M8000 Series of BER Test Solutions

M8020A High-Performance BERT

M8040A High-Performance BERT 64 GBd

M8050A High-Performance BERT 120 GBd

KEYSIGHT

PROGRAMMING GUIDE

Notices

Trademarks

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

Manual Part Number

M8000-91B09

Edition

Edition 19.1, December 2025

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.

Keysight Cybersecurity

Product and Solution Cybersecurity

Keysight is dedicated to ensuring the cybersecurity of its products and solutions. For detailed information, visit Product and Solution Cybersecurity.

Report a Product Cybersecurity Issue

If you encounter a cybersecurity issue with a Keysight product, report it immediately at: Product Cybersecurity Issue Reporting.

Responsible Disclosure Program

Keysight encourages responsible disclosure of security vulnerabilities. For more details, visit Responsible Disclosure Program.

Product Software Updates

Keysight releases periodic software updates to fix known defects, incorporate product enhancements, and address cybersecurity vulnerabilities, if any. Ensure that your product software is always up to date. To search for software updates for your product, visit the Keysight Technical Support page at: https://www.keysight.com/find/M8000.

User Documentation

User documentation includes comprehensive information on features that may impact the secure deployment, use, and decommissioning of our products. This includes details on exposed network ports, the use of cryptography and authentication, and firmware security settings, if necessary.

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.

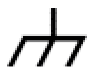
Instruments that appear damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.

External Connections

Any other instruments connected to this instrument shall be approved to a suitable safety standard and must include reinforced insulation from hazardous voltages, in particular mains.

Safety Symbols

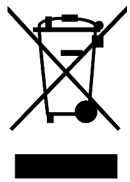
Table 1 Safety Symbols

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.
	Contains parts or assemblies susceptible to damage by electrostatic discharge (ESD). Use electrostatic discharge protective handling procedures to avoid malfunctions or potential damage to the instruments.
	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.
	Indicates that the product was tested and has met the certification requirements for electrical, plumbing and/or mechanical products.

Symbol	Description
CE CAN ICES/NMB-001(A) ISM GRP 1-A	The CE mark is a registered trademark of the European Community. This CE mark shows that the product complies with all the relevant European Legal Directives. CAN ICES/NMB-001(A) - This ISM device complies with the Canadian ICES-001(A). Cet appareil ISM est conforme a la norme NMB-001(A) du Canada. ISM GRP 1-A - This is an Industrial Scientific and Medical (ISM) Group 1 Class A product.
	This symbol on all primary and secondary packaging indicates compliance to China standard GB 18455-2001.

Compliance and Environmental Information

Table 2 Compliance and Environmental Information

Safety Symbol	Description
	The crossed out wheeled bin symbol indicates that separate collection for waste electric and electronic equipment (WEEE) is required, as obligated by DIRECTIVE 2012/19/EU and other National legislation. See http://about.keysight.com/en/companyinfo/environment/takeback.shtml to understand your Trade in options with Keysight in addition to product takeback instructions.

Contents

1 Programming Basics

Modular Configuration

In order to describe remote programming for the M8020A, M8040A and M8050A, the modular configuration must be discussed.

The M8020A, M8040A and M8050A 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 M8070B 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
- M8062A 32Gb/s Front-end for J-BERT M8020A High-Performance BERT

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

For complete details on the features and hardware components of each of the above mentioned modules, refer to *M8020A 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 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 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 3 M8020A configuration for 32 Gb/s BERT

M8040A Modular Configuration

The M8040A is a modular test solution which simplifies accurate receiver characterization of devices operating up to 32 and 64 GBd with NRZ and PAM4. It supports the following modules.

- M8045A pattern generator
- M8046A analyzer
- M8057A remote head

The following section describes and illustrates various setup combinations in which you can install the M8040A modules.

M8045A Pattern Generator Module

The M8045A module can be a one or two data channel system (a second channel can be added with license). A one channel instrument has to be returned to the factory for installing the second channel (hardware) and license. It occupies three slots of the 5-slot M9505A AXIe chassis. The following figure illustrates an M8045A module two data channel system) installed in an M9505A AXIe chassis.

M8045A Module
(2 Channels)

Figure 4 M8045A configuration for 2 channels

NOTE The M8045A module must be installed in slots 1 through 3 of the M9505A AXIe chassis for proper local bus communication unless the M9537A AXIe Embedded Controller is installed (must be in slot 1).

M8046A Analyzer Module

The M8046A module occupies single slot of the 5-slot M9505A AXIe chassis. The following figure illustrates an M8045A module with M8046A module installed in an M9505A AXIe chassis.

Figure 5 M8045A and M8046A configuration

M8057A Remote Head

The M8057A remote head is an external box which can be connected to each channel of M8045A module, using the matched pair of cables. It helps in minimizing signal degradations caused by lossy channels. The following figure illustrates an M8057A remote head connected with one channel of M8045A module.

Figure 6 M8045A, M8046A and M8057A configuration

M8050A Modular Configuration

The M8050A is a modular test solution which enables accurate characterization of receivers used in next-generation data center networks and server interfaces. With uncompromised signal integrity, support for NRZ, PAM4, PAM6, and PAM8 signals, and data rates up to 120 GBd, the flexible architecture of the M8050A supports 1.6T pathfinding as well as other leading-edge technologies. It supports the following modules.

The M8050A high-performance BERT is one part of the Keysight M8000 Series of BER test solutions. It can be combined with other hardware and software of the M8000 Series.

