLUGin:OLEVel:STARt

Syntax :PLUGin:OLEVel:STARt ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to start.

Description This command starts an output level measurement addressed by the measurement name identifier.
This SCPI is for M8041A, M8051A and M8062A.

Example The following example starts an output level measurement addressed with the measurement name identifier called ‘MyMeasurement’:
:PLUG:OLEV:STAR ‘MyMeasurement’

:PLUGin:OLEVel:STEP

Syntax :PLUGin:OLEVel:STEP ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to step.

Description This command allows you to step through an output level measurement, addressed by the measurement name identifier, that was stopped using the BREak command.
This SCPI is for M8041A, M8051A and M8062A.

Example: The following example triggers a single output level measurement step of the measurement addressed with the measurement name identifier called ‘MyMeasurement’:
:PLUG:OLEV:STEP ‘MyMeasurement’

:PLUGin:OLEVel:STOP

Syntax: :PLUGin:OLEVel:STOP ‘identifier’

Input Parameters: ‘identifier’: Specify the measurement name to stop.

Description: This command stops an output level measurement addressed by the measurement name identifier.
This SCPI is for M8041A, M8051A and M8062A.

Example: The following example stops an output level measurement with the measurement name identifier called ‘MyMeasurement’:
:PLUG:OLEV:STOP ‘MyMeasurement’

:PLUGin:OLEVel:ACQuisition Subnode

This subnode has the following SCPI structure:

```
:ACQuisition
├─:ALOCation
├─:COMPared
├─:EBEnabled
├─:ERRored
├─:HILevel
├─:LOWLevel
└─:TRESolution
```

This subnode has the following commands:

Table 65

Name	Description under
:ALOCation[?]	:PLUGin:OLEVel:ACQuisition:ALOCation[?] on page 314
:COMPared[?]	:PLUGin:OLEVel:ACQuisition:COMPared[?] on page 314
:EBEnabled[?]	:PLUGin:OLEVel:ACQuisition:EBEnabled[?] on page 315
:ERRored[?]	:PLUGin:OLEVel:ACQuisition:ERRored[?] on page 315
:HILevel[?]	:PLUGin:OLEVel:ACQuisition:HILevel[?] on page 316
:LOWLevel[?]	:PLUGin:OLEVel:ACQuisition:LOWLevel[?] on page 316
:TRESolution[?]	:PLUGin:OLEVel:ACQuisition:TRESolution[?] on page 317

:PLUGin:OLEVel:ACQuisition:ALOCation[?]

Syntax :PLUGin:OLEVel:ACQuisition:ALOCation 'Identifier', < location-string>

:PLUGin:OLEVel:ACQuisition:ALOCation? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Description This command sets the Location/Location Group string against which the Data Acquisition is performed. The query returns the current Location/Location group string against which measurement is configured for the Data Acquisition.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OLEV:ACQ:ALOC 'MyMeasurement',"M1.DataIn1"

:PLUGin:OLEVel:ACQuisition:COMPared[?]

Syntax :PLUGin:OLEVel:ACQuisition:COMPared[?] 'Identifier', <NRf>

Input Parameters Identifier: 'MyMeasurement'

Return Range 1E+0 to 1E+18

Description This command allows the user to specify the number of compared bits; that would be used as criteria for moving to the next measurement point. The query returns current value.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OLEV:ACQ:COMP 'MyMeasurement', 1E+7

:PLUGin:OLEVel:ACQuisition:EBEnabled[?]

Syntax	:PLUGin:OLEVel:ACQuisition:EBEnabled[?] 'Identifier', <0 | 1 | ON | OFF>
Input Parameters	Identifier: 'MyMeasurement'
Return Range	0|1
Description	If enabled then user can specify the number of errored bits additionally with number of compared bits in order to make them criteria together to move to the next measurement point. If this command sets the value as 0 then only number of compared bits are considered to move to the next measurement point while running the measurement. The query returns current value. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OLEV:ACQ:EBE 'MyMeasurement', 1

:PLUGin:OLEVel:ACQuisition:ERRored[?]

Syntax	:PLUGin:OLEVel:ACQuisition:ERRored[?] 'Identifier', <NRf>
Input Parameters	Identifier: 'MyMeasurement'
Return Range	1E+0 to 1E+18
Description	This command allows the user to specify the number of errored bits. This command can only be used when the command PLUGin:OLEVel:ACQuisition:EBEnabled is set to 1. This would be the criteria to move to the next measurement point while the measurement is running. The query returns current value. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OLEV:ACQ:ERR'MyMeasurement', 1E+4

:PLUGin:OLEVel:ACQuisition:HILevel[?]

Syntax	:PLUGin:OLEVel:ACQuisition:HILevel[?] 'Identifier', <NRf>
Input Parameters	Identifier: 'MyMeasurement'
Return Range	-5 V to 5 V
Description	This command allows the user to specify the sample threshold high level. The query returns current value of sample threshold high Level. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OLEV:ACQ:HIL 'MyMeasurement', 0.5

:PLUGin:OLEVel:ACQuisition:LOWLevel[?]

Syntax	:PLUGin:OLEVel:ACQuisition:LOWLevel[?] 'Identifier', <NRf>
Input Parameters	Identifier: 'MyMeasurement'
Return Range	-5 V to 5 V
Description	This command allows the user to specify the Sample Threshold Low Level. The query returns current value of sample threshold low level. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OLEV:ACQ:LOWL 'MyMeasurement', -0.5

:PLUGin:OLEVel:ACQuisition:TRESolution[?]

Syntax	:PLUGin:OLEVel:ACQuisition:TRESolution[?] 'Identifier', <NRf>
Input Parameters	Identifier: 'MyMeasurement'
Return Range	1E-3 to 1
Description	This command allows the user to specify the sample threshold resolution in terms of Volts. The query returns current value of sample threshold resolution in terms of Volts. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OLEV:ACQ:TRES 'MyMeasurement', 0.001

:PLUGin:OLEVel:EVALuation Subnode

This subnode has the following SCPI structure:

```
:EVALuation
├:BERThresh
└:MBQFactor
```

This subnode has the following commands:

Table 66

Name	Description under
:BERThresh[?]	:PLUGin:OLEVel:EVALuation:BERThresh[?] on page 318
:MBQFactor[?]	:PLUGin:OLEVel:EVALuation:MBQFactor[?] on page 319

:PLUGin:OLEVel:EVALuation:BERThresh[?]

Syntax :PLUGin:OLEVel:EVALuation:BERThresh[?] 'Identifier', <NRf>

Input Parameters Identifier: 'MyMeasurement'

Return Range 1E-15 to 1E-1

Description This command allows the user to specify the BER Threshold value against which the measurement parameters are calculated. The query returns current value of the BER threshold.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OLEV:EVAL:BERT 'MyMeasurement', 1E-10

:PLUGin:OLEVel:EVALuation:MBQFactor[?]

Syntax	:PLUGin:OLEVel:EVALuation:MBQFactor[?] 'Identifier', <NRf>
Input Parameters	Identifier: 'MyMeasurement'
Range	1e-15...1.0
Description	This command allows the user to specify the Minimum BER for Q-factor caluclations. Please note that in order to be able to do Q-Factor calculations the specified BER Threshold value must be greater than the value specified by this command. The query returns the minimum value specified to allow the Q-Factor calculations. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:OLEV:EVAL:MBQF 'MyMeasurement', 1E-15

:PLUGin:OLEVel:FETCh Subnode

This subnode has the following SCPI structure:

```
:FETCh
 ├:DATA?
 ├:FREQuency?
 ├:LEVel
 │ └...
 ├:NOISe
 │ └...
 └:QLEVel
   └...
```

This subnode has the following commands and subnodes:

Table 67

Name	Description under
:DATA?	:PLUGin:OLEVel:FETCh:DATA? on page 321
:FREQuency?	:PLUGin:OLEVel:FETCh:FREQuency? on page 321
Subnode	
:LEVel	:PLUGin:OLEVel:FETCh:LEVel Subnode on page 322
:NOISe	:PLUGin:OLEVel:FETCh:NOISe Subnode on page 325
:QLEVel	:PLUGin:OLEVel:FETCh:QLEVel Subnode on page 328

:PLUGin:OLEVel:FETCh:DATA?

Syntax	:PLUGin:OLEVel:FETCh:DATA? ‘identifier’[,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	Return the raw data of the output level measurement. This SCPI is for M8041A, M8051A and M8062A.
Example	A measurement run on location e.g 'M1.DataIn1' would return following results. :PLUG:OLEV:FETC:DATA? ‘MyMeasurement’,'M1.DataIn1' Return: ("M1.DataIn1", ... The first item in that comma separated list is the name of the location here "M1.DataIn1",the next values are TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes and SampleThreshold and these 6 values will be repeated. If the 'Location' is omitted and the measurement runs on multiple locations the response contains all results of current location. (("M1.DataIn1", ...),("M1.DataIn2", ...))

:PLUGin:OLEVel:FETCh:FREQuency?

Syntax	:PLUGin:OLEVel:FETCh:FREQuency? ‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	This command fetches frequency set for the whole module. It also prefixes the acquisition location. This SCPI is for M8041A, M8051A and M8062A.
Example	:plugin:olev:fetch:frequency? "MyMeasurement"

:PLUGin:OLEVel:FETCh:LEVel Subnode

This subnode has the following SCPI structure:

```
:LEVel
├─:AMPlitude?
├─:HIGH?
├─:LOW?
├─:MEAN?
└─:THMargin?
```

This subnode has the following commands:

Table 68

Name	Description under
:AMPlitude?	:PLUGin:OLEVel:FETCh:LEVel:AMPlitude? on page 323
:HIGH?	:PLUGin:OLEVel:FETCh:LEVel:HIGH? on page 323
:LOW?	:PLUGin:OLEVel:FETCh:LEVel:LOW? on page 323
:MEAN?	:PLUGin:OLEVel:FETCh:LEVel:MEAN? on page 323
:THMargin?	:PLUGin:OLEVel:FETCh:LEVel:THMargin? on page 324

:PLUGin:OLEVel:FETCh:LEVel:AMPlitude?

Syntax	:PLUGin:OLEVel:FETCh:LEVel:AMPlitude? ‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	This command returns the value of amplitude calculated after a run. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:LEVel:AMPlitude? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:LEVel:HIGH?

Syntax	:PLUGin:OLEVel:FETCh:LEVel:HIGH? ‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	This command returns the value of the high level. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:LEVel:HIGH? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:LEVel:LOW?

Syntax	:PLUGin:OLEVel:FETCh:LEVel:LOW?‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	This command returns the value of the low level. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:LEVel:LOW? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:LEVel:MEAN?

Syntax	:PLUGin:OLEVel:FETCh:LEVel:MEAN? ‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).

Description	This command returns the value of the mean level. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:LEVel:MEAN? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:LEVel:THMargin?

Syntax	:PLUGin:OLEVel:FETCh:LEVel:THMargin? ‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	This command returns the value of the threshold margin. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:LEVel:THMargin? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:NOISe Subnode

This subnode has the following SCPI structure:

```
:NOISe
├─:PPEak?
├─:HIGH
│ └─:STDeviation?
├─:LOW
│ └─:STDeviation?
└─:SNRatio
  ├─:PPEak?
  └─:RMSquare?
```

This subnode has the following commands:

Table 69

Name	Description under
:PPEak?	:PLUGin:OLEVel:FETCh:NOISe:PPEak? on page 326
:HIGH:STDeviation?	:PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation? on page 326
:LOW:STDeviation?	:PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation? on page 326
:SNRation:PPEak?	:PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak? on page 327
:SNRation:RNSquare?	:PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare? on page 327

:PLUGin:OLEVel:FETCh:NOISe:PPEak?

Syntax	:PLUGin:OLEVel:FETCh:NOISe:PPEak? 'identifier' [,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description	This command returns the value of the peak to peak noise level. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:NOISe:PPEak? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation?

Syntax	:PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation? 'identifier' [,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description	This command returns the value of the Q High Level Standard Deviation. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation?

Syntax	:PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation? 'identifier' [,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description	This command returns the value of the Q Low Level Standard Deviation. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak?

Syntax	:PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak? ‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	This command returns the peak to peak value of the signal to noise ratio. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare?

Syntax	:PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare? ‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	This command returns the value of the root mean square of signal to noise ratio. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:QLEVel Subnode

This subnode has the following SCPI structure:

```
:QLEVel
├─:QFActor?
├─:QOThreshold?
├─:QRBer?
├─:HIGH?
│ ├─:POINts?
│ ├─:RSQuare?
│ └─:STDeviation?
└─:LOW?
  ├─:POINts?
  ├─:RSQuare?
  └─:STDeviation?
```

This subnode has the following commands:

Table 70

Name	Description under
:QFActor?	:PLUGin:OLEVel:FETCh:QLEVel:QFActor? on page 329
:HIGH?	:PLUGin:OLEVel:FETCh:QLEVel:HIGH? on page 330
:HIGH:POINts?	:PLUGin:OLEVel:FETCh:QLEVel:HIGH:POINts? on page 330
:HIGH:RSQuare?	:PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare? on page 331
:HIGH:STDeviation?	:PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDeviation? on page 331

Name	Description under
:LOW?	:PLUGin:OLEVel:FETCh:QLEVel:LOW? on page 331
:LOW:POINts?	:PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts? on page 331
:LOW:RSQuare?	:PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare? on page 332
:LOW:STDeviation?	:PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation? on page 332

:PLUGin:OLEVel:FETCh:QLEVel:QFActor?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:QFActor? ‘identifier’ [,‘Location’]

Input Parameters ‘identifier’: Specify the measurement name.

[,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).

Description This command returns the value of the Q factor.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:FETCh:QLEVel:QFActor? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:QLEVel:QOThreshold?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:QOThreshold? ‘identifier’ [,‘Location’]

Input Parameters ‘identifier’: Specify the measurement name.

[,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).

Description This command returns the value of the Q Optimum threshold.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:FETCh:QLEVel:QOThreshold? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:QLEVel:QRBer?

Syntax	:PLUGin:OLEVel:FETCh:QLEVel:QRBer? ‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	This command returns the value of the Q Residual BER. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:QLEVel:QRBer? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:QLEVel:HIGH?

Syntax	:PLUGin:OLEVel:FETCh:QLEVel:HIGH? ‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	This command returns the value of the Q High Level. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:QLEVel:HIGH? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:QLEVel:HIGH:POINts?

Syntax	:PLUGin:OLEVel:FETCh:QLEVel:HIGH:POINts? ‘identifier’ [,‘Location’]
Input Parameters	‘identifier’: Specify the measurement name. [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).
Description	This command returns the value of the Q High Level Points. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:FETCh:QLEVel:HIGH:POINts? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q High Level R^2.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDeviation?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDeviation? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q High Level Standard Deviation.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDeviation? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:LOW?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:LOW? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q Low Level.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:FETCh:QLEVel:LOW? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q Low Level Points.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare? ‘identifier’ [,‘Location’]

Input Parameters ‘identifier’: Specify the measurement name.

[,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).

Description This command returns the value of the Q Low Level R^2.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare? ‘MyMeasurement’

:PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation? ‘identifier’ [,‘Location’]

Input Parameters ‘identifier’: Specify the measurement name.

[,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’ (optional).

Description This command returns the value of the Q Low Level Standard Deviation.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation? ‘MyMeasurement’

:PLUGin:OLEVel:SHOW Subnode

This subnode has the following SCPI structure:

```
:SHOW
├:SMPoints
├:GRAPhics
├:HSCale
└:SGLegends
```

This subnode has the following commands:

Table 71

Name	Description under
:SMPoints[?]	:PLUGin:OLEVel:SHOW:SMPoints[?] on page 334
:GRAPhics[?]	:PLUGin:OLEVel:SHOW:GRAPhics[?] on page 334
:HSCale[?]	:PLUGin:OLEVel:SHOW:HSCale[?] on page 335
:SGLegends[?]	:PLUGin:OLEVel:SHOW:SGLegends[?] on page 335

:PLUGin:OLEVel:SHOW:SMPoints[?]

Syntax :PLUGin:OLEVel:SHOW:SMPoints[?] 'Identifier', <0|1|ON|OFF>

Input Parameters Identifier: 'MyMeasurement'

Description This command allows the user to control the GUI if he/she wants to visualize the measurement points of the measurement run. Value of 0 indicates that user does not want to visualize the measurement points and 1 indicates that user wants to visualize the measurement points. The query returns current value.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OLEV:SHOW:SMP 'MyMeasurement', 1

:PLUGin:OLEVel:SHOW:GRAPhics[?]

Syntax :PLUGin:OLEVel:SHOW:GRAPhics[?] 'Identifier', <BVTH | QFBV >

Input Parameters Identifier: 'MyMeasurement'

Description The command allows the user to specify the Graphics Type to Visualize. If BVTH is used then BER versus Threshold graphics is visualized and the Horizontal Axis Scale (see :PLUGin:OLEVel:SHOW:HSCale[?]) can be visualized in terms of Logarithmic or Linear Scale. If QFBV is used then QBER versus Threshold graphics is visualized and the horizontal axis scale is always linear. The query returns current value.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OLEV:SHOW:GRAP 'MyMeasurement', BVTH

:PLUGin:OLEVel:SHOW:HSCale[?]