The M8050A is a modular instrument which supports the following modules:

- M8042A Pattern Generator Module
- M8043A Error Analyzer Module
- M8009A Clock Module
- M8058A/M8059A/M8068A/M8069A Remote Head for M8042A Pattern Generator
- M8052A Remote Head for M8043A Error Analyzer

The following section describes and illustrates various setup combinations in which you can install the M8040A modules.

M8042A Pattern Generator

The M8042A pattern generator module operates from 2 to 120 GBd. It is available as one channel or two channel version. You can select three symbol rate ranges.The M8042A is an instrument module that can be installed into the M9505A 5- slot AXIe chassis. The one channel version occupies two slots while the two channel version occupies three slots in he M9505A 5- slot AXIe chassis. The M8042A requires the clock module with jitter modulation M8009A, and one remote head for each data output channel. For operation above 64.8 GBd the 120 GBd pattern generator remote head M8059A or M8069A is required.

For the following generator functions a module option is required:

- Pattern generation up to 32 GBd for NRZ and PAM4 (M8042A-G32)
- Pattern generation up to 64 GBd for NRZ and PAM4 (M8042A-G642)
- Pattern generation up to 120 GBd for NRZ and PAM4 (M8042A-G12)
- De-emphasis, module-wide license (M8042A-0G4)
- PAM3 Encoding for USB 4v2 (M8042A-0P3)

- PAM6 Encoding for 224 Gbps Interfaces (M8042A-0P6)
- PAM8 Encoding for 224 Gbps Interfaces (M8042A-0P8)

The following figure displays the front panel of the M8042A module:

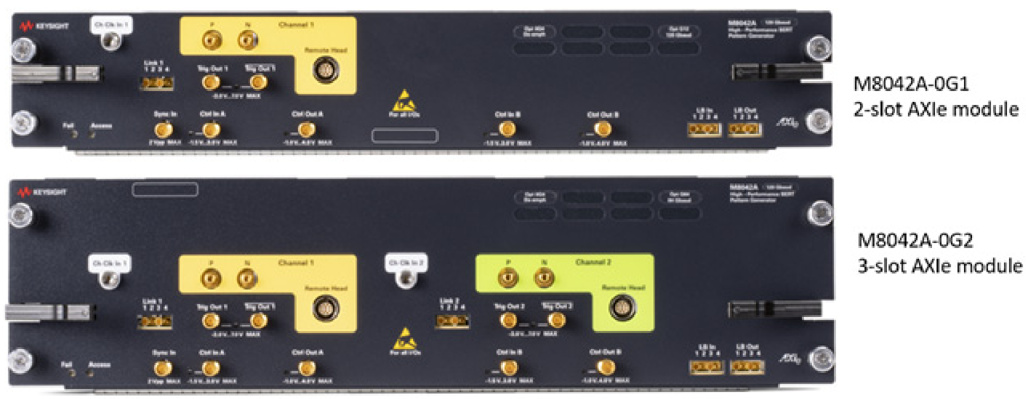


Figure 7 M8042A module front panel

M8043A Error Analyzer

The M8043A error analyzer module operates from 2.0 to 64.4 GBd. It requires the remote head M8052A. The M8043A is an instrument module that can be installed into the M9505A 5-slot AXIe chassis. This module occupies two slots in the M9505A 5-slot AXIe chassis and is always recommended to install on the slots above the M8042A and M8009A modules. The M8043A comes always with a built-in clock recovery.

Integrated equalization with FFE and CTLE is optional. For error analysis above 64.4 GBd we recommend using the UXR0802A/04A with control from M8070B. See below for more details.

Using the remote input of the M8043A module directly is prohibited.

The following analyzer options are provided:

- Error analysis up to 32.4 GBd for NRZ and PAM4 (M8043A-A32)
- Error analysis up to 64.4 GBd for NRZ and PAM4 (M8043A-A64)
- Equalization, license (M8043A-0A3)

The following figure displays the front panel of the M8043A module:

Figure 8 M8043A module front panel

M8009A Clock Generator

Keysight's M8009A clock generator is designed as sample clock source for the M8042A pattern generator. The M8009A clock module with integrated jitter modulation operates from 2 to 60 GHz. It can be locked to external reference clocks.

For the following functions a module option is required:

- Advanced jitter modulation for up to two channels, license (M8009A-0G3)
- Reference clock multiplier, license (M8009A-0G6)
- Clock Generator two channel 60 GHz, 1 slot AXIe (M8009A-062). This option is only available only for the new hardware i.e. M8009A-062.

The M8009A is an instrument module that can be installed into the M9505A 5-slot AXIe chassis. It comes as a 1-slot AXIe module, which allows the M8009A plus up to two M8042A pattern generator modules to be plugged into a single 5-slot AXIe chassis.

The following figure shows the front panel of the M8009A clock module:

Figure 9 M8009A-061 module front panel

The following figure shows the front panel of the M8009A-062 clock module with the new “Clk Out 32G” port:

Figure 10 M8009A-062 module front panel

M8058A/M8059A/M8068A/M8069A Remote Head for M8042A Pattern Generator

The M8058A/M8059A/M8068A/M8069A remote head includes an external amplifier that is used in combination with the M8042A Pattern Generator. It helps in minimizing signal degradations caused by lossy channels. Four remote heads are available for the pattern generator module M8042A.

The M8068A and M8069A offer significantly higher output amplitude than the M8058A and M8059A.

The following figure shows the M8058A and M8059A remote heads:

M8058A - Pattern Generator Remote Head up to 64.8 GBd

M8059A - Pattern Generator Remote Head up to 120 GBd

Figure 11 M8058A and M8059A remote heads

The following figure shows the M8068A and M8069A remote heads:

M8068A - Pattern Generator Remote Head up to 64.8 GBd   M8069A - Pattern Generator Remote Head up to 120 GBd

Figure 12   M8068A and M8069A remote heads

The 64 GBd M8058A/M8068A remote head is shown at the left side and this provides 1.85 mm connectors. It is used to accommodate close connection to the device under test for symbol rates up to 64.8 GBd.

The 120 GBd M8059A/M8069A remote head is shown on the right side and provides 1.0 mm connectors and is used to accommodate close connection to the device under test for symbol rates up to 120 GBd.

The three cables on the back side of the remote heads are used to connect with the M8042A pattern generator module and are not removable.

Please refer “*Tips for Preventing Damage*” document for mounting remote head cables.

M8052A Remote Head for M8043A Error Analyzer

The M8052A remote head includes an external amplifier and a CTLE that is used in combination with the M8043A Error Analyzer.

The following figure shows the M8052A remote head:

Figure 13 M8052A remote head

Two cables on the back side of the remote head are used to connect with the M8043A error analyzer and are not removable.

M8050A Module Setup

This section describe how to connect M8042A pattern generator and M8009A clock module in order to prepare a single-channel and two-channel systems.

The following table shows the cables required for preparing a single-channel and two-channel systems:

Clk Module Connections	Part No.	Option	Description
DAT Clk Cable channel 1	M8199-61624		Clock Cable for Pattern Generator M8042A, Channel 1
DAT Clk Cable channel 2	M8199-61624		Clock Cable for Pattern Generator M8042A, Channel 2
Sync Cable 3.5mm	M8042-61623	M8042A-803	Synchronization Cable, Semi-rigid for Pattern Generator M8042A and M8009A

Preparing Single-Channel System

For a single channel system, you would need a M8042A pattern generator (one channel) and M8009A clock module.

Follow the given steps to make the single-channel system:

1. Connect M8009A Channel Clock Out 1 port with the M8042A Channel Clock In 1 port using M8199-61624 cable or a semi-rigid (M8042A-801) cable.
2. Connect M8009A Sync Out A port with the M8042A Sync In port using a 3.5 mm sync cable.

Figure 14 Single-channel system

Preparing Two-Channel System

CASE – 1: Preparing two-channel system using one M8042A pattern generator (two channels) and one M8009A clock module

Follow the given steps to prepare the two-channel system:

1. Connect M8009A Channel Clock Out 1 port with the M8042A Channel Clock In 1 port using M8199-61624 cable or a semi-rigid (M8042A-801) cable.
2. Connect M8009A Channel Clock Out 2 port with the M8042A Channel Clock In 2 port using M8199-61624 cable or a semi-rigid (M8042A-802) cable.
3. Connect M8009A Sync Out A port with the M8042A Sync In port using a 3.5 mm sync (M8042-61623) cable.

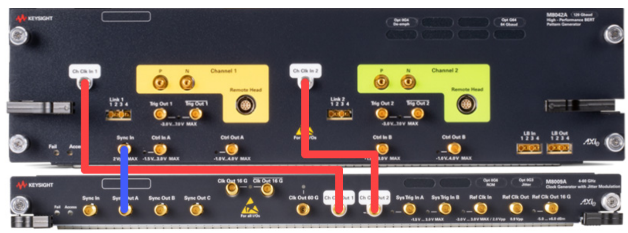


Figure 15 Two-channel system using one M8042A pattern generator (two channels) and one M8009A clock module

NOTE

Both channels must be properly clocked in order to arm the system. To ensure proper arming, connect both channels **Clock In** to the respective **Clock Out** on the clock module.

CASE – 2: Preparing two-channel system using two M8042A pattern generators (one channel) and one M8009A clock module

Follow the given steps to prepare the two-channel system:

1. Connect M8009A Channel Clock Out 1 port with the primary M8042A Channel Clock In 1 port using M8199-61624 cable or a semi-rigid (M8042A-801) cable.
2. Connect M8009A Channel Clock Out 2 port with the secondary M8042A Channel Clock In 1 port using M8199-61624 cable or a semi-rigid (M8042A-802) cable.

3 Connect M8009A Sync Out A port with the primary M8042A Sync In port using a 3.5 mm sync (M8042-61623) cable.

4 Connect M8009A Sync Out B port with the secondary M8042A Sync In port using a 3.5 mm sync (M8042-61623) cable.

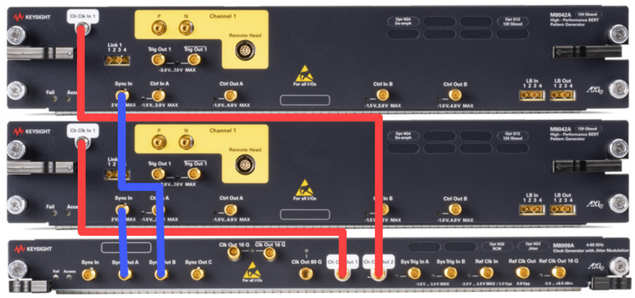


Figure 16 Two-channel system using two M8042A pattern generator (one channel) and one M8009A clock module

NOTE

When preparing a two-channel system, mixing one-channel and two-channel M8042A modules in a combined M8009A/M8042A instrument is not supported.

Preparing Four-Channel System

For a four-channel system, you will require two M8042A two-channel pattern generators, two M8009A clock modules, two 5-slot chassis, and the necessary connecting cables. Install one M8009A and M8042A in each of the chassis. Make sure the M8009A is in a lower slot number than M8042A.

Select one of the two chassis as primary and the other as secondary. This is important as the following cabling instructions rely on this setup.