Syntax	:PLUGin:OLEVel:SHOW:HSCale[?] 'Identifier', <LOG | LIN >
Input Parameters	Identifier: 'MyMeasurement'
Description	The command allows the user to specify the horizontal axis scale (i.e. BER scale) if BER versus Threshold graphics is visualized (see :PLUGin:OLEVel:SHOW:GRAPhics[?]), however if QBER versus threshold graphics is visualized then the Horizontal Axis Scale is always linear. If LOG is used then the BER values are in terms of Logarithmic scale and if the LIN is used then the BER values are in terms of linear scale .The query returns current value. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:SHOW:HSC 'MyMeasurement',LOG

:PLUGin:OLEVel:SHOW:SGLegends[?]

Syntax	:PLUGin:OLEVel:SHOW:SGLegends 'Identifier', <0|1|OFF|ON> :PLUGin:OLEVel:SHOW:SGLegends? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement' <0|1|OFF|ON>: Specify the show/hide state for graph legends.
Return Range	0|1
Description	This command shows/hides the graph legends. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:SHOW:SGLegends 'MyMeasurement', 1

:PLUGin:OLEVel:RUN Subnode

This subnode has the following SCPI structure:

```
:RUN
├:MESSage?
├:PROGress?
├:STATus?
├:HISTory
│ ├[:STATe][?]
│ └:CLEar
└:LOG?
```

This subnode has the following commands:

Table 72

Name	Description under
:MESSage?	:PLUGin:OLEVel:RUN:MESSage? on page 337
:PROGress?	:PLUGin:OLEVel:RUN:PROGress? on page 337
:STATus?	:PLUGin:OLEVel:RUN:STATus? on page 337
:HISTory:[:STATe]	:PLUGin:OLEVel:RUN:HISTory[:STATe][?] on page 338
:HISTory:CLEar	:PLUGin:OLEVel:RUN:HISTory:CLEar on page 338
:LOG?	:PLUGin:OLEVel:RUN:LOG? on page 339

:PLUGin:OLEVel:RUN:MESSage?

Syntax :PLUGin:OLEVel:RUN:MESSage? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Description This command returns a string describing the state of an output level measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Suspended, Finished, Error, or Stopped.
This SCPI is for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:RUN:MESSage? 'MyMeasurement'

:PLUGin:OLEVel:RUN:PROGress?

Syntax :PLUGin:OLEVel:RUN:PROGress? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range 0 to 1

Description This command returns a number in the range of 0.0 to 1.0 to indicate the progress of an output level measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.
This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OLEV:RUN:PROGress? 'MyMeasurement'

:PLUGin:OLEVel:RUN:STATus?

Syntax :PLUGin:OLEVel:RUN:STATus? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range 0|1

Description This command returns the running status of an output level measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.
This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:OLEV:RUN:STATus? 'MyMeasurement'

:PLUGin:OLEVel:RUN:HISTory[:STATe][?]

Syntax	:PLUGin:OLEVel:RUN:HISTory[:STATe] 'Identifier', <0|1|OFF|ON>
	:PLUGin:OLEVel:RUN:HISTory[:STATe]? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	0|1
Description	This command enables/disables the storage of output level measurement results addressed by the measurement name identifier. This query returns the current setting. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:RUN:HISTory[:STATe] 'MyMeasurement',1

:PLUGin:OLEVel:RUN:HISTory:CLEar

Syntax	:PLUGin:OLEVel:RUN:HISTory:CLEar 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Description	This command deletes the output level measurement history addressed by the measurement name identifier. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:RUN:HISTory:CLEar 'MyMeasurement'

:PLUGin:OLEVel:RUN:LOG?

Syntax	:PLUGin:OLEVel:RUN:LOG? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Description	This command returns logs for the addressed measurement. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUGin:OLEVel:RUN:LOG? 'MyMeasurement' Output: #286 07/20/2015 18:27:55,Measurement.Output Level.Output Level 1,Info,"Start Measurement"

:PLUGin:JTOLerance Subnode

The jitter tolerance commands are used to determine the capacity of a serial data receiver to tolerate jitter. There are two modes: compliance or characterization. Jitter frequency is set at multiple test points. At each point, the jitter amplitude is modulated using various search algorithms to determine the level of jitter which causes the device to fail.

This subnode has the following SCPI structure:

```
:PLUGin
└─:JTOLerance
  ├─:INSTruments
  │ ├─:GENerator[?]
  │ └─:ANALyzer[?]
  ├─:BSETup
  │ ├─:TBERatio[?]
  │ ├─:CLEVel[?]
  │ ├─:FRTime[?]
  │ └─:ARTime[?]
  ├─:MSETup
  │ ├─:TEMPlate
  │ │ └─:FILE[?]
  │ ├─:FREQuency
  │ │ ├─:STARt[?]
  │ │ └─:STOp[?]
  │ ├─:MTPoints[?]
  │ ├─:NPOints[?]
  │ ├─:MODE[?]
  │ ├─:CMARgin[?]
  │ ├─:ALGorithm[?]
  │ ├─:BINary
  │ │ └─:SSIZe[?]
  │ ├─:LINear
  │ │ └─:SSIZe[?]
  │ ├─:LOG
  │ │ └─:SSIZe[?]
  │ └─:CLAuto[?]
  ├─:GRAPh
  │ ├─:TLIMits[?]
  │ ├─:TPOints[?]
  │ └─:CLIMits[?]
  ├─:FETCh
  │ └─:DATA?
  ├─:BREak
  ├─:CATalog?
  ├─:CONTinue
  ├─:NEW
  ├─:DELete
  ├─:RESet
  ├─:STARt
  ├─:STOP
  ├─:STEP
  └─:RUN
    ├─:HISTory
    │ ├─[:STATe][?]
    │ └─:CLEar
    ├─:LOG?
    ├─:MESSage?
    ├─:PROGress?
    └─[:STATus]?
```

This subnode has the following commands:

Table 73

Name	Description under
:INSTruments:GENerator[?]	:PLUGin:JTOLerance:INSTruments:GENerator[?] on page 343
:INSTruments:ANALyzer[?]	:PLUGin:JTOLerance:INSTruments:ANALyzer[?] on page 343
:BSETup:TBERatio[?]	:PLUGin:JTOLerance:BSETup:TBERatio[?] on page 344
:BSETup:CLEVel[?]	:PLUGin:JTOLerance:BSETup:CLEVel[?] on page 344
:BSETup:FRTime[?]	:PLUGin:JTOLerance:BSETup:FRTime[?] on page 345
:BSETup:ARTime[?]	:PLUGin:JTOLerance:BSETup:ARTime[?] on page 345
:MSETup:TEMPlate:FILE[?]	:PLUGin:JTOLerance:MSETup:TEMPlate:FILE[?] on page 346
:MSETup:FREQuency:STARt[?]	:PLUGin:JTOLerance:MSETup:FREQuency:STARt[?] on page 346
:MSETup:FREQuency:STOp[?]	:PLUGin:JTOLerance:MSETup:FREQuency:STOp[?] on page 347
:MSETup:MTPoints[?]	:PLUGin:JTOLerance:MSETup:MTPoints[?] on page 347
:MSETup:NPOints[?]	:PLUGin:JTOLerance:MSETup:NPOints[?] on page 347
:MSETup:MODE[?]	:PLUGin:JTOLerance:MSETup:MODE[?] on page 348
:MSETup:CMARgin[?]	:PLUGin:JTOLerance:MSETup:CMARgin[?] on page 348
:MSETup:ALGorithm[?]	:PLUGin:JTOLerance:MSETup:ALGorithm[?] on page 350
:MSETup:BINary:SSIZe[?]	:PLUGin:JTOLerance:MSETup:BINary:SSIZe[?] on page 352
:MSETup:LINear:SSIZe[?]	:PLUGin:JTOLerance:MSETup:LINear:SSIZe[?] on page 353

Name	Description under
:MSETup:LOG:SSIZe[?]	:PLUGin:JTOLerance:MSETup:LOG:SSIZe[?] on page 354
:MSETup:CLAuto[?]	:PLUGin:JTOLerance:MSETup:CLAuto[?] on page 355
:GRAPh:TLIMits[?]	:PLUGin:JTOLerance:GRAPh:TLIMits[?] on page 355
:GRAPh:TPOints[?]	:PLUGin:JTOLerance:GRAPh:TPOints[?] on page 355
:GRAPh:CLIMits[?]	:PLUGin:JTOLerance:GRAPh:CLIMits[?] on page 356
:FETCh:DATA?	:PLUGin:JTOLerance:FETCh:DATA? on page 356
:BREak	:PLUGin:JTOLerance:BREak on page 357
:CATalog?	:PLUGin:JTOLerance:CATalog? on page 357
:CONTinue	:PLUGin:JTOLerance:CONTinue on page 358
:NEW	:PLUGin:JTOLerance:NEW on page 358
:DELete	:PLUGin:JTOLerance:DELete on page 359
:RESet	:PLUGin:JTOLerance:RESet on page 359
:STARt	:PLUGin:JTOLerance:STARt on page 360
:STOP	:PLUGin:JTOLerance:STOP on page 360
:STEP	:PLUGin:JTOLerance:STEP on page 360
:RUN:HISTory[:STATe]	:PLUGin:JTOLerance:RUN:HISTory[:STATe][?] on page 361
:RUN:HISTory:CLEar	:PLUGin:JTOLerance:RUN:HISTory:CLEar on page 361
:RUN:LOG?	:PLUGin:JTOLerance:RUN:LOG? on page 362
:RUN:MESSage?	:PLUGin:JTOLerance:RUN:MESSage? on page 362
:RUN:PROGress?	:PLUGin:JTOLerance:RUN:PROGress? on page 362
:RUN[:STATus]?	:PLUGin:JTOLerance:RUN[:STATus]? on page 363

:PLUGin:JTOLerance:INSTruments:GENerator[?]

Syntax
:PLUGin:JTOLerance:INSTruments:GENerator 'identifier',<location-string>

:PLUGin:JTOLerance:INSTruments:GENerator? 'identifier'

Input Parameters
'identifier': Specify the measurement name.

<location-string>: Specify the location identifier or group name identifier.

Return Range
<location or group name>

Description
This command is used to select the pattern generator module/channel for the jitter tolerance measurement. The location-string refers to either a location name or a group name identifier associated with a measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example
:PLUG:JTOL:INST:GEN 'MyMeasurement','M1.DataOut1'

:PLUGin:JTOLerance:INSTruments:ANALyzer[?]

Syntax
:PLUGin:JTOLerance:INSTruments:ANALyzer 'identifier',<location-string>

:PLUGin:JTOLerance:INSTruments:ANALyzer? 'identifier'

Input Parameters
'identifier': Specify the measurement name.

<location-string>: Specify the location identifier or group name identifier.

Return Range
<location or group name>

Description
This command is used to select the analyzer (error detector) module/channel for the jitter tolerance measurement. The location-string refers to either a location name or a group name identifier associated with a measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example
:PLUG:JTOL:INST:ANAL 'MyMeasurement','M1.DataIn1'

:PLUGin:JTOLerance:BSETup:TBERatio[?]

Syntax	:PLUGin:JTOLerance:BSETup:TBERatio ‘identifier’, <NRf>
	:PLUGin:JTOLerance:BSETup:TBERatio? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name.
	<NRf>: Set the target BER.
Return Range	1E-15 to 1E-1
Description	This command is used to set the target BER of the measurement specified by the measurement name identifier.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:BSET:TBER ‘MyMeasurement’,1E-15

:PLUGin:JTOLerance:BSETup:CLEVel[?]

Syntax	:PLUGin:JTOLerance:BSETup:CLEVel ‘identifier’, <NRf>
	:PLUGin:JTOLerance:BSETup:CLEVel? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name.
	<NRf>: Set the percent confidence interval.
Return Range	0.1 to 99.9
Description	This command is used to set the percent confidence interval of the measurement specified by the measurement name identifier.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:BSET:CLEV ‘MyMeasurement’, 95%

:PLUGin:JTOLerance:BSETup:FRTime[?]

Syntax	:PLUGin:JTOLerance:BSETup:FRTime 'identifier', <NRf>
	:PLUGin:JTOLerance:BSETup:FRTime? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
	<NRf>: Set the frequency relax time.
Return Range	0 ms to 60 s
Description	This command is used to set the frequency relax time after a change in jitter modulation frequency for the measurement specified by the measurement name identifier.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:BSET:FRT 'MyMeasurement',30s

:PLUGin:JTOLerance:BSETup:ARTime[?]

Syntax	:PLUGin:JTOLerance:BSETup:ARTime 'identifier', <NRf>
	:PLUGin:JTOLerance:BSETup:ARTime? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
	<NRf>: Set the amplitude relax time.
Return Range	0 ms to 60 s
Description	This command is used to set the amplitude relax time after a change in jitter modulation amplitude for the measurement specified by the measurement name identifier.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:BSET:ART 'MyMeasurement', 30s

:PLUGin:JTOLerance:MSETup:TEMPlate:FILE[?]

Syntax	:PLUGin:JTOLerance:MSETup:TEMPlate:FILE 'identifier', <filepath-string> :PLUGin:JTOLerance:MSETup:TEMPlate:FILE? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <filepath-string>: Specify path/filename of template to load.
Return Range	Path/filename of current measurement template.
Description	This command is used to specify the path/filename of the measurement template to load for the measurement specified by the measurement name identifier. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUGin:JTOLerance:MSETup:TEMPlate:FILE "MyMeasurement","Current/default.jtt"

:PLUGin:JTOLerance:MSETup:FREQuency:STARt[?]

Syntax	:PLUGin:JTOLerance:MSETup:FREQuency:STARt 'identifier', <NRf> :PLUGin:JTOLerance:MSETup:FREQuency:STARt? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <NRf>: Set the start frequency.
Return Range	100 Hz to 500 MHz
Description	This command is used to set the start (minimum) modulation frequency for the measurement specified by the measurement name identifier. The default is 100 kHz. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:MSET:FREQ:STAR 'MyMeasurement', 10kHz

:PLUGin:JTOLerance:MSETup:FREQuency:STOp[?]

Syntax	:PLUGin:JTOLerance:MSETup:FREQuency:STOp 'identifier', <NRf> :PLUGin:JTOLerance:MSETup:FREQuency:STOp? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <NRf>: Set the stop frequency.
Return Range	100 Hz to 500 MHz
Description	This command is used to set the stop (maximum) modulation frequency for the measurement specified by the measurement name identifier. The default is 100 MHz. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:MSET:FREQ:STO 'MyMeasurement', 100MHz

:PLUGin:JTOLerance:MSETup:MTPoints[?]

Syntax	:PLUGin:JTOLerance:MSETup:MTPoints[?] 'identifier', <NRf>
Input Parameters	'identifier': Specify the measurement name.
Return Range	0 | 1 OFF | ON
Description	This command selects either template points or specified range. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUGin:JTOLerance:MSETup:MTPoints 'MyMeasurement'

:PLUGin:JTOLerance:MSETup:NPOints[?]

Syntax	:PLUGin:JTOLerance:MSETup:NPOints 'identifier', <NRf> :PLUGin:JTOLerance:MSETup:NPOints? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <NRf>: Set the number of measurement points.
Return Range	1 to 100

Description: This command is used to set the number of measurement points for the measurement specified by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example: :PLUG:JTOL:MSET:NPO ‘MyMeasurement’, 50

:PLUGin:JTOLerance:MSETup:MODE[?]

Syntax: :PLUGin:JTOLerance:MSETup:MODE ‘identifier’, <COMPliance|CHARacterize>

:PLUGin:JTOLerance:MSETup:MODE? ‘identifier’

Input Parameters: ‘identifier’: Specify the measurement name.

<COMPliance|CHARacterize>: Specify the measurement mode.

Return Range: COMPliance|CHARacterize

Description: This command is used to specify compliance or characterization as the measurement mode for the measurement specified by the measurement name identifier. Compliance is used for checking that a device passes the minimum margin required for compliance to a standard. Characterization tests a device for tolerance to jitter to establish the pass/fail amplitude at each measurement point.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example: :PLUG:JTOL:MSET:MODE ‘MyMeasurement’,CHAR

:PLUGin:JTOLerance:MSETup:CMARgin[?]

Syntax: :PLUGin:JTOLerance:MSETup:CMARgin ‘identifier’, <NRf>

:PLUGin:JTOLerance:MSETup:CMARgin? ‘identifier’

Input Parameters: ‘identifier’: Specify the measurement name.

<NRf>: Specify the compliance margin.

Return Range: -75% to 1000%

Description	This command is used to specify the compliance margin for the measurement specified by the measurement name identifier. The default is 0%. Compliance is used for checking that a device passes the minimum margin required for compliance to a standard. Characterization tests a device for tolerance to jitter to establish the pass/fail amplitude at each measurement point. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:MSET:CMAR ‘MyMeasurement’, 10

:PLUGin:JTOLerance:MSETup:ALGorithm[?]