Follow the steps outlined below to set up the four-channel system:

1 Connect M8009A Sync Out A port with the M8042A Sync In port using a 3.5 mm sync cable.

2 Connect M8009A Channel Clock Out 1 port with the M8042A Channel Clock In 1 port using either an M8199-61624 cable or a semi-rigid (M8042A-801) cable.

3 Connect M8009A Channel Clock Out 2 port with the M8042A Channel Clock In 2 port using either an M8199-61624 cable or a semi-rigid (M8042A-802) cable.

4 Connect the M8009A Ref Clk Out 16 G port to the M8009A Ref Clk In port using either an M8199-61624 cable or a semi-rigid M8042A-802 cable. Additionally, connect the M8009A Sync Out B port to the M8009A Sync In port. Finally, connect the Clock Out port of the chassis defined as primary to the Clock In port of the System Module defined as secondary chassis.

5 Use a local bus (LB) cable (M8041-61601) to connect the M8042A LB Out port to the M8042A LB In port on the second chassis.

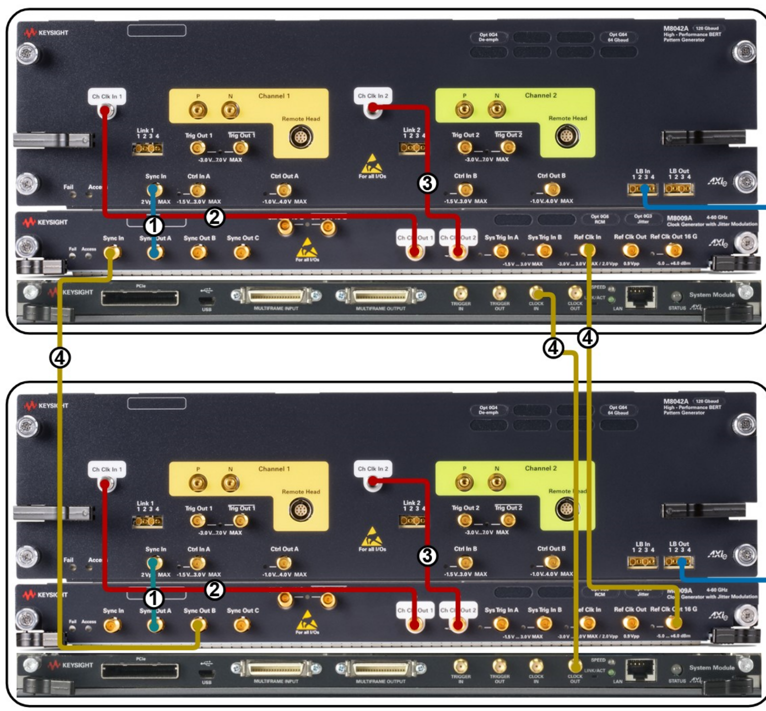


Figure 17 Four-channel system using two M8042A pattern generators (two channels) and two M8009A clock modules

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 of the channels, such as high level, low level and so on. 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/M8040A/M8050A platform supports a plugin interface. C# assemblies which implement certain interfaces are recognized by the software and integrated into the M8070B 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 M8070B 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/M8040A/M8050A 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, you 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/M8040A Getting Started Guide for information on terminating the M8062A 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/M8040A/M8050A data and clock output ports, four parameters are changed:

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

The M8020A 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

Reading the M8020A/M8040A/M8050A’s Status

Reading the M8020A/M8040A/M8050A’s Status - Concepts

The M8020A/M8040A/M8050A 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/M8040A/M8050A that can lead to questionable results.
- Operation Status Register
  The Operation Status Register indicates when certain operations have been completed.

How the M8020A/M8040A/M8050A 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/M8040A/M8050A's status system.

Overview of the M8020A/M8040A/M8050A’s Status System

The M8020A/M8040A/M8050A 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/M8040A/M8050A Status System Structure

The M8020A/M8040A/M8050A'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/M8040A/M8050A's events and conditions. All other individual registers are used to determine the specific events or conditions.

Status Register Group Model

Figure 18 on page -39 illustrates the typical structure of a status register.

The M8020A/M8040A/M8050A instrument status register model follows the structure described in IEEE 488.2, section 11.4.2.

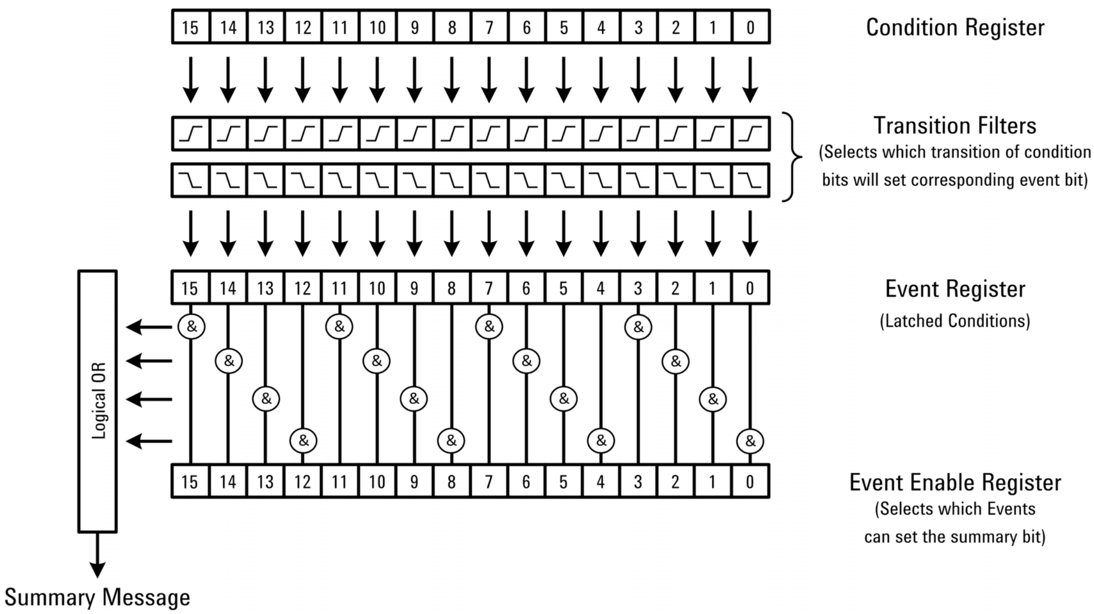


Figure 18 Status register structure

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/M8040A/M8050A Status Reporting Structure

Figure 19 on page -41 shows the M8020A/M8040A/M8050A's status reporting structure.

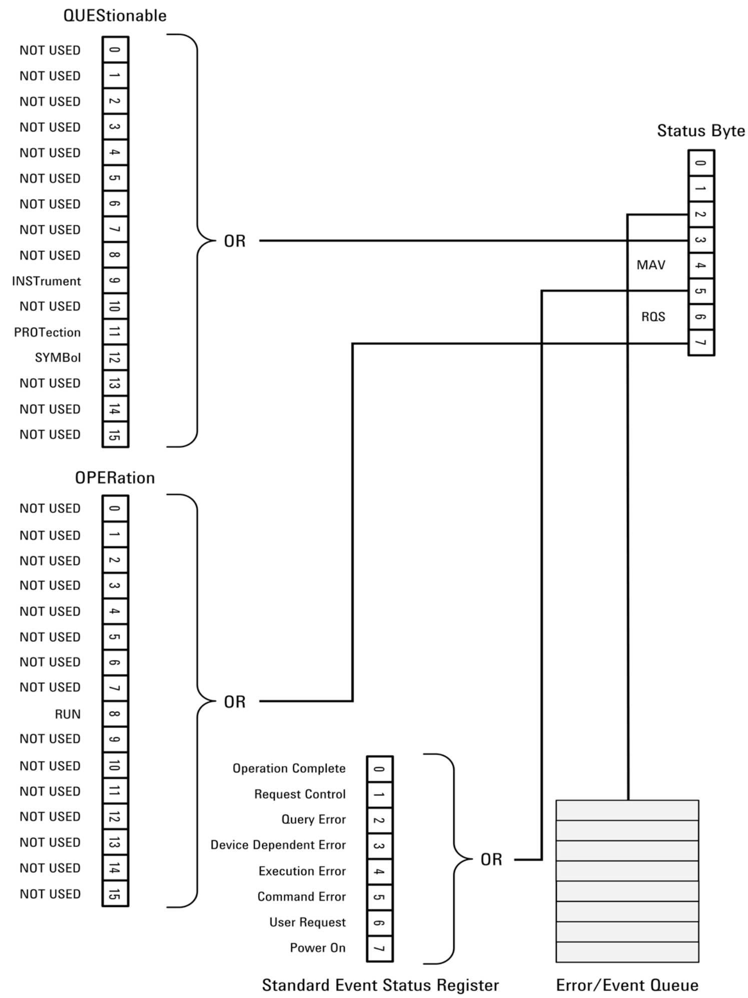


Figure 19 Status reporting structure

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 3 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 4 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 5 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 6 Operation status register descriptions

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

Program Snippet Extending the *opc? Functionality

Whenever changes are done that require the hardware to stop clock generation, a “*opc?” command does not wait long enough. It only waits until the clock generation is stopped and not when the clock generation is again started. You have to wait until the hardware is in run condition again (clocks are again running). In general, once you perform this kind of operation, you should execute “*opc?” command to make sure whether the operation is completed or not.

This can be polled in the status system, e.g. with a program snippet like this:

```
response = scpiAccess.Query("stat:oper:run:cond?");
var timeStart = DateTime.Now;
while ((response != "1") && ((DateTime.Now -
timeStart).TotalSeconds < 60))
response = scpiAccess.Query("stat:oper:run:cond?");
Assert.AreEqual("1", response);
```

This program snippet can be used after the following operations:

- Frequency change or anything related. Everything that is in the M8070B GUI under “ClkGen” and “Synthesizer” functional bocks.
- SSC parameter change (everything that is in the M8070B GUI under “ClkGen”, “SSC” functional blocks or the global SSC on/off)
- Changes in the sequence symbol width
- *rst or any other reset command or setting load, as they change all parameters.

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 545 for more information.

The :DATA:PATTern:USE SCPI command is used to specify the patterns and attributes. Refer to :DATA:PATTern:USE[?] on page 544 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 7 Plain bit coding without using mask or squelch

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

Table 8 Mask is used

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

Table 9 Squelch is used

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

Table 10 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 11 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 12 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 13 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 14 128/130 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 15 128/132 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 20 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 21 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 22 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 M8070B software interface as shown in Figure 23 on page -53.