Syntax

:PLUGin:JTOLerance:MSETup:ALGorithm ‘identifier’, <BINary|BDLInear|BULInear|DLINear|DLOGarithmic|ULINear|ULOGarithmic|ULL>

:PLUGin:JTOLerance:MSETup:ALGorithm? ‘identifier’

Input Parameters

‘identifier’: Specify the measurement name.

<BINary|BDLInear|BULInear|DLINear|DLOGarithmic|ULINear|ULOGarithmic|ULL>: Specify the measurement algorithm.

Return Range

BINary|BDLinear|BULinear|DLINear|DLOGarithmic|ULINear|ULOGarithmic|ULL

Description

This command is used to specify the measurement algorithm during characterization measurements for the measurement specified by the measurement name identifier.

BINary: The Binary algorithm uses a variable step size to find the highest passing SJ amplitude at each SJ frequency in the template. Starting at the maximum SJ amplitude determined by the upper limit in the template file or the instrument limit, whichever is lower, followed by the minimum SJ amplitude determined by the lower limit in the template file, the Binary algorithm then sets subsequent SJ amplitudes at the logarithmic midpoint between the highest passing and lowest failing SJ amplitudes. A test point is determined to be a passing point if the BER measured is below the BER threshold set by the user.

BDLinear: The Binary + Down Linear algorithm is the same as the Binary algorithm followed by the Down Linear algorithm. Refer to the descriptions for Binary and Down Linear.

BULinear: The Binary + Up Linear algorithm is the same as the Binary algorithm followed by the Up Linear algorithm. Refer to the descriptions for Binary and Up Linear.

DLINear: The Down Linear search algorithm starts at the maximum jitter value determined by the template.

If a BER measurement has errors above the measurement threshold, the jitter amplitude is adjusted lower by the linear step size.

This algorithm exits when the BER is measured below the measurement threshold BER, or if the last amplitude measurement point was at the minimum jitter value in the template.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

DLOGarithmic: The Down Logarithmic search algorithm starts at the maximum jitter value determined by the template.

If a BER measurement has errors above the measurement threshold, the jitter amplitude is adjusted lower by the coefficient calculated from the logarithmic step size. For example, if the coefficient is 10%, then the next amplitude equals the previous amplitude minus 10% of the previous amplitude.

This algorithm exits when the BER is measured below the measurement threshold BER, or if the last amplitude measurement point was at the minimum jitter value in the template.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

ULINear: The Up Linear search algorithm starts at the minimum jitter value determined by the template.

If a BER measurement has errors below the measurement threshold, the jitter amplitude is adjusted higher by the linear step size.

This algorithm exits when BER is measured above the measurement threshold BER, or if the last amplitude measurement point was at the max jitter value in the template.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

ULOGarithmic: The Up Logarithmic search algorithm starts at the minimum jitter value determined by the template.

If a BER measurement has errors below the measurement threshold, the jitter amplitude is adjusted higher by the coefficient calculated from the logarithmic step size. For example, if the coefficient is 10%, then the next amplitude equals the previous amplitude plus 10% of the previous amplitude.

This algorithm exits when the BER is measured above the measurement threshold BER, or if the last amplitude measurement point was at the maximum jitter value determined by the template.

ULL: The Up Log + Linear search algorithm performs the same algorithm as Up Logarithmic, but in addition, it returns to the last passing amplitude and steps linearly up until it reaches a fail point. The algorithm continues to increase the SJ amplitude until failure, even if it surpassed the original failed point.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example :PLUG:JTOL:MSET:ALG ‘MyMeasurement’,BIN

:PLUGin:JTOLerance:MSETup:BINary:SSIZe[?]

Syntax :PLUGin:JTOLerance:MSETup:BINary:SSIZe ‘identifier’, <NRf>

:PLUGin:JTOLerance:MSETup:BINary:SSIZe? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

<NRf>: Set the step size.

Return Range 2 mUI to 5 UI

Description This command is used to set the step size for the binary algorithm for the measurement specified by the measurement name identifier. The step size defines the exit criteria for the algorithm. The Binary search algorithm stops once its step size falls below this user defined Step Size.

The Binary Step Size applies to the Binary algorithm and the binary portions of the Binary + Down Linear and Binary + Up Linear algorithms.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example :PLUG:JTOL:MSET:BIN:SSIZ ‘MyMeasurement’, 100mUI

:PLUGin:JTOLerance:MSETup:LINear:SSIZe[?]

Syntax	:PLUGin:JTOLerance:MSETup:LINear:SSIZe ‘identifier’, <NRf> :PLUGin:JTOLerance:MSETup:LINear:SSIZe? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name. <NRf>: Set the step size.
Return Range	5 mUI to 100 UI
Description	This command is used to set the step size for each step of the Up Linear and Down Linear algorithms for the measurement specified by the measurement name identifier. When Down Linear is enabled as the search algorithm, the measurement starts from the maximum (which depends on the jitter frequency). A step size of 100 mUI, for example, may result in a sequence of 100 UI, 99.9 UI, 99.8 UI and so on. The test for one frequency stops when the BER limit is met or zero amplitude is reached. When Up Linear is enabled as the search algorithm, the measurement starts from the minimum. A step size of 100 mUI, for example, will result in a sequence of 0.1 UI, 0.2 UI, 0.3 UI and so on. The test for one frequency stops when the BER limit is exceeded or the maximum amplitude is reached. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:MSET:LIN:SSIZe ‘MyMeasurement’, 100mUI

:PLUGin:JTOLerance:MSETup:LOG:SSIZe[?]

Syntax

:PLUGin:JTOLerance:MSETup:LOG:SSIZe ‘identifier’, <NRf>

:PLUGin:JTOLerance:MSETup:LOG:SSIZe? ‘identifier’

Input Parameters

‘identifier’: Specify the measurement name.

<NRf>: Set the step size.

Return Range

0.1% to 50%

Description

This command is used to set the step size for each step of the Up Logarithmic and Down Logarithmic algorithms for the measurement specified by the measurement name identifier.

When Down Logarithmic is enabled as the search algorithm, the measurement starts from the maximum (which depends on the jitter frequency). A percentage of 50%, for example, may result in a sequence of 100 UI, 50 UI, 25 UI and so on. The test for one frequency stops when the BER limit is met or the specified minimum amplitude is reached.

When Up Logarithmic is enabled as the search algorithm, the measurement starts from the specified minimum amplitude. A percentage of 50%, for example, may result in a sequence of 0.1 UI, 0.15 UI, 0.23 UI and so on. The test for one frequency stops when the BER limit is crossed or the maximum amplitude (which depends on the jitter frequency) is reached.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example

:PLUG:JTOL:MSET:LOG:SSIZe ‘MyMeasurement’, 25

:PLUGin:JTOLerance:MSETup:CLAuto[?]

Syntax	:PLUGin:JTOLerance:MSETup:CLAuto ‘identifier’, <ON|OFF|1|0>
	:PLUGin:JTOLerance:MSETup:CLAuto? ‘identifier’
Input Parameters	‘identifier’: ‘Specify the measurement name.
	<ON|OFF|1|0>: Enable/disable auto loop bandwidth.
Return Range	1|0
Description	This command sets the external CDR loop bandwidth selection during a Jitter Tolerance measurement. If AUTO state is ON, will set the CDR LBW based on the jitter frequency. If AUTO state is OFF, CDR LBW is not altered during the measurement. The query returns the current auto mode state.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:MSET:CLA ‘MyMeasurement’,ON

:PLUGin:JTOLerance:GRAPh:TLIMits[?]

Syntax	:PLUGin:JTOLerance:GRAPh:TLIMits ‘identifier’, <ON|OFF|1|0>
	:PLUGin:JTOLerance:GRAPh:TLIMits? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name.
	<ON|OFF|1|0>: Enable/disable template limit lines.
Return Range	1|0
Description	This command enables/disables the template limits in the graph for the measurement specified by the measurement name identifier. These limits show the search range amplitude (UI) during characterization defined using the template editor in the M8070A software interface.
	This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:GRAP:TLIM ‘MyMeasurement’, ON

:PLUGin:JTOLerance:GRAPh:TPOints[?]

Syntax	:PLUGin:JTOLerance:GRAPh:TPOints ‘identifier’, <ON|OFF|1|0>
	:PLUGin:JTOLerance:GRAPh:TPOints? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name.
	<ON|OFF|1|0>: Enable/disable template test points.

Return Range	1|0
Description	This command enables/disables the template measurement points in the graph for the measurement specified by the measurement name identifier. These test points are defined using the template editor in the M8070A software interface. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:GRAP:TPO ‘MyMeasurement’, ON

:PLUGin:JTOLerance:GRAPh:CLIMits[?]

Syntax	:PLUGin:JTOLerance:GRAPh:CLIMits ‘identifier’, <ON|OFF|1|0> :PLUGin:JTOLerance:GRAPh:CLIMits? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name. <ON|OFF|1|0>: Enable/disable compliance limits.
Return Range	1|0
Description	This command enables/disables the compliance limits in the graph for the measurement specified by the measurement name identifier. These limits are defined using the template editor in the M8070A software interface. This SCPI is for M8041A, M8051A, M8061A and M8062A.
Example	:PLUG:JTOL:GRAP:CLIM ‘MyMeasurement’, ON

:PLUGin:JTOLerance:FETCh:DATA?

Syntax	:PLUGin:JTOLerance:FETCh:DATA? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name.
Description	This command returns the jitter tolerance measurement results addressed by the measurement name identifier. "M1.DataIn1",100000,0.01,3000000000,0,0,"PASS" The first item in the comma separated list is the name of the location which is “M1.DataIn1”. The next values are Frequency, Amplitude, Number of Bits, Number of Errors, BER and Pass/Fail Results and are repeated in the same order for subsequent measured points. This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example :PLUG:JTOL:FETC:DATA? ‘MyMeasurement’

:PLUGin:JTOLerance:BREak

Syntax :PLUGin:JTOLerance:BREak ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to break/pause.

Description This command breaks/pauses a measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example breaks a jitter tolerance measurement with the measurement name identifier called ‘MyMeasurement’:

:PLUG:JTOL:BRE ‘MyMeasurement’

:PLUGin:JTOLerance:CATalog?

Syntax :PLUGin:JTOLerance:CATalog?

Description This command returns a list of all created jitter tolerance measurement names currently available for measuring.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example Assume the following is a list of created jitter tolerance measurement names:

:PLUG:JTOL:NEW ‘JTOL_1’
:PLUG:JTOL:NEW ‘JTOL_2’
:PLUG:JTOL:NEW ‘JTOL_3’

The command and returned list would look like the following:
:PLUG:JTOL:CAT?
“JTOL_1,JTOL_2,JTOL_3”

:PLUGin:JTOLerance:CONTinue

Syntax :PLUGin:JTOLerance:CONTinue 'identifier'

Input Parameters 'identifier': Specify the measurement name to continue.

Description This command allows you to continue a jitter tolerance measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example continues a halted jitter tolerance measurement addressed with the measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:CONT 'MyMeasurement'

:PLUGin:JTOLerance:NEW

Syntax :PLUGin:JTOLerance:NEW 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command creates a new jitter tolerance measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example creates a jitter tolerance measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:NEW 'MyMeasurement'

NOTE Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use "0" as a parameter input in this command.

Example- :PLUG:JTOL:NEW 'MyMeasurement',0

:PLUGin:JTOLerance:DELete

Syntax :PLUGin:JTOLerance:DELete ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to delete.

Description This command deletes a previously created jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example deletes a jitter tolerance measurement addressed by the measurement name identifier called ‘MyMeasurement’:

:PLUG:JTOL:DEL ‘MyMeasurement’

:PLUGin:JTOLerance:RESet

Syntax :PLUGin:JTOLerance:RESet ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to reset.

Description This command resets a jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example resets a jitter tolerance measurement addressed with the measurement name identifier called ‘MyMeasurement’:

:PLUG:JTOL:RES ‘MyMeasurement’

:PLUGin:JTOLerance:STARt

Syntax :PLUGin:JTOLerance:STARt ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to start.

Description This command starts a jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example starts a jitter tolerance measurement addressed with the measurement name identifier called ‘MyMeasurement’:

:PLUG:JTOL:STAR ‘MyMeasurement’

:PLUGin:JTOLerance:STOP

Syntax :PLUGin:JTOLerance:STOP ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to stop.

Description This command stops a jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example stops a jitter tolerance measurement with the measurement name identifier called ‘MyMeasurement’:

:PLUG:JTOL:STOP ‘MyMeasurement’

:PLUGin:JTOLerance:STEP

Syntax :PLUGin:JTOLerance:STEP ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to step.

Description This command allows you to step through a jitter tolerance measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example

The following example triggers a single jitter tolerance measurement step of the measurement addressed with the measurement name identifier called ‘MyMeasurement’:

:PLUG:JTOL:STEP ‘MyMeasurement’

:PLUGin:JTOLerance:RUN:HISTory[:STATe][?]

Syntax

:PLUGin:JTOLerance:RUN:HISTory[:STATe] ‘identifier’,<0|1>

:PLUGin:JTOLerance:RUN:HISTory[:STATe]? ‘identifier’

Input Parameters

‘identifier’: Specify the measurement name.

Return Range

0|1

Description

This command enables/disables the storage of jitter tolerance measurement results addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example

The following example enables storage of jitter tolerance measurement results with the measurement name identifier called ‘MyMeasurement’:

:PLUG:JTOL:RUN:HIST ‘MyMeasurement’, 1

:PLUGin:JTOLerance:RUN:HISTory:CLEar

Syntax

:PLUGin:JTOLerance:RUN:HISTory:CLEar ‘identifier’

Input Parameters

‘identifier’: Specify the measurement name.

Description

This command deletes the stored jitter tolerance measurement history addressed by the measurement name identifier.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example

The following example deletes the stored jitter tolerance measurement history with the measurement name identifier called ‘MyMeasurement’:

:PLUG:JTOL:RUN:HIST:CLE ‘MyMeasurement’

:PLUGin:JTOLerance:RUN:LOG?

Syntax :PLUGin:JTOLerance:RUN:LOG? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns logs for the addressed measurement.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example returns logs for the measurement name identifier called ‘MyMeasurement’:

:PLUG:JTOL:RUN:LOG? ‘MyMeasurement’

Output:

#294 07/20/2015 18:24:53,Measurement.Jitter Tolerance.Jitter Tolerance 1,Info,"Start Measurement"

:PLUGin:JTOLerance:RUN:MESSage?

Syntax :PLUGin:JTOLerance:RUN:MESSage? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns a string describing the state of an jitter tolerance measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Suspended, Finished, Error, or Stopped.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example The following example returns the state of a jitter tolerance measurement with the measurement name identifier called ‘MyMeasurement’:

:PLUG:JTOL:RUN:MESS? ‘MyMeasurement’
Running

:PLUGin:JTOLerance:RUN:PROGress?

Syntax :PLUGin:JTOLerance:RUN:PROGress? ‘identifier’

Input Parameters ‘identifier’: Specify measurement name.

Return Range 0.0 to 1.0

Description: This command returns a number in the range of 0.0 to 1.0 to indicate the progress of a jitter tolerance measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example: The following example returns the progress of a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:RUN:PROG? 'MyMeasurement'
0.51
```

:PLUGin:JTOLerance:RUN[:STATus]?

Syntax: :PLUGin:JTOLerance:RUN[:STATus]? 'identifier'

Input Parameters: 'identifier': Specify measurement name.

Return Range: 0|1

Description: This command returns the running status of a jitter tolerance measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is for M8041A, M8051A, M8061A and M8062A.

Example: The following example returns the running status of a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:RUN? 'MyMeasurement'
1
```

:PLUGin:EDIagram Subnode

The M8070A system software provides quick design analysis with the eye diagram capability. The measurement allows a quick check for the DUT's signal output and determines the signal quality. The eye contour lines display the measured eye at a deeper BER level, for accurate results. The eye diagram generates a three dimensional graph of the bit error rate (BER). This measurement helps in determining and analyzing the quality of the DUT's signal output. The eye diagram results comprise of voltage(y), time(x) and BER(z). The measurement comes with the set of Acquisition and Evaluation Parameters which are used to configure the measurement run. The Acquisition parameters can only be configured before the measurement starts (i.e. during the measurement run you cannot change the value of Acquisition parameter); however the Evaluation parameters can be configured any time even after the measurement has been run.

This subnode has the following SCPI structure:

```
:EDIagram
├─ :BREak
├─ :CATalog?
├─ :CONTinue
├─ :DELete
├─ :NEW
├─ :RESet
├─ :STARt
├─ :STEP
├─ :STOP
├─ :ACQuisition
│   └─ ...
├─ :EVALuation
│   └─ ...
├─ :FETCh[:RESult]
│   └─ ...
├─ :RUN
│   └─ ...
└─ :SHOW
    └─ ...
```

This subnode has the following commands and subnodes:

Table 74

Name	Description under
:BREak	:PLUGin:EDIagram:BREak on page 366
:CATalog?	:PLUGin:EDIagram:CATalog? on page 366
:CONTinue	:PLUGin:EDIagram:CONTinue on page 366
:DELete	:PLUGin:EDIagram:DELete on page 367
:NEW	:PLUGin:EDIagram:NEW on page 367
:RESet	:PLUGin:EDIagram:RESet on page 368
:STARt	:PLUGin:EDIagram:STARt on page 368
:STEP	:PLUGin:EDIagram:STEP on page 368
:STOP[	:PLUGin:EDIagram:STOP on page 369
Subnodes	
:ACQuisition:	:PLUGin:EDIagram:ACQuisition Subnode on page 370
:EVALuation	:PLUGin:EDIagram:EVALuation Subnode on page 373
:FETCh[:RESult]	:PLUGin:EDIagram:FETCh[:RESult] Subnode on page 375
:RUN	:PLUGin:EDIagram:RUN Subnode on page 381
:SHOW	:PLUGin:EDIagram:SHOW Subnode on page 384

:PLUGin:EDIagram:BREak

Syntax :PLUGin:EDIagram:BREak ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to break/pause.

Description This command breaks/pauses a measurement addressed by the measurement name identifier.
This SCPI is for M8041A, M8051A and M8062A.

Example The following example breaks an eye diagram measurement with the measurement name identifier called ‘MyMeasurement’:
:PLUG:EDI:BRE ‘MyMeasurement’

:PLUGin:EDIagram:CATalog?

Syntax :PLUGin:EDIagram:CATalog?

Description This command returns a list of all created eye diagram measurement names currently available for measuring.

This SCPI is for M8041A, M8051A and M8062A.

Example Assume three eye diagram measurements named as:EDIagram_1, :EDIagram_2 and :EDIagram_3 are created using NEW command:
:PLUG:EDI:NEW ‘EDIagram_1’
:PLUG:EDI:NEW ‘EDIagram_2’
:PLUG:EDI:NEW ‘EDIagram_3’
The command and returned list would look like the following:
:PLUG:EDI:CAT?
“EDIagram_1”,”EDIagram_2”,”EDIagram_3”

:PLUGin:EDIagram:CONTinue

Syntax :PLUGin:EDIagram:CONTinue ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to continue.

Description This command allows you to continue an eye diagram measurement, addressed by the measurement name identifier, which was stopped using the BREak command.
This SCPI is for M8041A, M8051A and M8062A.

Example The following example continues a halted eye diagram measurement addressed with the measurement name identifier called ‘MyMeasurement’:
:PLUG:EDI:CONT ‘MyMeasurement’

:PLUGin:EDIagram:DELete

Syntax :PLUGin:EDIagram:DELete ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to delete.

Description This command deletes a previously created eye diagram measurement addressed by the measurement name identifier.
This SCPI is for M8041A, M8051A and M8062A.

Example The following example deletes an eye diagram measurement addressed by the measurement name identifier called ‘MyMeasurement’:
:PLUG:EDI:DEL ‘MyMeasurement’

:PLUGin:EDIagram:NEW

Syntax :PLUGin:EDIagram:NEW ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command creates a new eye diagram measurement name identifier.
This SCPI is for M8041A, M8051A and M8062A.

Example The following example creates an eye diagram measurement name identifier called ‘MyMeasurement’:
:PLUG:EDI:NEW ‘MyMeasurement’

NOTE Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use “0” as a parameter input in this command.

Example- :PLUG:EDI:NEW ‘MyMeasurement’,0

:PLUGin:EDIagram:RESet

Syntax	:PLUGin:EDIagram:RESet 'identifier'
Input Parameters	'identifier': Specify the measurement name to reset.
Description	This command resets an eye diagram measurement addressed by the measurement name identifier. This SCPI is for M8041A, M8051A and M8062A.
Example	The following example resets an eye diagram measurement addressed with the measurement name identifier called 'MyMeasurement': :PLUG:EDI:RES 'MyMeasurement'

:PLUGin:EDIagram:STARt

Syntax	:PLUGin:EDIagram:STARt 'identifier'
Input Parameters	'identifier': Specify the measurement name to start.
Description	This command starts an eye diagram measurement addressed by the measurement name identifier. This SCPI is for M8041A, M8051A and M8062A.
Example	The following example starts an eye diagram measurement addressed with the measurement name identifier called 'MyMeasurement': :PLUG:EDI:STAR 'MyMeasurement'

:PLUGin:EDIagram:STEP

Syntax	:PLUGin:EDIagram:STEP 'identifier'
Input Parameters	'identifier': Specify the measurement name to step.
Description	This command allows you to step through an eye diagram measurement, addressed by the measurement name identifier, that was stopped using the BREak command. This SCPI is for M8041A, M8051A and M8062A.

Example The following example triggers a single eye diagram measurement step of the measurement addressed with the measurement name identifier called 'MyMeasurement':
:PLUG:EDI:STEP 'MyMeasurement'

:PLUGin:EDIagram:STOP

Syntax :PLUGin:EDIagram:STOP 'identifier'

Input Parameters 'identifier': Specify the measurement name to stop.

Description This command stops an eye diagram measurement addressed by the measurement name identifier.
This SCPI is for M8041A, M8051A and M8062A.

Example The following example stops an eye diagram measurement with the measurement name identifier called 'MyMeasurement':
:PLUG:EDI:STOP 'MyMeasurement'

:PLUGin:EDIagram:ACQuisition Subnode

This subnode has the following SCPI structure:

```
:ACQuisition
├ :ALOCation[?]
├ :NEYEs[?]
├ :PERSistence[?]
└ :TIME[?]
```

This subnode has the following commands:

Table 75

Name	Description under
:ALOCation[?]	:PLUGin:EDIagram:ACQuisition:ALOCation[?] on page 371
:NEYEs[?]	:PLUGin:EDIagram:ACQuisition:NEYEs[?] on page 371
:PERSistence[?]	:PLUGin:EDIagram:ACQuisition:PERSistence [?] on page 372
:TIME[?]	:PLUGin:EDIagram:ACQuisition:TIME[?] on page 372

:PLUGin:EDIagram:ACQuisition:ALOCation[?]

Syntax	:PLUGin:EDIagram:ACQuisition:ALOCation 'Identifier', < location-string> :PLUGin:EDIagram:ACQuisition:ALOCation? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement' <location-string>: ‘M*.DataIn*’
Description	This command sets the location string against which the data acquisition is performed. The query returns the current location string against which measurement is configured for the data acquisition. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:ACQ:ALOC 'MyMeasurement',"M1.DataIn1"

:PLUGin:EDIagram:ACQuisition:NEYEs[?]

Syntax	:PLUGin:EDIagram:ACQuisition:NEYEs 'Identifier', TDZ | ODF :PLUGin:EDIagram:ACQuisition:NEYEs? 'Identifier',
Input Parameters	Identifier: 'MyMeasurement'
Range	TDZ | ODF
Description	This command sets the number eyes to be displayed in UI. It can be either 1.5 or 2.0. The query returns current value. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:ACQ:NEYEs 'MyMeasurement', TDZ

:PLUGin:EDIagram:ACQuisition:PERSistence[?]

Syntax	:PLUGin:EDIagram:ACQuisition:PERSistence 'Identifier', <INF | FTIM>
	:PLUGin:EDIagram:ACQuisition:PERSistence? 'Identifier',
Input Parameters	Identifier: 'MyMeasurement'
Return Range	INF | FTIM
Description	This command sets the criteria for persistence. Persistence could be infinite or fixed time.
	The query returns the current setting.
	This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:ACQ:PERS 'MyMeasurement',INF

:PLUGin:EDIagram:ACQuisition:TIME[?]

Syntax	:PLUGin:EDIagram:ACQuisition:TIME 'Identifier', <NRf>
	:PLUGin:EDIagram:ACQuisition:TIME? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	50s to 216ks
Description	This command allows the user to specify the persistence time.
	The query returns current setting.
	This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:ACQ:TIME 'MyMeasurement', 80

:PLUGin:EDIagram:EVALuation Subnode

This subnode has the following SCPI structure:

```
:EVALuation
├─ :BERThresh[?]
├─ :BERThresh:THReshold[?]
└─ :TTIME[?]
```

This subnode has the following commands:

Table 76

Name	Description under
:BERThresh[?]	:PLUGin:EDIagram:EVALuation:BERThresh[?] on page 373
:BERThresh:THReshold[?]	:PLUGin:EDIagram:EVALuation:BERThresh:THReshold[?] on page 374
:TTIMe[?]	:PLUGin:EDIagram:EVALuation:TTIMe[?] on page 374

:PLUGin:EDIagram:EVALuation:BERThresh[?]

Syntax

:PLUGin:EDIagram:EVALuation:BERThresh 'Identifier', <ZERR | BTHR>

:PLUGin:EDIagram:EVALuation:BERThresh? 'Identifier'

Input Parameters

Identifier: 'MyMeasurement'

Return Range

ZERR | BTHR

Description

This command tells how to calculate measurement results, such as, eye height, eye width and JPPeak either using '0 errors' (ZERO) or a specific 'BER Threshold' (THReshold). In the latter case, numeric value of the threshold is specified with the :PLUGin:EDIagram:EVALuation:BERThresh:THReshold[?] on page 374 command.

The query returns either 0 errors (ZERR) or BER threshold (BTHR).

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:EVAL:BERT 'MyMeasurement', BTHR

:PLUGin:EDIagram:EVALuation:BERThresh:THReshold[?]

Syntax :PLUGin:EDIagram:EVALuation:BERThreshold 'Identifier', <NRf>

:PLUGin:EDIagram:EVALuation:BERThreshold? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Range 1E-12 to 1E-1

Description This command allows the user to specify BER Threshold value.

The query returns the current BER Threshold value.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:EVAL:BERT 'MyMeasurement', 1E-10

:PLUGin:EDIagram:EVALuation:TTIMe[?]

Syntax :PLUGin:EDIagram:EVALuation:TTIMe 'Identifier', <TEIG | TNIN>

:PLUGin:EDIagram:EVALuation:TTIMe? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Range TEIG | TNIN

Description This command allows the user to specify the transition time. It can be either 10/90 or 20/80.

The query returns the current setting.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:EVAL:TTIM 'MyMeasurement', TNIN

:PLUGin:EDIagram:FETCh[:RESult] Subnode

This subnode has the following SCPI structure:

```
:FETCh
└[:RESult]
  ├ :AMPLitude?
  ├ :CVOLtage?
  ├ :DCDistortion?
  ├ :FALLtime?
  ├ :HEIGht?
  ├ :HILevel?
  ├ :JPPeak?
  ├ :JRMSquare?
  ├ :LOLevel?
  ├ :RISetime?
  ├ :SCOunt?
  ├ :SNRatio?
  └ :WIDTh?
```

This subnode has the following commands and subnodes:

Table 77

Name	Description under
AMPLitude?	:PLUGin:EDIagram:FETCh[:RESult]:AMPLitude? on page 376
:CVOLtage?	:PLUGin:EDIagram:FETCh[:RESult]:CVOLtage? on page 377
:DCDistortion?	:PLUGin:EDIagram:FETCh[:RESult]:DCDistortion? on page 377
:FALLtime?	:PLUGin:EDIagram:FETCh[:RESult]:FALLtime? on page 377

Name	Description under
:HEIGht?	:PLUGin:EDIagram:FETCh[:RESult]:HEIGht? on page 378
:HILevel?	:PLUGin:EDIagram:FETCh[:RESult]:HILevel? on page 378
:JPPeak?	:PLUGin:EDIagram:FETCh[:RESult]:JPPeak? on page 378
:JRMSquare?	:PLUGin:EDIagram:FETCh[:RESult]:JRMSquare? on page 379
:LOLevel?	:PLUGin:EDIagram:FETCh[:RESult]:LOLevel? on page 379
:RISetime?	:PLUGin:EDIagram:FETCh[:RESult]:RISetime? on page 379
:SCOunt?	:PLUGin:EDIagram:FETCh[:RESult]:SCOunt? on page 380
:SNRatio?	:PLUGin:EDIagram:FETCh[:RESult]:SNRatio? on page 380
:WIDTh?	:PLUGin:EDIagram:FETCh[:RESult]:WIDTh? on page 380

:PLUGin:EDIagram:FETCh[:RESult]:AMPLitude?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:AMPLitude? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the eye amplitude which is the difference between the mean logic 1 level values and the mean logic 0 level values in a histogram of an eye diagram.
This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:FETC:AMPL? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:CVOLtage?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:CVOLtage? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns the cross voltage which is the measurement of the crossing point level in relation to the logic 1 level and logic 0 level.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:FETC:CVOL? "MyMeasurement"

:PLUGin:EDIagram:FETCh[:RESult]:DCDistortion?

Syntax ::PLUGin:EDIagram:FETCh[:RESult]:DCDistortion? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns the value of DC Distortion calculated after a run. This value is the difference between the period of a 1 bit and a 0 bit.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:FETC:DCD? ‘MyMeasurement’

:PLUGin:EDIagram:FETCh[:RESult]:FALLtime?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:FALLtime? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement nameDescription

Description This command returns the value of the fall time. Fall time is a measurement of the mean transition time of the data on the downward slope of an eye diagram. This transition time is either 2080 or 1090. See :PLUGin:EDIagram:EVALuation:TTIMe[?] on page 374.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:FETC:FALL? ‘MyMeasurement’

:PLUGin:EDIagram:FETCh[:RESult]:HEIGht?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:HEIGht? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the value of the eye height which is a measurement of the vertical opening of an eye diagram. This opening is affected by the BER threshold that is set by the BER threshold command. See :PLUGin:EDIagram:EVALuation:BERThresh[?] on page 373

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:FETC:HEIG? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:HILevel?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:HILevel? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the measurement of the mean value of the logical 1 in the eye diagram. This is directly affected by the values set for Eye Boundaries. One Level is a measure of the mean value of the logical 1 of an eye diagram.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:FETC:HIL? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:JPPeak?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:JPPeak? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns Jitter P-P value. It is the full width of the eye diagram at the eye crossing point. This is affected by the BER threshold that is set by the BER threshold command. See :PLUGin:EDIagram:EVALuation:BERThresh:THReshold[?] on page 374

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:FETC:JPP? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:JRMSquare?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:JRMSquare? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns the Jitter Root-Mean Square value. It is the standard deviation of the normal distribution of random jitter. It is dependent on the BER Threshold.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:FETC:JRMS? ‘MyMeasurement’

:PLUGin:EDIagram:FETCh[:RESult]:LOLevel?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:LOLevel? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns the measurement of the mean value of the logical 1 in the eye diagram. This is directly affected by the values set for eye boundaries.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:FETC:LOL? ‘MyMeasurement’

:PLUGin:EDIagram:FETCh[:RESult]:RISetime?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:RISetime? ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command returns rise time which is a measurement of the mean transition time of the data on the upward slope of an eye diagram. This transition time is either 20/80 or 10/90. See :PLUGin:EDIagram:EVALuation:TTIMe[?] on page 374

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:FETC:RIS? ‘MyMeasurement’

:PLUGin:EDIagram:FETCh[:RESult]:SCOunt?

Syntax	:PLUGin:EDIagram:FETCh[:RESult]:SCOunt? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
Description	This command returns the sample count. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:FETC:SCO? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:SNRatio?

Syntax	:PLUGin:EDIagram:FETCh[:RESult]:SNRatio? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
Description	This command returns signal-to-noise ratio which is a measurement of the signal difference between 1 level and 0 level in relation to the rms value of 1-level noise + rms value of 0-level noise. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:FETC:SNR? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:WIDTh?

Syntax	:PLUGin:EDIagram:FETCh[:RESult]:WIDTh? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
Description	This command returns the calculated eye width which is horizontal measurement of the eye opening at a specified BER Threshold.It can be either at the eye crossing point (CROssing) or custom defined (CUStom). This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:FETC:WIDT? 'MyMeasurement'

:PLUGin:EDIagram:RUN Subnode

This subnode has the following SCPI structure:

```
:RUN
├:MESSage?
├:PROGress?
├:STATus?
├:HISTory
│ ├[:STATe][?]
│ └:CLEar
└:LOG?
```

This subnode has the following commands:

Table 78

Name	Description under
:MESSage?	:PLUGin:EDIagram:RUN:MESSage? on page 382
:PROGress?	:PLUGin:EDIagram:RUN:PROGress? on page 382
:STATus?	:PLUGin:EDIagram:RUN:STATus? on page 382
:HISTory:[:STATe]	:PLUGin:EDIagram:RUN:HISTory[:STATe][?] on page 383
:HISTory:CLEar	:PLUGin:EDIagram:RUN:HISTory:CLEar on page 383
:LOG?	:PLUGin:EDIagram:RUN:LOG? on page 383

:PLUGin:EDIagram:RUN:MESSage?

Syntax	:PLUGin:EDIagram:RUN:MESSage? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Description	This command returns a string describing the state of an eye diagram measurement addressed by the measurement name identifier. Possible states include Not Started, Running, Suspended, Finished, Error, or Stopped. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:RUN:MESS? 'MyMeasurement'

:PLUGin:EDIagram:RUN:PROGress?