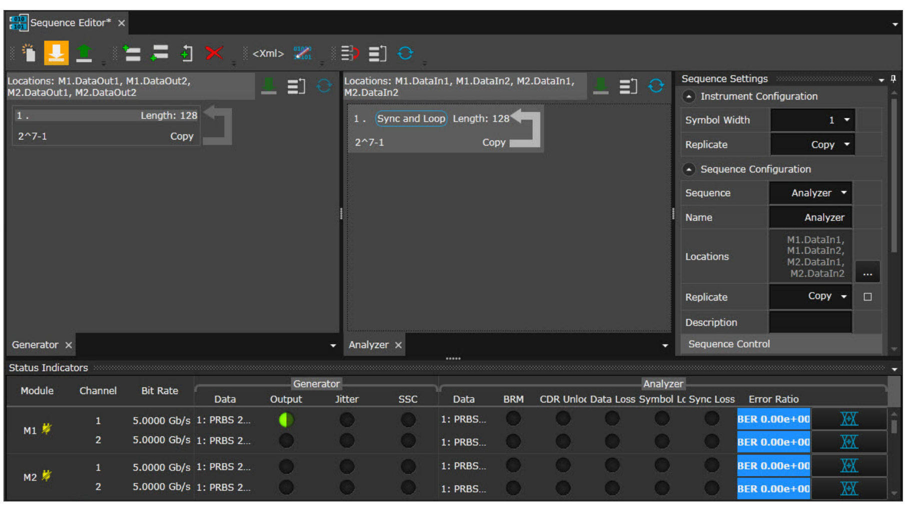


Figure 23 Sequence Editor in M8070B software interface

A default “Generator” and “Analyzer” sequence are configured in the Sequence Editor as shown in Figure 23 on page -53. 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 24 on page -54.

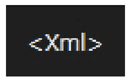


Figure 24 <Xml> button

Figure 25 on page -54 shows the XML elements for the default Generator and Analyzer. Clicking the UI button will return to the graphical interface.

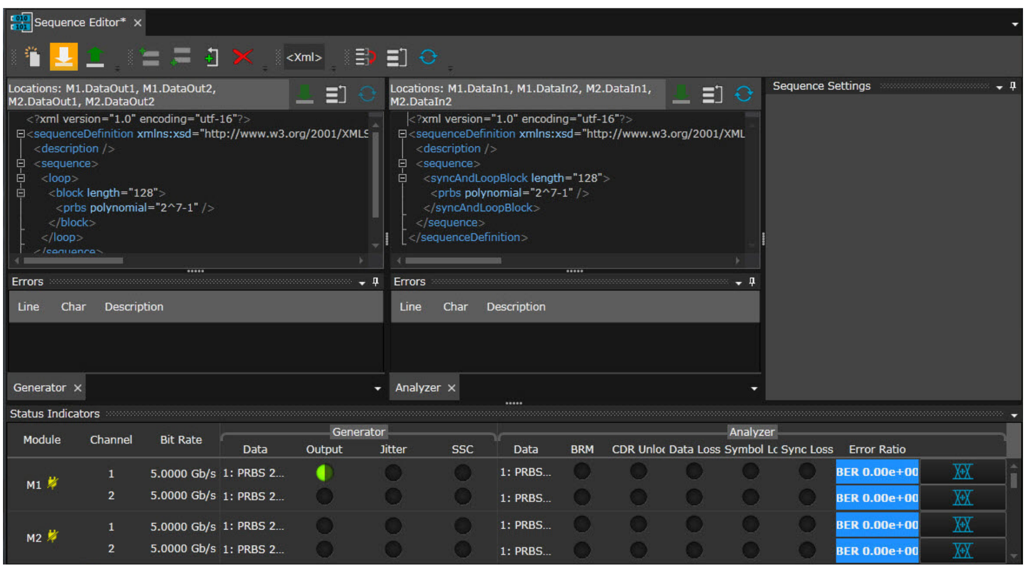


Figure 25 XML view

In UI view, click in a pattern sequence block to display its sequence settings as shown in Figure 26 on page -55.

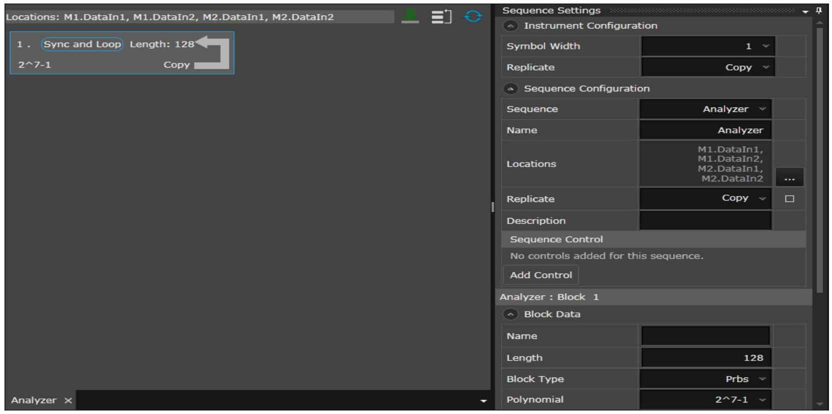


Figure 26 Sequence settings

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.keysight.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 16 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.

- Copy plus phase adjust: In this each location gets a copy of the pattern. Scrambler phases of the different locations are set differently. This is for the parallel side of a serial bus.

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:

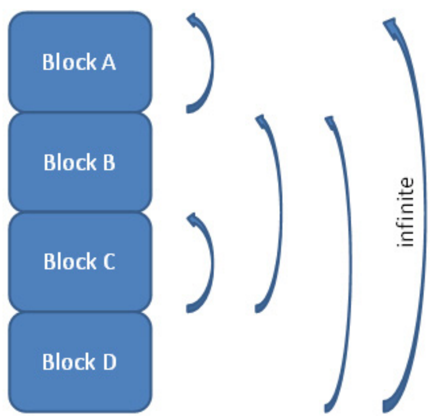


Figure 27 Loop nesting

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 ="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 M8070B 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 310 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 M8070B software interface, select **Patterns** > **Sequence Editor**.