Syntax	:PLUGin:EDIagram:RUN:PROGress? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	0 to 1
Description	This command returns a number in the range of 0.0 to 1.0 to indicate the progress of an eye diagram measurement addressed by the measurement name identifier. 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished. This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:RUN:PROG? 'MyMeasurement'

:PLUGin:EDIagram:RUN:STATus?

Syntax	:PLUGin:EDIagram:RUN:STATus? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	0|1
Description	This command returns the running status of an eye diagram measurement addressed by the measurement name identifier. 0 indicates the measurement is not running and a 1 indicates the measurement is running. This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:RUN:STAT? 'MyMeasurement'

:PLUGin:EDIagram:RUN:HISTory[:STATe][?]

Syntax :PLUGin:EDIagram:RUN:HISTory[:STATe] 'Identifier', <0|1|ON|OFF>

:PLUGin:EDIagram:RUN:HISTory[:STATe]? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range 0|1

Description This command enables/disables the storage of eye diagram measurement results addressed by the measurement name identifier.

This query returns the current setting.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:RUN:HIST 'MyMeasurement', 1

:PLUGin:EDIagram:RUN:HISTory:CLEar

Syntax :PLUGin:EDIagram:RUN:HISTory:CLEar 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Description This command deletes the eye diagram measurement history addressed by the measurement name identifier.
This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:RUN:HIST:CLE 'MyMeasurement'

:PLUGin:EDIagram:RUN:LOG?

Syntax :PLUGin:EDIagram:RUN:LOG? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Description This command returns logs for the addressed measurement.
This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:RUN:LOG? 'MyMeasurement'

Output:

#286 01/20/2016 18:27:55,Measurement.Eye Diagram.Eye Diagram 1,Info,"Start Measurement"

:PLUGin:EDIagram:SHOW Subnode

This subnode has the following SCPI structure:

```
:SHOW
├ :CONTour[?]
│  └:LEGend [?]
├ :UNIT[?]
├ :WAVEform[?]
├ :SGRaph[?]
└ :THReshold[?]
```

This subnode has the following commands:

Table 79

Name	Description under
:CONTour	:PLUGin:EDIagram:SHOW:CONTour[?] on page 385
:LEGend	:PLUGin:EDIagram:SHOW:CONTour:LEGend[?] on page 385
:UNIT	:PLUGin:EDIagram:SHOW:UNIT[?] on page 385
:WAVEform	:PLUGin:EDIagram:SHOW:WAVeform[?] on page 386
:SGRaph	:PLUGin:EDIagram:SHOW:WAVeform:SGRaph[?] on page 386
:THReshold	:PLUGin:EDIagram:SHOW:WAVeform:THReshold[?] on page 387

:PLUGin:EDIagram:SHOW:CONTour[?]

Syntax	:PLUGin:EDIagram:SHOW:CONTour 'Identifier', <0|1|ON|OFF>
	:PLUGin:EDIagram:SHOW:CONTour? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	0|1
Description	This command shows/hides the contour lines on the graph.
	This query returns the current setting.
	This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:SHOW:CONT 'MyMeasurement', 1

:PLUGin:EDIagram:SHOW:CONTour:LEGend[?]

Syntax	:PLUGin:EDIagram:SHOW:CONTour:LEGend 'Identifier', <0|1|ON|OFF>
	:PLUGin:EDIagram:SHOW:CONTour:LEGend? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	0|1
Description	The command shows/hides contour legends.
	This query returns the current setting.
	This SCPI is for M8041A, M8051A and M8062A.
Example	:PLUG:EDI:SHOW:CONT:LEG 'MyMeasurement', 1

:PLUGin:EDIagram:SHOW:UNIT[?]

Syntax	:PLUGin:EDIagram:SHOW:UNIT 'Identifier', <UNIT | TIME >
	:PLUGin:EDIagram:SHOW:UNIT? 'Identifier',
Input Parameters	Identifier: 'MyMeasurement'
Return Range	UNIT | TIME
Description	The command sets the display unit. It can be either in terms of seconds or UNIT.

This query returns the current setting.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:SHOW:UNIT 'MyMeasurement', UNIT

:PLUGin:EDIagram:SHOW:WAVeform[?]

Syntax :PLUGin:EDIagram:SHOW:WAVeform 'Identifier', <0|1|OFF|ON>

:PLUGin:EDIagram:SHOW:WAVeform? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

<0|1|OFF|ON>: Show/hide waveform

Return Range 0|1

Description This command shows/hides the waveform. When this feature is enabled, the waveform is displayed together with the contour lines. Disabling this feature will only display contour lines, if available.

This query returns the current setting.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:SHOW:WAV 'MyMeasurement', 1

:PLUGin:EDIagram:SHOW:WAVeform:SGRaph[?]

Syntax :PLUGin:EDIagram:SHOW:WAVeform:SGRaph 'Identifier', <0|1|OFF|ON>

:PLUGin:EDIagram:SHOW:WAVeform:SGRaph? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

<0|1|OFF|ON>: Smooth waveform graphics.

Return Range 0|1

Description This command smooths out the waveform graphics. When this feature is enabled, the waveform is displayed together with the contour lines. Disabling this feature will only display contour lines, if available.

This query returns the current setting.

This SCPI is for M8041A, M8051A and M8062A.

Example :PLUG:EDI:SHOW:WAV:SGR 'MyMeasurement', 1

:PLUGin:EDIagram:SHOW:WAVeform:THReshold[?]

Syntax	:PLUGin:EDIagram:SHOW:WAVeform:THReshold 'Identifier', <NR3>
	:PLUGin:EDIagram:SHOW:WAVeform:THReshold? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	1E-6 to 1E-1
Description	This command sets the waveform BER threshold value.
	This query returns the current setting.
	This SCPI is for M8041A, M8051A and M8062A
Example	:PLUG:EDI:SHOW:WAV:THR 'MyMeasurement', 1E-3

:PLUGin:CCAPture Subnode

The M8020A/M8030A Analyzer captures the data received from the device under test. The captured data bits are displayed in the pattern capture pan in binary or 8b/10b symbol coding. The received data is compared with the expected data and the errored bits/symbols are highlighted. The captured data can be saved for post processing. The maximum capture memory is 2 Gb. However, it also depends on the holdoff length which represents the amount of symbols in which the trigger events will be ignored.

This subnode has the following SCPI structure:

```
:CCAPture
├─:CATalog?
├─:DELete
├─:NEW
├─:RESet
├─:STARt
├─:STOP
├─:SAVE
├─:TRIGger
│ └─:IMMediate
│   └─[:ONCE]
├─:SHOW
│ ├─:BIT
│ │ └─:PATTern
│ │   └─:VIEW
│ └─:SYMBol
│   └─:PATTern
│     └─:VIEW
├─:FETCh
│ └─...
├─:ACQuisition
│ └─...
└─:RUN
  └─...
```

This subnode has the following commands and subnodes:

Table 80

Name	Description under
:CATalog?	:PLUGin:CCAPture:CATalog? on page 390
DELete[?]	:PLUGin:CCAPture:DELete on page 391
NEW[?]	:PLUGin:CCAPture:NEW on page 391
RESet[?]	:PLUGin:JTOLerance:RESet on page 359
STARt[?]	:PLUGin:CCAPture:STARt on page 392
STOP[?]	:PLUGin:CCAPture:STOP on page 392
SAVE[?]	:PLUGin:CCAPture:SAVE on page 392
:TRIGger:IMMediate[:ONCE]	:PLUGin:CCAPture:TRIGger:IMMediate[:ONCE] on page 393
:SHOW:BIT:PATTern:VIEW	:PLUGin:CCAPture:SHOW:BIT[:PATTern][:VIEW][?] on page 393
:SHOW:SYMBol:PATTern:VIEW	:PLUGin:CCAPture:SHOW:SYMBol[:PATTern][:VIEW][?] on page 394
Subnodes	
FETCh	:PLUGin:CCAPture:FETCh Subnode on page 395
ACQuisition	:PLUGin:CCAPture:ACQuisition Subnode on page 400
RUN	:PLUGin:CCAPture:RUN Subnode on page 403

:PLUGin:CCAPture:CATalog?

Syntax :PLUGin:CCAPture:CATalog?

Description This command returns a list of all created compare and capture measurement names currently available for measuring.
This SCPI is only for M8041A and M8051A.

Example Assume the following is a list of created compare and capture measurement

names:

:PLUG:CCAP:NEW 'CCAP_1'

:PLUG:CCAP:NEW 'CCAP_2'

:PLUG:CCAP:NEW 'CCAP_3'

The command and returned list would look like the following:

:PLUG:CCAP:CAT?

"CCAP_1,CCAP_2,CCAP_3"

:PLUGin:CCAPture:DELete

Syntax :PLUGin:CCAPture:DELete ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to delete.

Description This command deletes a previously created compare and capture measurement addressed by the measurement name identifier.
This SCPI is only for M8041A and M8051A.

Example The following example deletes a compare and capture measurement addressed by the measurement name identifier called ‘MyMeasurement’:
:PLUG:CCAP:DEL ‘MyMeasurement’

:PLUGin:CCAPture:NEW

Syntax :PLUGin:CCAPture:NEW ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name.

Description This command creates a new compare and capture measurement name identifier.
This SCPI is only for M8041A and M8051A.

Example The following example creates a compare and capture measurement name identifier called ‘MyMeasurement’:
:PLUG:CCAP:NEW ‘MyMeasurement’

:PLUGin:CCAPture:RESet

Syntax :PLUGin:CCAPture:RESet ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to reset.

Description This command resets a compare and capture measurement addressed by the measurement name identifier.
This SCPI is only for M8041A and M8051A.

Example The following example resets a compare and capture measurement addressed with the measurement name identifier called ‘MyMeasurement’:
:PLUG:CCAP:RES ‘MyMeasurement’

:PLUGin:CCAPture:STARt

Syntax :PLUGin:CCAPture:STARt ‘identifier’

Input Parameters identifier’: Specify the measurement name to start.

Description This command starts a compare and capture measurement addressed by the measurement name identifier.
This SCPI is only for M8041A and M8051A.

Example The following example starts a compare and capture measurement addressed with the measurement name identifier called ‘MyMeasurement’:
:PLUG:CCAP:STAR ‘MyMeasurement’

:PLUGin:CCAPture:STOP

Syntax :PLUGin:CCAPture:STOP ‘identifier’

Input Parameters ‘identifier’: Specify the measurement name to stop.

Description This command stops the capturing of data and data is not uploaded into the measurement 'plug-in.'

This SCPI is only for M8041A and M8051A.

Example The following example stops a compare and capture measurement with the measurement name identifier called ‘MyMeasurement’:
:PLUG:CCAP:STOP ‘MyMeasurement’

:PLUGin:CCAPture:SAVE

Syntax :PLUGin:CCAPture:SAVE ‘identifier’,‘Location’,‘pattern-path’ [,overwrite-flag]

Input Parameters ‘identifier’: Specify the measurement name to save.

‘Location’: 'M*.DataIn1' or 'M*.DataIn2'

'<pattern-path>': Patterns can be saved as data and are stored in three different locations:

1 Local to current setting ("current:")

2 Shared between settings ("shared:")

3 Factory supplied standard patterns ("factory:"). These patterns are read

only and cannot be modified.

[,<overwrite-flag>]: ON | OFF | 0 | 1

Optional flag is a boolean parameter which specifies if an existing pattern file should be overwritten or not.

Description: The currently uploaded pattern is stored into the specified file on the hard disk.

This SCPI is only for M8041A and M8051A.

Example: The following example saves a compare and capture measurement with the measurement name identifier called ‘MyMeasurement’:

:PLUGin:CCAPture:SAVE "MyMeasurement", "M1.Datain1", "current:CapData"

:PLUGin:CCAPture:SAVE "MyMeasurement", "M1.Datain1", "current:CapData",1

:PLUGin:CCAPture:TRIGger:IMMediate[:ONCE]

Syntax: :PLUGin:CCAPture:TRIGger:IMMediate[:ONCE]

Input Parameters: ‘identifier’: Specify the measurement name.

'location' 'M*.DataIn1' or 'M*.DataIn2'

Description: This command simulates a stop event. Compare and capture is stopped and the recorded data is uploaded into the measurement.

This SCPI is only for M8041A and M8051A.

Example: PLUGin:CCAPture:TRIGger:IMMediate:ONCE ‘MyMeasurement’

:PLUGin:CCAPture:SHOW:BIT[:PATTern][:VIEW][?]

Syntax: :PLUGin:CCAPture:SHOW:BIT:PATTern:VIEW 'identifier', ‘Mode’

:PLUGin:CCAPture:SHOW:BIT:PATTern:VIEW? 'identifier'

Input Parameters: ‘identifier’: Specify the measurement name.

‘Mode’: BIN|HEX

Return Range: BIN|HEX

Description: This command selects the bit pattern view mode viz. BIN|HEX.

This SCPI is only for M8041A and M8051A.

Example :PLUGin:CCAPture:SHOW:BIT:PATTern:VIEW 'MyMeasurement', BIN

:PLUGin:CCAPture:SHOW:SYMBol[:PATTern][:VIEW][?]

Syntax :PLUGin:CCAPture:SHOW:SYMBol:PATTern:VIEW 'identifier', ‘Mode’

:PLUGin:CCAPture:SHOW:SYMBol:PATTern:VIEW? 'identifier'

Input Parameters ‘identifier’: Specify the measurement name.

‘Mode’: BIN|HEX|SYMBol

Return Range BIN|HEX|SYMBol

Description This command selects 8/10 bit pattern view mode viz. BIN|HEX|SYMBol.

This SCPI is only for M8041A and M8051A.

Example :PLUGin:CCAPture:SHOW:SYMBol:PATTern:VIEW 'MyMeasurement', BIN

:PLUGin:CCAPture:FETCh Subnode

This subnode has the following SCPI structure:

```
:FETCh
├ :ERRor
│  └:COUNt
├ :PATTern
│  ├:DATA
│  │ ├:HOLDoff?
│  │ └:DEPTh?
│  └:ERRor
│    └:SYMBol
│      ├:FIRSt?
│      └:LAST?
└ :RUN
   └[:STATus]?
```

This subnode has the following commands:

Table 81

Name	Description under
:ERRor:COUNt?	:PLUGin:CCAPture:FETCh:ERRor:COUNt? on page 396
:PATTern:DATA:HOLDoff?	:PLUGin:CCAPture:FETCh:PATTern[:DATA]:HOLDoff? on page 397
:PATTern:DATA:DEPTh?	:PLUGin:CCAPture:FETCh:PATTern[:DATA]:DEPTh? on page 397
:PATTern:ERRor:SYMBol:FIRSt?	:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:FIRSt? on page 398
:PATTern:ERRor:SYMBol:LAST?	:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:LAST? on page 398
:RUN[:STATus]?	:PLUGin:CCAPture:FETCh:RUN[:STATus]? on page 399

:PLUGin:CCAPture:FETCh:ERRor:COUNt?

Syntax :PLUGin:CCAPture:FETCh:ERRor:COUNt? ‘identifier’ [,‘Location’]

Input Parameters ‘identifier’: Specify the measurement name.

[,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’

Description This command gives the error count of the captured data. If simply run without specifying the location, then it returns the results for the current run.

This SCPI is only for M8041A and M8051A.

Example :PLUGin:CCAPture:FETCh:ERRor:COUNt? ‘MyMeasurement’,‘M1.DataIn1’

:PLUGin:CCAPture:FETCh:PATTern[:DATA]:HOLDoff?

Syntax :PLUGin:CCAPture:FETCh:PATTern[:DATA]:HOLDoff? 'identifier' [,‘Location’]

Input Parameters identifier: ‘Name of Measurement’

[,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’

Description This command returns the actual holdoff i.e. number of data bits captured before the trigger event. Returned value of this query is only valid when capture measurement is finished.

This SCPI is only for M8041A and M8051A.

Example :PLUGin:CCAPture:FETCh:PATTern:DATA:HOLDoff? 'MyMeasurement', ‘M1.DataIn1’

Return: (“M1.DataIn1”, ...)

:PLUGin:CCAPture:FETCh:PATTern[:DATA]:DEPTh?

Syntax :PLUGin:CCAPture:FETCh:PATTern[:DATA]:DEPTh? 'identifier' [,‘Location’]

Input Parameters identifier: ‘Name of Measurement’

[,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’

Description This command returns the total number of data bits captured. Returned value of this query is only valid when capture measurement is finished.

This SCPI is only for M8041A and M8051A.

Example :PLUGin:CCAPture:FETCh:PATTern:DATA:DEPTh? 'MyMeasurement', ‘M1.DataIn1’

Return: (“M1.DataIn1”, ...)

:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:FIRSt?

Syntax	:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:FIRSt? 'identifier' [,‘Location’]
Input Parameters	identifier: ‘Name of Measurement’ [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’
Description	This command fetches the first error symbol. This SCPI is only for M8041A and M8051A.
Example	:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:FIRSt? 'MyMeasurement',‘M1.DataIn1’ Return: (“M1.DataIn1”, ...)

:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:LAST?

Syntax	:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:LAST? 'identifier' [,‘Location’]
Input Parameters	identifier: ‘Name of Measurement’ [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’
Description	This command fetches the last error symbol. This SCPI is only for M8041A and M8051A.
Example	:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:LAST? 'MyMeasurement',‘M1.DataIn1’ Return: (“M1.DataIn1”, ...)

:PLUGin:CCAPture:FETCh:RUN[:STATus]?

Syntax	:PLUGin:CCAPture:FETCh:RUN[:STATus]? 'identifier' [,‘Location’]
Input Parameters	identifier: ‘Name of Measurement’ [,‘Location’]: ‘M*.DataIn1’ or ‘M*.DataIn2’
Description	This command returns the capture measurement status for each analyzer location. Returned value can be any of the following: STOP: Capture measurement is stopped or not currently running. No valid data is available in capture memory. STAR: Capture measurement is starting. No valid data is available in capture memory. ARM: Capture measurement is started (Armed) and waiting for trigger. No valid data is available in capture memory. TRIG: Capture logic is triggered and data is being captured. Valid data will be available soon in capture memory. UPL: Data capturing is done and data is being uploaded and analyzed. Valid data is available in captured memory. FIN: Capture measurement is finished. This SCPI is only for M8041A and M8051A.
Example	:PLUGin:CCAPture:FETCh:RUN:STATus? 'MyMeasurement',‘M1.DataIn1’ Return: (“M1.DataIn1”, ...)

:PLUGin:CCAPture:ACQuisition Subnode

This subnode has the following SCPI structure:

```
:ACQuisition
├─:ALOCation
├─:DEPTh
├─:HOLDoff
├─:SOURce
└─:SLOPe
```

This subnode has the following commands:

Table 82

Name	Description under
:ALOCation[?]	:PLUGin:CCAPture:ACQuisition:ALOCation[?] on page 401
:DEPTh[?]	:PLUGin:CCAPture:ACQuisition:DEPTh[?] on page 401
:HOLDoff[?]	:PLUGin:CCAPture:ACQuisition:HOLDoff[?] on page 401
:SOURce[?]	:PLUGin:CCAPture:ACQuisition:SOURce[?] on page 402
:SLOPe[?]	:PLUGin:CCAPture:ACQuisition:SLOPe[?] on page 402

:PLUGin:CCAPture:ACQuisition:ALOCation[?]

Syntax	:PLUGin:CCAPture:ACQuisition:LOCation ‘identifier’,'Location'
	:PLUGin:CCAPture:ACQuisition:LOCation? ‘identifier’
Input Parameters	identifier: ‘Name of Measurement’
	‘Location’: ‘M*.DataIn1’ or ‘M*.DataIn2’
Description	Specified the location the compare & capture measurement cooperates with. Examples for a location are 'M1.DataIn1', 'M3.DataIn2',... or a name of an existing group of DataIns.
	This SCPI is only for M8041A and M8051A.
Example	PLUGin:CCAPture:ACQuisition:ALOCation ‘My Measurement’, ‘M1.DataIn1’

:PLUGin:CCAPture:ACQuisition:DEPTh[?]

Syntax	:PLUGin:CCAPture:ACQuisition:DEPTh ‘identifier’,<NRf>
	:PLUGin:CCAPture:ACQuisition:DEPTh? ‘identifier’
Input Parameters	identifier: ‘Name of Measurement’
Range	0 to 2147483648
Description	This command configures how long the capture logic will write the capture memory after detecting the trigger event.
	This SCPI is only for M8041A and M8051A.
Example	:PLUGin:CCAPture:ACQuisition:DEPTh ‘My Measurement’, 1

:PLUGin:CCAPture:ACQuisition:HOLDoff[?]

Syntax	:PLUGin:CCAPture:ACQuisition:HOLDoff ‘identifier’,<NRf>
	:PLUGin:CCAPture:ACQuisition:HOLDoff? ‘identifier’
Input Parameters	identifier: ‘Name of Measurement’
Range	0 to 2147483648
Description	Specifies the amount of bits the trigger event is ignored after starting capturing of data.

This SCPI is only for M8041A and M8051A.

Example :PLUGin:CCAPture:ACQuisition:HOLDoff ‘My Measurement’, 1

:PLUGin:CCAPture:ACQuisition:SOURce[?]

Syntax	:PLUGin:CCAPture:ACQuisition:SOURce ‘identifier’,<CINA | CINB | ERRor I IMM>
	:PLUGin:CCAPture:ACQuisition:SOURce? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name.
	<CINA | CINB | ERRor | IMM>
Return Range	CINA | CINB | ERRor | IMM
Description	Specifies the event for triggering the compare & capture measurement. CTRL IN A or CRTL IN B or an ERRor occurred can be chosen to trigger the capture recording. IMM can be used to capture data without any trigger.
	This SCPI is only for M8041A and M8051A.
Example	:PLUGin:CCAPture:ACQuisition:SOURce ‘My Measurement’,CINA

:PLUGin:CCAPture:ACQuisition:SLOPe[?]

Syntax	:PLUGin:CCAPture:ACQuisition:SLOPe ‘identifier’, <POS | NEG>
	:PLUGin:CCAPture:ACQuisition:SLOPe? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name.
	<POS | NEG>
Return Range	POS | NEG
Description	If CINA or CINB is selected for triggering the capture process this command specifies the slope the trigger react with.
	This SCPI is only for M8041A and M8051A.
Example	:PLUGin:CCAPture:ACQuisition:SLOPe ‘My Measurement’, POS

:PLUGin:CCAPture:RUN Subnode

This subnode has the following SCPI structure:

```
:RUN
├:MESSage?
├:PROGress?
├:STATus?
└:LOG?
```

This subnode has the following commands:

Table 83

Name	Description under
:MESSage?	:PLUGin:CCAPture:RUN:MESSage? on page 404
:PROGress?	:PLUGin:CCAPture:RUN:PROGress? on page 404
:STATus?	:PLUGin:CCAPture:RUN:STATus? on page 405
LOG?	:PLUGin:CCAPture:RUN:LOG? on page 405

:PLUGin:CCAPture:RUN:MESSage?

Syntax	:PLUGin:CCAPture:RUN:MESSage? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
Description	This command returns a string describing the state of the compare & capture measurement. Possible states include NotStarted, Running, Stopped and Triggered. This SCPI is only for M8041A and M8051A.
Example	The following example returns the state of a compare and capture measurement with the measurement name identifier called 'MyMeasurement': :PLUG:CCAP:RUN:MESS? 'MyMeasurement' Running

:PLUGin:CCAPture:RUN:PROGress?

Syntax	:PLUGin:CCAPture:RUN:PROGress? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
Return Range	0.0 to 1.0
Description	This command returns a number in the range of 0.0 to 1.0 to indicate the progress of a compare and capture measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished. This SCPI is only for M8041A and M8051A.
Example	The following example returns the progress of a compare and capture measurement with the measurement name identifier called 'MyMeasurement': :PLUG:CCAP:RUN:PROG? 'MyMeasurement' 0.51

:PLUGin:CCAPture:RUN:STATus?

Syntax	:PLUGin:CCAPture:RUN:STATus? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name.
Return Range	0|1
Description	This command returns the running status of the compare & capture measurement. A 0 indicates the measurement is not running and a 1 indicates the measurement is running. This SCPI is only for M8041A and M8051A.
Example	The following example returns the running status of a compare & capture measurement with the measurement name identifier called ‘MyMeasurement’: :PLUG:CCAP:RUN? ‘MyMeasurement’

:PLUGin:CCAPture:RUN:LOG?

Syntax	:PLUGin:CCAPture:RUN:LOG? ‘identifier’
Input Parameters	‘identifier’: Specify the measurement name.
Description	This command return the logs for the addressed measurement.
Example	The following example return the logs for the measurement name identifier called ‘MyMeasurement’: :PLUGin:CCAPture:RUN:LOG? ‘MyMeasurement’ Output: #292 07/20/2015 18:23:41,Measurement.Pattern Capture.MyMeasurement,Info,"Start Measurement"

SYSTem Subsystem

The SYSTem subsystem is used for general system functions.

This subsystem has the following SCPI structure:

```
:SYSTem
├─:INSTrument
│ ├─:RESet
│ ├─:GROup
│ │ ├─:DEFine
│ │ ├─:DELete
│ │ ├─:MEMBer?
│ │ └─[:LIST]?
│ └─:LTRaining
│   ├─[:LOG]?
│   ├─[:LOG]
│   │ ├─:CLEar
│   │ └─:BLOCk
│   └─:EQUalization
│     └─[:STATus]?
├─:INFormation
│ ├─[:INSTrument]
│ │ ├─:IDENtifier?
│ │ ├─:LOCations?
│ │ └─:GROups
│ │   └─:USER?
│ └─:LICenses
│   ├─:HOST[:LIST]?
│   └─:MODules[:LIST]?
├─:COMMunicate
│ └─:TCPip
│   └─:CONTrol?
├─:DCINterface
│ ├─[:PROGram]
│ │ ├─:LOAD
│ │ ├─:UNLoad
│ │ └─:SELected?
│ ├─:EXECute[:INIT]
│ ├─:DEVice
│ │ ├─:CONNect[?]
│ │ └─:MODE[?]
│ └─:LANE
│   └─:MODE[?]
└─:RESet
  ├─[:ALL]
  └─:INSTrument
```

This subsystem has the following commands:

Table 84

Name	Description under
:INSTrument:RESet	:SYSTem:INSTrument:RESet on page 409
:INSTrument:GROup:DEFine	:SYSTem:INSTrument:GROup:DEFine on page 409
:INSTrument:GROup:DELete	:SYSTem:INSTrument:GROup:DELete on page 409
:INSTrument:GROup:MEMBer?	:SYSTem:INSTrument:GROup:MEMBer? on page 410
:INSTrument:GROup[:LIST]?	:SYSTem:INSTrument:GROup[:LIST]? on page 410
:INSTrument:LTRaining[:LOG]?	:SYSTem:INSTrument:LTRaining[:LOG]? on page 411
:INSTrument:LTRaining[:LOG]:CLEar	:SYSTem:INSTrument:LTRaining[:LOG]:CLEar on page 412
:INSTrument:LTRaining[:LOG]:BLOCk?	:SYSTem:INSTrument:LTRaining[:LOG]:BLOCk? on page 412
:INSTrument:LTRaining:EQUalization[:STATus]?	:SYSTem:INSTrument:LTRaining:EQUalization[:STATus]? on page 413
:INFormation[:INSTrument]:IDENtifier?	:SYSTem:INFormation[:INSTrument]:IDENtifier? on page 414
:INFormation[:INSTrument]:LOCations?	:SYSTem:INFormation[:INSTrument]:LOCations? on page 414
:INFormation[:INSTrument]:GROups:USER?	:SYSTem:INFormation[:INSTrument]:GROups:USER? on page 414
:INFormation:LICenses:HOST[:LIST]?	SYSTem:INFormation:LICenses:HOST[:LIST]? on page 415
:INFormation:LICenses:MODules[:LIST]?	:SYSTem:INFormation:LICenses:MODules[:LIST]? on page 416
:COMMunicate:TCPip:CONTrol?	:SYSTem:COMMunicate:TCPip:CONTrol? on page 416
:DCINterface[:PROGram]:LOAD	:SYSTem:DCINterface[:PROGram]:LOAD on page 417

Name	Description under
:DCINterface[:PROGram]:UNLoad	:SYSTem:DCINterface[:PROGram]:UNLoad on page 417
:DCINterface[:PROGram]:SELected?	:SYSTem:DCINterface[:PROGram]:SELected? on page 418
:DCINterface:EXEcute[:INIT]	:SYSTem:DCINterface:EXEcute[:INIT] on page 418
:DCINterface:DEVice:CONNect	:SYSTem:DCINterface:DEVice:CONNect[?] on page 418
:DCINterface:DEVice:MODE	:SYSTem:DCINterface:DEVice:MODE[?] on page 419
:DCINterface:LANE:MODE	:SYSTem:DCINterface:LANE:MODE[?] on page 420
:RESet[:ALL]	:SYSTem:RESet[:ALL] on page 421
:RESet:INSTrument	:SYSTem:RESet:INSTrument on page 421

:SYSTem:INSTrument:RESet

Syntax :SYSTem:INSTrument:RESet

Description This command resets the instrument state to its default values. This command does not influence the SCPI status system as *RST does.

Example :SYST:INST:RES

:SYSTem:INSTrument:GROup:DEFine

Syntax :SYSTem:INSTrument:GROup:DEFine
’GroupName’,‘identifier’,‘identifier’,‘identifier’,‘...’

Input Parameters ‘GroupName’: Assign a name to this group of location identifiers.

‘identifier’: Define the group of location identifiers.

Description This command defines a group of location identifiers under one group name. All properties belonging to these location identifiers can be addressed simultaneously using the group name.

NOTE **If you define a group name that already exists, the existing name with its group of defined location identifiers will be redefined without notification.**

Example :SYST:INST:GRO:DEF ‘Outputs’,‘M1.DataOut1’,‘M1.DataOut2’
VOLT:AMPL ‘Outputs’,0.05

:SYSTem:INSTrument:GROup:DELete

Syntax :SYSTem:INSTrument:GROup:DELete ‘GroupName’

Input Parameters ‘GroupName’: Specify group name to delete.

Description This command deletes the specified group.

Example :SYST:INST:GRO:DEL ‘Outputs’

:SYSTem:INSTrument:GROup:MEMBer?

Syntax	:SYSTem:INSTrument:GROup:MEMBer? ‘GroupName’
Input Parameters	‘GroupName’: Specify group name.
Description	This command returns the comma separated list of member names contained within the specified group.
Example	:SYST:INST:GRO:MEMB? ‘Outputs’ “M1.DataOut1”,“M1.DataOut2”

:SYSTem:INSTrument:GROup[:LIST]?

Syntax	:SYSTem:INSTrument:GROup[:LIST]?
Description	This command returns a comma separated list of all group names.
Example	:SYST:INST:GRO? “Outputs”

:SYSTem:INSTrument:LTRaining[:LOG]?

Syntax :SYSTem:INSTrument:LTRaining[:LOG]?

Description This command returns accumulated Link Training logs for the selected DataOut locations in text format.

Example :SYST:INST:LTR:LOG?

Link Training Logging for M1.DataOut1 at 5/4/2015 12:42:36 PM

State	Execution Time
Detect.Active	1.04864 ms
Polling.Active	2.064992 ms
Polling.Configuration	152.112 us
Configuration.Linkwidth.Start	2.896 us
Configuration.Linkwidth.Accept	304 ns
Configuration.Lanenum.Wait	2.8 us
Configuration.Lanenum.Accept	544 ns
Configuration.Complete	3.696 us
Configuration.Idle	256 ns
L0	224 ns
Recovery.RcvrLock	3.52 us
Recovery.RcvrCfg	2.464 us
Recovery.Speed	8.464 us
Recovery.RcvrLock	448 ns
Recovery.Equalization.Phase1	2.20608 ms
Recovery.Equalization.Phase2	21.033456 ms
Recovery.Equalization.Phase3	1.904 us
Recovery.RcvrLock	2.016 us
Recovery.RcvrCfg	656 ns
Recovery.Idle	112 ns
Loopback.Entry	2.816 us
Loopback.Active	-

BERT Tx Equalization

Accept	PresetNumber	PreCursor	MainCursor	PostCursor	FullSwing	LowFrequency
True	P7	-	-	-	24	8
True	P4	-	-	-	24	8

DUT Tx Equalization

Event	Accept	PresetNumber	PreCursor	MainCursor	PostCursor	FullSwing	LowFrequency
Initial	-	P7	-	-	-	40	12
Target	-	P4	-	-	-	40	12
Reported	True	P4	0	40	0	40	12

:SYSTem:INSTrument:LTRaining[:LOG]:CLEar

Syntax :SYSTem:INSTrument:LTRaining[:LOG]:CLEar

Description This command clears the Link Training logs.

Example :SYST:INST:LTR:LOG:CLE

:SYSTem:INSTrument:LTRaining[:LOG]:BLOCk?

Syntax :SYSTem:INSTrument:LTRaining[:LOG]:BLOCk?

Description The content of the log file is returned as definite block. (see :SYSTem:INSTrument:LTRaining[:LOG]?) Control character (e.g. Line Feed) are contained in the log file. This can interrupt the transfer to the host computer. A definite block is used to transfer data streams that can contain special characters. see SCPI language definition or IEEE 488.2 -BLOCK PROGRAM DATA.

Example :SYSTem:INSTrument:LTRaining:LOG:BLOCk?

:SYSTem:INSTrument:LTRaining:EQUalization[:STATus]?

Syntax :SYSTem:INSTrument:LTRaining:EQUalization[:STATus]? ‘identifier’

Input Parameters ‘identifier:’ 'M*.DataOut1' or 'M*.DataOut2'

Description This query returns the Dut transmitter equalization information from the link up block. The results are combined to an expression. An expression is comparable to a struct in a programming language. It’s a set of parameter of different types separated by a comma and enclosed by parenthesis.