2 Under the **Block Branches** settings, change **Condition** to **Break**.

Refer to :DATA:SEQuence:BREak on page 557 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 M8070B 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 M8070B 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

To select the physical layer protocol, the following node can be added

```
<phyProtocol name="PCIe3" />
```

The available PHY protocol choices are as follows:

- None (default)
- CCIX
- PCIe2
- PCIe3
- PCIe4
- PCIe5
- USB 3.0
- USB 3.1
- SATA3
- SAS2
- SAS3

These choices are further restricted by the module being used, and the installed licenses on the modules.

For M8020A, only the PHY protocols for which an LTSSM is available are supported (PCIe3, PCIe4, USB 3.0, and USB 3.1).

For M8045A, the allowed PHY protocols are PCIe3, PCIe4, and PCIe5.

For M8046A the availability is controlled by the licenses M8046A-0S2, M8046A-0S4 and M8046A-0S6 and the corresponding upgrade and trial licenses.

For M8042A, below PHY protocols are supported (LTSSM available):

- PCIe3
- PCIe4
- PCIe5
- PCIe6

The PHY protocol “None” is always available.

On M8040A modules, the PHY protocol selection does choose the coding specific configurations automatically and it will override any user configurations.

The M8040A modules do only support a symbolWidth of 1, and bit coded patterns. They do not support the coded patterns like M8020A.

Therefore, the patterns must contain a continuous sequence of bits as required by the PHY protocol. All protocol specific encoding and validity rules must be provided by the pattern.

Additionally the following requirements exist on M8040 setups:

- The patterns for M8045A must contain the align, filler, SKP OS symbol sequences as required by the standard being tested.

  For 8b10b patterns the pattern must contain the 10b symbols with a correct running disparity.
- The patterns for M8046A may not contain align, filler, SKP OS symbol sequences.

  For 8b10b patterns, the pattern must contain the 10b symbols, each symbol normalized to negative disparity.
- The granularity requirements are defined by the channel being used (512 bit for M8040A) and the PHY protocol that is selected. This leads to the requirement that in the pattern sequence (block length * loop count) must be a multiple of LeastCommonMultiple (512, protocol granularity).

The PHY protocol granularity is as follows:

- 10 for 8b10b
- 130 for 128b130b
- 132 for 128b132b
- 150 for 128b150b

For sequence blocks that do not match the granularity requirement, the loop count can be calculated as such:

- Pattern Granularity = LeastCommonMultiple (512, PHY protocol granularity).
- Required Block Length = LeastCommonMultiple (Pattern Granularity, Pattern Length).
- Loop count = Required Block Length / Pattern Length.
- The M8046A does support algorithmic PRBS patterns only when the PHY protocol is set to None.

On M8020A setups, 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.

General Parameters

These parameters are required for general setting up the LTSSM operation.

Name	Possible Values	Description	Used for DUT type
DUT Type	Add-in card System Board	Specifies which role the BERT should play during link training. It can either be an upstream device for testing a downstream port or vice versa.	Both
Target Speed	32.0GT/s 16.0GT/s 8.0GT/s	Determines the target speed when doping the speed change. The Target Speed 32.0GT/s) is applicable for M8040A only.	Both
Target State	Loopback via Recovery Loopback via Configuration	Determines the target state when bringing up the link.	Both
Trigger	LTSSM States	This parameter determines when the trigger output signal of the LTSSM block should be asserted. Additionally, this information can be translated into a trigger pulse on entering or leaving the specified state. This is done by the pulse shaper of the TCRL OUT connector and no direct functionality of the LTSSM.	Both
Clock Architecture	Common | Separate	Specifies the clock architecture whether common or separate. In case of common clock, a common RefClk is expected. The M8000 instrument and the DUT run on the same clock. The RefClk is provided by the 'Trig Out' of the clock module of the M8000 instrument. In case of separate clock, the DUT runs on its own RefClk provided by itself or by an external source. The 'Trig Out' of the M8000 instrument is not used as RefClk for the DUT.	Both
Loopback through	L0 / Recovery | Configuration | L0 / Recovery with Speed Bypass | Configuration with Equalization	Specifies if the real-time equalization is done while going into loopback through L0 / Recovery or through Configuration. When selecting Loopback through L0-Recovery, the LTSSM does a link training as before. Selecting Loopback through Configuration does a link training without real time equalization. When the Loopback through Configuration is selected, all preset parameters are disabled.	Both
Compliance Receive Bit	Deasserted Asserted	Specifies whether the compliance received bit is asserted or de-asserted in TS1.	Both