The returned expression contains the location and a set of values.

(“Location”, Accepted, PreCursor, MainCursor, PostCursor, FullSwing, LowFrequency)

Location: It’s a string referred to the place the Dut tx equalization information is located. In this case its either M*.DataOut1 or M*.DataOut2. This parameter is important if a group name is used as ‘identifier’.

Accepted: Requested coefficients are accepted or rejected, either 0 or 1.

PreCursor: Received pre-cursor coefficient value during Phase 2 or 3 of the equalization. If no data is present, 9.91 E 37 is returned.

MainCursor: Received data-cursor coefficient value during Phase 2 or 3 of the equalization. If no data is present, 9.91 E 37 is returned.

PostCursor: Received post-cursor coefficient value during Phase 2 or 3 of the equalization. If no data is present, 9.91 E 37 is returned.

FullSwing: Full swing value received during Phase 1 of the equalization. If no data is present, 9.91 E 37 is returned.

LowFrequency: Low Frequency value received during Phase 1 of the equalization. If no data is present, 9.91 E 37 is returned.

If this query is sent to a ‘group’ of inputs the response will be expanded by additional sets of counted values. The expression contains for every input belonging to a group a separate set of values starting with the location.

Example :SYST:INST:LTR:EQU:STAT? 'M1.DataOut1'

:SYSTem:INFormation[:INSTrument]:IDENtifier?

Syntax :SYSTem:INFormation[:INSTrument]:IDENtifier?

Description This command returns a list of all identifiers defined in the system.

Example :SYST:INF:IDEN?
(M1.System,M1.SysInA,M1.DataIn1,M1.SysInB,M1.DataIn2,M1.CtrlInA,M1.CtrlInB,M1.DataOut2,M1.Simulation1,M1.DataOut1,M1.ClkOut,M1.Simulation2,M1.ClkGen,M1.TrigOut,M1.SysOutA,M1.SysOutB,M1.CtrlOutA,M1.RefClkOut,M2.DataOut2,M2.CtrlOutA,M2.DataOut1,M2.DataIn1,M2.CtrlInA,M2.DataIn2,M2.CtrlInB,M2.Simulation1,M2.Simulation2,M3.ClkOut,M3.DataIn2,M3.DataOut,M3.ElectIdleIn,M3.DataIn1,M3.MuxMode,M3.DataIn)

:SYSTem:INFormation[:INSTrument]:LOCations?

Syntax :SYSTem:INFormation[:INSTrument]:LOCations?

Description This command returns a list of all locations defined in the system.

Example :SYST:INF:LOC?
(ClkGen,System,RefClkOut,TrigOut,SysOutA,SysOutB,CtrlOutA,SysInA,SysInB,CtrlInA,CtrlInB,DataIn1,DataIn2,DataOut1,DataOut2,Simulation1,Simulation2,MuxMode,ElectIdleIn)

:SYSTem:INFormation[:INSTrument]:GROups:USER?

Syntax :SYSTem:INFormation[:INSTrument]:GROups:USER?

Description This command returns a list of all user defined parameter group names currently in use by the system.

Example :SYST:INF:GRO:USER?
(Outputs,Inputs)

SYSTem:INFormation:LICenses:HOST[:LIST]?

Syntax SYSTem:INFormation:LICenses:HOST[:LIST]?

Description This command lists the description of all the host licenses supported by the device.

Example SYSTem:INFormation:LICenses:HOST[:LIST]?

((PCSERNO,AB99889386,("M8070A-CAL","CAL","1.0","System Software for M8000 Series, internal-use only, transportable perpetual license","Installed","Perpetual","12/31/9999 11:59:59 PM","clockgroup-wide")("M8070A-0TP","0TP","1.0","System Software for M8000 Series of BER Test Solutions, transportable, perpetual license","Not installed","Perpetual","12/31/9999 11:59:59 PM","clockgroup-wide")("M8070A-0NP","0NP","1.0","System Software for M8000 Series of BER Test Solutions, network/floating, perpetual license","Not installed","Perpetual","12/31/9999 11:59:59 PM","clockgroup-wide")("M8085A-CT1","CT1","1.0","MIPI C-PHY Editor for M819xA AWG, Transportable, Perpetual License","Not installed","Perpetual","12/31/9999 11:59:59 PM","clockgroup-wide")("M8085A-CN1","CT1","1.0","MIPI C-PHY Editor for M819xA AWG, Network/Floating, Perpetual License","Not installed","Perpetual","12/31/9999 11:59:59 PM","clockgroup-wide")("M8085A-CD1","CT1","1.0","MIPI C-PHY Editor for M819xA AWG, Temporary License","Not installed","Perpetual","12/31/9999 11:59:59 PM","clockgroup-wide")("M8085A-DT1","DT1","1.0","MIPI D-PHY Editor for M819xA AWG, Transportable, Perpetual License","Not installed","Perpetual","12/31/9999 11:59:59 PM","clockgroup-wide")("M8085A-DN1","DT1","1.0","MIPI D-PHY Editor for M819xA AWG, Network/Floating, Perpetual License","Not installed","Perpetual","12/31/9999 11:59:59 PM","clockgroup-wide")("M8085A-DD1","DT1","1.0","MIPI D-PHY Editor for M819xA AWG, Temporary License","Not installed","Perpetual","12/31/9999 11:59:59 PM","clockgroup-wide")))

:SYSTem:INFormation:LICenses:MODules[:LIST]?

Syntax :SYSTem:INFormation:LICenses:MODules[:LIST]?

Description This command lists the description of all the module licenses supported by the device.

Example :SYSTem:INFormation:LICenses:MODules:LIST?

:SYSTem:COMMunicate:TCPip:CONTrol?

Syntax :SYSTem:COMMunicate:TCPip:CONTrol?

Description This command implements SCPI over TCP with Control Connection.

The Keysight Automation Advantage (AAA) standard requires that instruments that support SCPI also support SCPI traffic over TCP with a control connection. This control channel is set up by sending SYSTem:COMMunicate:TCPip:CONTrol? to the instrument. To use the control port:

- Open a socket connection to port 5025, the standard SCPI- over-sockets port.
- Over that socket connection, issue the SYST:COMM:TCP:CONT? query to retrieve the control port.
- Open a second socket connection to the control port.

Now you can send control commands: for example, send “DCL\n” on the control port to issue a device clear.

Example :SYST:COMM:TCP:CONT?

:SYSTem:DCINterface[:PROGram]:LOAD

Syntax :SYSTem:DCINterface[:PROGram]:LOAD “PathToPythonProgramFile”

Description This commands loads the given Python script into the DUT control interface. This DUT control interface program can be used to read the bit error rate of an externally connected DUT and provide these to the M8070A software as if the counters are part of the instrument. These values can then be used by the built in Bit Error Ratio measurement or Jitter Tolerance measurement.

In addition the script can implement functions that extend the measurements in order to adapt the measurement to the specific test setup and device under test.

Python scripts can be part of settings and shared areas. The addressing scheme looks like:

"Factory/HookExample.py" // Read only factory shared area

"Shared/HookExample.py" // User shared area

"Current/HookExample.py" // Current setting

Example :SYSTem:DCINterface:PROGram:LOAD "factory/DutCounterIntegration.py"

:SYSTem:DCINterface[:PROGram]:UNLoad

Syntax :SYSTem:DCINterface[:PROGram]:UNLoad

Description This command unloads the currently used DUT Control Interface script and all the resources that the script integrated into the system.

Example :SYSTem:DCINterface:PROGram:UNLoad

:SYSTem:DCINterface[:PROGram]:SELected?

Syntax :SYSTem:DCINterface[:PROGram]:SELected?

Description This query returns the DUT Control Interface script that is currently in use. If the DUT Control Interface is not using a script, an empty string is returned.

Example :SYSTem:DCINterface:PROGram:SELected?

:SYSTem:DCINterface:EXEcute[:INIT]

Syntax :SYSTem:DCINterface:EXEcute[:INIT] ‘Location’[,“ArgumentsForInit”]

Input Parameters ‘Location’: ‘*.Lane*’

Description This command matches the DUT Control Interface hook function DUT_Init(location, argument).

Sending this command results in executing the function DUT_Init with the given parameters if it is implemented by the script that is currently in use (see syst:dcin:load).

The Location argument shall be one of the locations defined by the scripts DUT_getLocations, but as full system location identifier.

For example, if,DUT_getLocations defines the locations "Lane1" and "Lane2", then valid Location arguments are "DCI.Lane1" and "DCI.Lane2"

The argument string is not checked anyway, but simply passed to the script implementation of DUT_Init.

Example :SYSTem:DCINterface:execute:init "DCI.Lane1"

:SYSTem:DCINterface:DEVice:CONNect[?]

Syntax :SYSTem:DCINterface:DEVice:CONNect ‘identifier’, <ON|OFF|1|0>

:SYSTem:DCINterface:DEVice:CONNect ‘identifier’?

Input Parameters	‘identifier’: “DCI.Control”
Description	This command relates to the DUT Control Interface hook functions DUT_connect and DUT_disconnect. When these functions are implemented by the currently used script (see syst:dcin:load), then this command is controlling the Connection parameter under the function Device for the Control location of the device under test. The location argument must be the Control location as full system location identifier as displayed in the parameter editor (e.g. "DCI.Control"). This query returns the current setting.
Example	:SYSTem:DCINterface:DEVice:CONNect ‘DCI.Control’, 1

:SYSTem:DCINterface:DEVice:MODE[?]

Syntax	:SYSTem:DCINterface:DEVice:MODE ‘identifier’, "mode" :SYSTem:DCINterface:DEVice:MODE? ‘identifier’
Input Parameters	‘identifier’: “DCI.Control” ‘mode’: Individual modes or configurations.
Description	This command relates to the DUT Control Interface hook functions DUT_getDeviceModes and DUT_setDeviceMode. When these functions are implemented by the currently used script (see syst:dcin:load), then this command is controlling the Mode parameter under the function Device for the Control location of the device under test. The location argument must be the Control location as full system location identifier as displayed in the parameter editor (e.g. "DCI.Control"). The mode argument must be one of the strings returned by DUT_getDeviceModes. This can be used to control configurations that are valid for the whole device under test. When sending this command in one SCPI transaction with :syst:dcin:lane:mode, then the device mode hook (DUT_setDeviceMode) will be executed before the lane specific mode (DUT_setLaneMode). This query returns the current setting.
Example	:SYSTem:DCINterface:DEVice:MODE ‘DCI.Control’,‘Configuration1’

:SYSTem:DCINterface:LANE:MODE[?]

Syntax

:SYSTem:DCINterface:LANE:MODE "Location", "mode"

:SYSTem:DCINterface:LANE:MODE? "Location"

Input Parameters

‘Location’: ‘*.Lane*’

‘mode’: Individual modes or configurations.

Description

This command relates to the DUT Control Interface hook functions DUT_getLaneModes and DUT_setLaneMode.

When these functions are implemented by the currently used script (see syst:dcin:load), then this command is controlling the Mode parameter under the function Lane for the given lane of the device under test.

The location argument must be the Control location as full system location identifier as displayed in the parameter editor (e.g. "DCI.Lane1").

The mode argument must be one of the strings returned by DUT_getLaneModes.

This can be used to control configurations that are valid for an individual lane or channel of the device under test.

When sending this command in one SCPI transaction with :syst:dcin:dev:mode, then the device mode hook (DUT_setDeviceMode) will be executed before the lane specific mode (DUT_setLaneMode).

This query returns the current setting.

Example

:SYSTem:DCINterface:LANE:MODE ‘DCI.Lane1’,‘MODE1’

:SYSTem:RESet[:ALL]

Syntax	:SYSTem:RESet[:ALL] [HARD | SOFT]
Input Parameters	[HARD | SOFT]
Description	This command resets the instrument. If SOFT is passed the instrument will load its default values and defined 'Groups' in the GUI will be untouched. If HARD is passed the instrument will load the factory setting and all 'Group' definitions will be cleared. Independent of passing HARD or SOFT all plug-ins (measurements) are closed.
Example	:SYSTem:RESet:ALL SOFT

:SYSTem:RESet:INSTrument

Syntax	:SYSTem:RESet:INSTrument [HARD | SOFT]
Input Parameters	[HARD | SOFT]
Description	This command resets the instrument. If SOFT is passed the instrument will load its default values and defined 'Groups' in the GUI will be untouched. If HARD is passed the instrument will load the factory setting and all 'Group' definitions will be cleared.
Example	:SYSTem:RESet:INSTrument SOFT

DATA Subsystem

The DATA subsystem is used to select patterns, define symbol parameters, define sequence blocks and loops and synchronize the pattern.

This subsystem has the following SCPI structure:

```
:DATA
├:PATTern
│└...
│
├:SEQuence
│└...
│
└:SYNC
 └...
```

The subsystem has the following subnodes:

Table 85

Name	Description under
:PATTern	:DATA:PATTern Subnode on page 423
:SEQuence	:DATA:SEQuence Subnode on page 431
:SYNC	:DATA:SYNC Subnode on page 436

:DATA:PATTern Subnode

For additional information about patterns, refer to Working with Patterns on page 42.

This subnode has the following SCPI structure:

```
:DATA
└:PATTern
  ├:USE[?]
  ├:IDATa[?]
  ├:BLENgth?
  ├:CATalog?
  ├:CINFormation?
  ├:DESCription[?]
  ├:DELete
  ├:PACKed[?]
  ├:IMPort
  └:EXPort
```

This subnode has the following commands:

Table 86

Name	Description under
:USE[?]	:DATA:PATTern:USE[?] on page 424
:IDATa[?]	:DATA:PATTern:IDATa[?] on page 425
:BLENgth?	:DATA:PATTern:BLENgth? on page 425
:CATalog?	:DATA:PATTern:CATalog? on page 426
:CINFormation?	:DATA:PATTern:CINFormation? on page 426
:DESCription[?]	:DATA:PATTern:DESCription[?] on page 427
:DELete	:DATA:PATTern:DELete on page 427
:PACKed[?]	:DATA:PATTern:PACKed[?] on page 428
:IMPort	:DATA:PATTern:IMPort on page 429
:EXPort	:DATA:PATTern:EXPort on page 430

:DATA:PATTern:USE[?]

Syntax

:DATA:PATTern:USE <pattern-name>,<num-symbols>[,<symbol-coding>

[,<mask-used>[,<squelch-used>,<error-used>]]]

:DATA:PATTern:USE? <pattern-name>

Input Parameters

<pattern-name>: Name of the pattern. Name consists of root node (“factory:”, “shared:”, or “current:”) plus folder names separated by ‘/’ plus pattern name.

<num-symbols>: Length of the pattern in number of symbols.

<symbol- coding>: BIT|B8B10|B128B130|B128B132 for bit coding, 8b10b coding, 128/130 coding or 128/132 coding. Default is BIT.

<mask- used>: 0|1|OFF|ON to specify if mask attribute is used. Default is OFF. Not using mask is mostly useful for bit coding to save space.

<squelch-used>: 0|1|OFF|ON to specify if squelch attribute is used. Default is OFF. Not using squelch is mostly useful for bit coding to save space.

<error-used>: 0|1|OFF|ON to specify if error attribute is used. Default is OFF. Error Attribute is used for captured patterns. When <error-used> is ON, <mask-used> and <squelch-used> must be OFF.

Return Range

<num-symbols>, <symbol-coding>, <mask-used>, <squelch-used>,<error-used>

Description

This command creates a new pattern. If a pattern exists with the same name, it is overwritten.

Example :DATA:PATT:USE "current:demo",20

Example :DATA:PATT:USE "current:demo8b10b",5,b8b10

Example :DATA:PATT:USE "current:demo128b130b",5,b128b130

Example :DATA:PATT:USE "current:demo128b132b",5,b128b132

Example :DATA:PATT:USE? "current:demo"

5,BIT,0,0,0

Example :DATA:PATT:USE? "factory:SATA/LTDP_short_b8b10"

512,B8B10,1,1,0

:DATA:PATTern:IDATa[?]

Syntax

:DATA:PATTern:IDATa <pattern-name>,<bit-offset>,<num-bits>,<data-block>

:DATA:PATTern:IDATa? <pattern-name>,<bit-offset>,<num-bits>

Input Parameters

<pattern-name>: Name of the pattern. Name consists of root node ("shared:" or "current:") plus folder names separated by '/' plus pattern name. The "factory:" root node is not allowed here, as these patterns are read only and cannot be modified.

<bit-offset>: Offset into the pattern bit sequence. Bit offset is used, to allow modifications within symbols, for example just modifying a single attribute bit.

<num-bits>: Number of bits to modify. Specified in bits to allow modifications within symbols, e.g. just modifying a single attribute bit.

<data-block>: Bit sequence as definite length block. Number of bits encoded in one byte is specified by the :DATA:PATT:PACK command.

Return Range

<pattern-name>, <bit-offset>, <num-bits>

Description

This command is used to modify a pattern.

Example

The following example enables the scrambler in the first symbol of an 8b10b coded pattern.

:DATA:PATT:IDAT 'current:demo8b10b',12,1,

#111

:DATA:PATTern:BLENgth?

Syntax

:DATA:PATTern:BLENgth? <pattern-name>

Input Parameters

<pattern-name>: Name of the pattern. Name consists of root node ("factory:", "shared:", or "current:") plus folder names separated by '/' plus pattern name.

Return Range

<num-bits>

Description

This command returns the pattern length in bits of a specified pattern.

Example

:DATA:PATT:BLEN? "factory:SATA/LTDP_short_b8b10"

8192

:DATA:PATTern:CATalog?

Syntax :DATA:PATTern:CATalog? <pattern-path>

Input Parameters <pattern-path>: Location of patterns.

Description This command returns a list of patterns found at the specified path.

Example :DATA:PATT:CAT? "factory:XAUI"
"CJPAT_bit,CRPAT_b8b10,CRPAT_bit"

:DATA:PATTern:CINFormation?

Syntax :DATA:PATTern:CINFormation?
<symbol-coding>[,<mask-used>[,<squelch-used>]]

Input Parameters <symbol- coding>: BIT|B8B10|B128B130|B128B132 for bit coding, 8b10b coding, 128/130 coding or 128/132 coding.

<mask- used>: 0|1|OFF|ON to specify if mask attribute is used. Default is OFF. Not using mask is mostly useful for bit coding to save space.

<squelch-used>: 0|1|OFF|ON to specify if squelch attribute is used. Default is OFF. Not using squelch is mostly useful for bit coding to save space.

Return Range <symbol-coding>, <mask-used>, <squelch-used>

Description This command returns coding information.

Example :DATA:PATT:CINF? b8b10,on,on
16,"Data",0,7,"K/D",8,8,"Mask",10,10,"Squelch",11,11,"Enable Scrambler",12,12,"Pause Scrambler",13,13,"Reset Scrambler",14,14,"Start of Frame",15,15

:DATA:PATTern:DESCription[?]

Syntax
:DATA:PATTern:DESCription <pattern-name>,<description>

:DATA:PATTern:DESCription? <pattern-name>

Input Parameters
<pattern-name>: Name of the pattern. Name consists of root node (“factory:”, “shared:”, or “current:”) plus folder names separated by ‘/’ plus pattern name; like “shared:SATA/LTDP_short_b8b10”.

<description>: Enter a pattern description.

Return Range
<description>

Description
This command is used to add a description to the specified pattern. This SCPI doesn’t allow adding description to factory patterns.

Example
Create a new pattern by the name “bit_pattern” and save it to “shared” memory location. Now, use the following example to set “sample bit pattern” as the description.

:DATA:PATT:DESC “shared:bit_pattern”,“sample bit pattern”

:DATA:PATTern:DELete

Syntax
:DATA:PATTern:DELete <pattern-name>

Input Parameters
<pattern-name>: Name of the pattern. Name consists of root node (“factory:”, “shared:”, or “current:”) plus folder names separated by ‘/’ plus pattern name.

Description
This command deletes the specified pattern. A factory pattern can’t be deleted.

Example
:DATA:PATT:DEL "shared:SATA/LTDP_short_b8b10"

:DATA:PATTern:PACKed[?]

Syntax	:DATA:PATTern:PACKed <packing>
	:DATA:PATTern:PACKed?
Input Parameters	<packing>: Specify number of bits (1|4|8) that are encoded in a byte when using the :DATA:PATT:IDAT command or query.
Return Range	1|4|8
Description	This command specifies how many bits are encoded in a byte when using the :DATA:PATT:IDAT command or query. The default is “8” for most efficient coding.
Example	:DATA:PATTern:PACKed 1

:DATA:PATTern:IMPort

Syntax :DATA:PATTern:IMPort <pattern-name>, <file-name>

Description This command imports an N4903B J-BERT pattern or M8020A/M8030A pattern file.

M8020A/M8030A File Format:

File content consists of a short header and the actual symbol sequence.

The file content looks like the following:

Version=<version>
Use=<num-symbols>[,<symbol-coding>[,<mask-used>[,<squelch-used>,<error-used>]]]
[Description=<description>]
[Pack=<packing>]
Data=
<data>

Optional lines or fields are enclosed in square brackets. Format and notation is intended to follow the concepts of the other :DATA:PATTern SCPI commands.The first line must include a line with the file format version. Accepted value for <version> is M8000 1.0.0 or M8000 1.0.1.

<error-used> field requires version 1.0.1.

Note that if <error-used> is 1, <mask-used> and <squelch-used> must both be 0. Also the coding must be BIT or B8B10. B128B130 and B128B132 are not supported.

The next line describes the number of symbols, symbol coding, mask usage and squelch usage as described in :DATA:PATTern:USE command.

The next line allows specifying a description.

The next line allows changing the packing of the actual <data> part similar to the :DATA:PATTern:PACK command. If not specified, 8 is assumed.

The <data> in a separate line contains the sequence of bytes as specified by the packing.

Example File:

Version=M8000 1.0.0
Use=4
Pack=1
Data=
1010

N4903B J-BERT Patterns

N4903B J-BERT patterns are always imported as bit coded patterns with no mask and no squelch. The B part of an alternate pattern will be imported as <pattern-name>B.

The following is an example of an N4903B J-BERT pattern import:

Example DATA:PATT:IMP "current:x","C:\N4903B\Pattern\xxx.ptrn"

:DATA:PATTern:EXPort

Syntax :DATA:PATTern:EXPort <pattern-name>, <file-name>

Description This command exports a pattern to a file in the M8020A/M8030A import file format.

Example :DATA:PATT:EXP "current:x","C:/Users/jsmith/Desktop/test"

:DATA:SEQuence Subnode

With this subsystem sequencing can be changed programmatically. For additional information, refer to Creating Pattern Sequences on page 49.

This subnode has the following SCPI structure:

```
:DATA
└:SEQuence
  ├:NEW
  ├:NAMes?
  ├:BIND[?]
  ├:VALue[?]
  ├:DELete
  ├:DALL
  ├:BREak
  ├:RESTart
  │ └:IDENtifier
  └:BLOCk?
    └:IDENtifier?
```

This subnode has the following commands:

Table 87

Name	Description under
:NEW	:DATA:SEQuence:NEW on page 432
:NAMes?	:DATA:SEQuence:NAMes? on page 432
:BIND[?]	:DATA:SEQuence:BIND[?] on page 432
:VALue[?]	:DATA:SEQuence:VALue[?] on page 433
:DELete	:DATA:SEQuence:DELete on page 434
:DALL	:DATA:SEQuence:DALL on page 434
BREak	:DATA:SEQuence:BREak on page 434
:RESTart	:DATA:SEQuence:RESTart on page 434

Name	Description under
:RESTart:IDENtifier	:DATA:SEQuence:RESTart:IDENtifier on page 435
:BLOCk?	:DATA:SEQuence:BLOCk? on page 435
:BLOCk:IDENtifier	:DATA:SEQuence:BLOCk:IDENtifier? on page 435

:DATA:SEQuence:NEW

Syntax :DATA:SEQuence:NEW <sequence-name>

Input Parameters <sequence-name>: Specify sequence name.

Description This command is used to create a new sequence.

Example :DATA:SEQ:NEW ‘MySequence’

:DATA:SEQuence:NAMes?

Syntax :DATA:SEQuence:NAMes?

Description This command returns the sequence names currently in use.

Example :DATA:SEQ:NAM?
“Generator”,“Analyzer”,“MySequence”

:DATA:SEQuence:BIND[?]

Syntax :DATA:SEQuence:BIND <sequence-name>,<identifier-list>

:DATA:SEQuence:BIND? <sequence-name>

Input Parameters <sequence-name>: Specify sequence name.

<identifier-list>: Specify location or group name identifier.

Return Range <identifier-list>

Description This command binds the identifiers to a specified sequence. The identifier is either a location or a group name identifier. If locations are already used in another sequence, they get re-assigned to this sequence.

This SCPI is only for M8041A, M8051A and M8195A. However, for M8195A, it is only valid for Data Out locations.

Example :DATA:SEQ:BIND ‘MySequence’,‘M1.DataIn1’

:DATA:SEQuence:VALue[?]

Syntax :DATA:SEQuence:VALue <sequence-name>,<sequence-string>

:DATA:SEQuence:VALue? <sequence-name>

Input Parameters <sequence-name>: Specify sequence name.

<sequence-string>: Enter definite length block and pattern sequence XML string.

Return Range <sequence-string>

Description This command is used to enter a sequence string consisting of the pattern sequence parameters for the specified sequence name and download it to the hardware. A definite length block must be entered indicating the length of the string to download.

NOTE **It is recommended to edit the sequence string in the Sequence Editor in either “UI” or “<Xml>” mode and then copy/paste the xml sequence string.**

In the following example, #3340 defines the length of the string:

#3 indicates that a definite length arbitrary data block follows and its length is specified in 3 digits.

340 indicates that 340 bytes follow. In this example, the bytes are all readable ascii characters.

Example :DATA:SEQuence:VALue 'Generator',#3342<?xml version="1.0" encoding="utf-16"?><sequenceDefinition xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.agilent.com/schemas/M8000/DataSequence"><description /><sequence><loop><block length="128"><prbs polynomial="2^7-1" /></block></loop></sequence></sequenceDefinition>

:DATA:SEQuence:DELete

Syntax	:DATA:SEQuence:DELete <sequence-name>
Input Parameters	<sequence-name>: Specify sequence name.
Description	This command deletes the specified sequence.
Example	:DATA:SEQ:DEL 'MySequence'

:DATA:SEQuence:DALL

Syntax	:DATA:SEQuence:DALL
Description	This command deletes all sequences. This is a useful function if you do not want to use the reset sequence ("Generator", "Analyzer").
Example	:DATA:SEQ:DALL

:DATA:SEQuence:BREak

Syntax	:DATA:SEQuence:BREak 'identifier'
Input Parameters	<identifier>: 'M*.System'
Description	This command sends a break signal addressed by the identifier. This will leave a loop and execute the next block or execute an arbitrary jump to another block.
Example	:DATA:SEQ:BRE 'M1.System'

:DATA:SEQuence:RESTart

Syntax	:DATA:SEQuence:RESTart <sequence-name>
Input Parameters	<sequence-name>: Specify sequence name.
Description	This command restarts the specified sequence (jumps to the first block). For example, this command is useful for ED sequences. Usually there is a synch block first and then a loop to compare the synchronized pattern. Performing a restart there causes a synchronization of the sequence.
Example	:DATA:SEQ:REST 'EdSequence'

:DATA:SEQuence:RESTart:IDENtifier

Syntax: :DATA:SEQuence:RESTart:IDENtifier <identifier>

Input Parameters: <identifier>: 'M*.DataIn1', 'M*.DataIn2', 'M*.DataIn', 'M*.DataOut', 'M*.DataOut1' and 'M*.DataOut2'

Description: This command restarts the specified sequence (jumps to the first block). For example, this command is useful for ED sequences. Usually there is a synch block first and then a loop to compare the synchronized pattern. Performing a restart there causes a synchronization of the sequence.

Example: :DATA:SEQ:REST:IDEN ‘M1.DataIn1’

:DATA:SEQuence:BLOCk?

Syntax: :DATA:SEQuence:BLOCk? <sequence-name>

Input Parameters: <sequence-name>: Specify sequence name.

Description: This command returns the block of sequences currently executed.

Example: :DATA:SEQuence:BLOCk? ‘Generator’

:DATA:SEQuence:BLOCk:IDENtifier?

Syntax: :DATA:SEQuence:BLOCk:IDENtifier? <identifier>

Input Parameters: <identifier>: 'M*.DataIn1', 'M*.DataIn2', 'M*.DataIn', 'M*.DataOut', 'M*.DataOut1' and 'M*.DataOut2'

Description: This command returns the block of sequences currently executed for a specific location.

Example: :DATA:SEQ:BLOC:IDEN? ‘M1.DataIn1’

:DATA:SYNC Subnode

This subnode has the following SCPI structure:

```
:DATA
└:SYNC
  ├[:ONCE]
  ├:TYPE[?]
  └:THReshold[?]
```

This subnode has the following commands:

Table 88

Name	Description under
[:ONCE]	:DATA:SYNC[:ONCE] on page 436
:TYPE[?]	:DATA:SYNC:TYPE[?] on page 437
:THReshold[?]	:DATA:SYNC:THReshold[?] on page 437

:DATA:SYNC[:ONCE]

Syntax :DATA:SYNC[:ONCE] <identifier>

Input Parameters <identifier>: 'M*.DataIn1' or 'M*.DataIn2'

Description This command initiates a pattern resynchronization.

This SCPI is only for M8041A, M8051A and M8062A.

Example :DATA:SYNC ‘M1.DataIn1’

:DATA:SYNC:TYPE[?]

Syntax	:DATA:SYNC:TYPE ‘identifier’, AUTO|MANual
	:DATA:SYNC:TYPE? ‘identifier’
Input Parameters	<identifier>: 'M*.DataIn1' or 'M*.DataIn2'
	AUTO|MANual: Enable/disable automatic resynchronization.
Return Range	AUTO|MAN
Description	This command enables/disables automatic resynchronization.
	This SCPI is only for M8041A, M8051A and M8062A.
Example	:DATA:SYNC:TYPE ‘M1.DataIn1’, AUTO

:DATA:SYNC:THReshold[?]

Syntax	:DATA:SYNC:THReshold ‘identifier’, <NRf>
	:DATA:SYNC:THReshold? ‘identifier’
Input Parameters	<identifier>: 'M*.DataIn1' or 'M*.DataIn2'
	<NRf>: Set the threshold level.
Return Range	1E-8 to 1E-1
Description	This command sets the threshold level of error ratio at which synchronization is successful. Exceeding this BER threshold triggers a pattern synchronization when in automatic sync mode.
	This SCPI is only for M8041A, M8051A and M8062A.
Example	:DATA:SYNC:THR ‘M1.DataIn1’, 1E-5

:DATA:LINecoding Subnode

The :DATA:LINecoding subnode controls which line coding is being used when sending or receiving the digital data.

This subnode has the following SCPI structure:

```
:DATA
└:LINecoding
  ├[:VALue][?]
  ├:PAM4:MAPPing[?]
  └:PAM4[:SYMBol]:LEVel(1|2)[?]
```

This subnode has the following commands:

Table 89

Name	Description under
:DATA:LINecoding[:VALue][?]	:DATA:LINecoding[:VALue][?] on page 438
:DATA:LINecoding:PAM4:MAPPing[?]	:DATA:LINecoding:PAM4:MAPPing[?] on page 439
:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2)[?]	:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2)[?] on page 439

:DATA:LINecoding[:VALue][?]

Syntax: :DATA:LINecoding[:VALue] ‘identifier’, NRZ | PAM4

:DATA:LINecoding[:VALue]? ‘identifier’

Input Parameters: <identifier>: M*.DataOut1, M*.DataOut2, M*.DataOut3 and M*.DataOut4

Description: The command selects the line coding to be used. The availability of individual selections depends on the hardware capabilities.

This SCPI is only for M8195A.

Example: :DATA:LIN ‘M1.DataOut1’, PAM4

:DATA:LINecoding:PAM4:MAPPing[?]

Syntax :DATA:LINecoding:PAM4:MAPPing ‘identifier’, NONE | GRAY

:DATA:LINecoding:PAM4:MAPPing? ‘identifier’

Input Parameters <identifier>: M*.DataOut1, M*.DataOut2, M*.DataOut3 and M*.DataOut4

Description The command selects how the pattern bits are mapped to symbol numbers.

- Select NONE if the mapping shall be ‘transparent’. This is useful if the pattern does already contain the required coding scheme.
- Select GRAY if consecutive pairs of bits shall be Gray coded in order to map to the PAM-4 symbols.

This SCPI is only for M8195A.

Example :DATA:LIN:PAM4:MAPP ‘M1.DataOut1’, NONE

:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2)[?]

Syntax :DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2) ‘identifier’, <NRf>

:DATA:LINecoding:PAM4:MAPPing? ‘identifier’

Input Parameters <identifier>: M*.DataOut1, M*.DataOut2, M*.DataOut3 and M*.DataOut4

Range Symbol 1 Level: 1% to 65%

Symbol 2 Level: 34% to 99%

Description The command controls the PAM-4 level mappings. It can be used to adjust the actual output level that is being generated by a data output for a specific PAM-4 symbol.

The levels of PAM-4 symbol 0 and 3 cannot be adjusted and are always kept at the value of 0% and 100% of the actual output voltage swing.

It is best practice to configure symbol level 1 and 2 in a combined remote command transaction (see example below) to minimize hardware reconfiguration duration. This is especially important when the location is addressing a data output that is located on an arbitrary waveform generator module (e.g. M8195A).

This SCPI is only for M8195A.

Example :DATA:LIN:PAM4:LEV1 ‘M1.DataOut1’, 35pct;LEVel2 ‘M1.DataOut1’, 70pct

Index

Symbols

A

B

C

D

E

F

G

I

P

T

V

X

This information is subject to change without notice.
© Keysight Technologies 2016
Edition 5.1, March 2016

M8020-91040

www.keysight.com