Name	Possible Values	Description	Used for DUT type
Ref Clock Architecture	Common, Separate	Defines the PCIe Reference Clock Architecture - common reference clock / separate reference clock.	Both
Link EQ states	Bypass Presets only Full	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 (i.e. individual cursor) requests should be accepted.	Both
Lane Number	0-31	The lane number being used.	Both
Link Number	0	The link number being used.	Add-in Card*
BERT Start Preset Gen3	P0-P9	This is the preset used by the BERTs TX port after switching to Gen 3 operation and when operating as an upstream device.	Both
RX Preset Hint	0 (-6 dB) 1 (-7 dB) 2 (-8 dB) 3 (-9 dB) 4 (-10 dB) 5 (-11 dB) 6 (-12 dB) 7 (reserved)	This is the preset hint being sent by the BERT to the DUT during phase 0 of the link equalization procedure. It's only used when the BERT operates as upstream device.	Add-in Card
DUT Start Preset Gen3	P0-P9	This is the preset the BERT transfers to the DUT in phase 0 of the link equalization procedure. It's only used when the BERT operates as an upstream device (2.5G to 8G)	Add-in Card
DUT Target Preset Gen3	P0-P9, P10, Cursor†	This is the preset the BERT requests the DUT to switch to during link equalization. Depends on the role the BERT is playing, this is done in either phase 2 or 3 of the link equalization training (8G)	Both
Select BERT Start Preset Gen4	User Defined LTSSM Defined LTSSM EQTS2 defined	Use BERT start preset specified by user or use resulting preset of phase 2 of 8GT/s TXEQ as start preset. LTSSM EQTS2 defined - preset received in the EQTS2 before the speed change. LTSSM will use preset received in EQTS2 if EQTS2 is sent. But if the endpoint does not send EQTS2, it will use the user-defined preset instead.	Add-in Card
BERT Start Preset Gen4	P0-P9	This is the preset used by the BERTs TX port after switching to Gen4 operation and when operating as an upstream device.	Add-in Card
DUT Start Preset Choice Gen4	System Board Defined User Defined	Use DUT start preset defined by System Board or user.	System Board
DUT Start Preset Gen4	P0-P9	This is the preset the BERT transfers to the DUT in phase 0 (Gen4) of the link equalization procedure. It's only used when the BERT operates as an upstream device.	Both

Name	Possible Values	Description	Used for DUT type
DUT Target Preset Gen4	P0-P9, P10, Cursor†	This is the preset the BERT requests the DUT to switch to during link equalization (Gen4). Depends on the role the BERT is playing, this is done in either phase 2 or 3 of the link equalization training.	Both
Select BERT Start Preset Gen5	User Defined LTSSM Defined LTSSM EQTS2 defined	Use BERT start preset specified by user or use resulting preset of phase 2 of 16GT/s TXEQ as start preset. LTSSM EQTS2 defined - preset received in the EQTS2 before the speed change. LTSSM will use preset received in EQTS2 if EQTS2 is sent. But if the endpoint does not send EQTS2, it will use the user-defined preset instead.	Add-in Card
BERT Start Preset Gen5	P0-P9	This is the preset used by the BERTs TX port after switching to Gen5 operation and when operating as an upstream device.	Add-in Card
DUT Start Preset Choice Gen5	System Board Defined User Defined	Use DUT start preset defined by System Board or user.	System Board
DUT Start Preset Gen5	P0-P9	This is the preset the BERT transfers to the DUT in phase 0 (Gen5) of the link equalization procedure. It's only used when the BERT operates as an upstream device.	Both
DUT Target Preset Gen5	P0-P9, P10, Cursor†	This is the preset the BERT requests the DUT to switch to during link equalization (Gen5). Depends on the role the BERT is playing, this is done in either phase 2 or 3 of the link equalization training.	Both
DUT Target Pre-Cursor @ Target Speed	0-63	This is the pre-cursor value that the BERT requests the DUT to use during link equalization in cursor mode at Target Speed. Depends on the role the BERT is playing this is done in either phase 2 or 3 of the link equalization training. If not specified 0 is used.	Both
DUT Target Main-Cursor @ Target Speed	0-63	This is the main cursor value that the BERT requests the DUT to use during link equalization in cursor mode at Target Speed. Depends on the role the BERT is playing, this is done in either phase 2 or 3 of the link equalization training. If not specified 63 is used.	Both
DUT Target Post-Cursor @ Target Speed	0-63	This is the post-cursor value that the BERT requests the DUT to use during link equalization in cursor mode at Target Speed. Depends on the role the BERT is playing, this is done in either phase 2 or 3 of the link equalization training. If not specified 0 is used.	Both
Speed Change Control	Dut Bert	Specifies whether the speed change to Gen3 (i.e. 8Gbps) speed will be initiated by DUT or BERT during link training. It's only used when the BERT operates as a downstream device. If not specified DUT will initiate the speed change and will also request BERT for the same.	System Board
Full Swing (FS)	24-63	This is the sum of the absolute values of all three cursors. Therefore, it specifies the granularity in which cursors can be controlled.	Both
DUT Preset	P0-P9, P10	This is the preset value that BERT requests the DUT to use when attempting link training through Configuration.	Both

Name	Possible Values	Description	Used for DUT type
DUT Start Preset Choice Gen 6	System Board Defined User Defined	Use DUT start preset defined by System Board or user.	System Board
DUT Initial Preset Gen 6	Q0-Q10	Specifies the preset the BERT transfers to the DUT in phase 0 of the link equalization procedure. It is only used when the BERT operates as an upstream device.	Both
DUT Target Preset Gen 6	Q0-Q10, Cursor†	This is the preset the BERT requests the DUT to switch to during link equalization (Gen5). Depends on the role the BERT is playing, this is done in either phase 2 or 3 of the link equalization training.	Both
DUT Target Pre-Cursor	0-63	This is the pre-cursor value that the BERT requests the DUT to use during link equalization in cursor mode at Target Speed.	Both
DUT Target Pre-Cursor 2	0-63	This is the pre-cursor value that the BERT requests the DUT to use during link equalization in cursor mode at Target Speed.	Both
DUT Target Main Cursor	0-63	This is the main cursor value that the BERT requests the DUT to use during link equalization in cursor mode at Target Speed.	Both
DUT Target Post-Cursor	0-63	This is the post-cursor value that the BERT requests the DUT to use during link equalization in cursor mode at Target Speed.	Both
State Timeout File	<xml_file>	Enables you to select and configure an XML file for state timeouts map. There are two types of XML files available for Addin Card and System Board which enables different DUTs. ▪ ConfigA ▪ ConfigB	Both

* When testing a System Board, this parameter must be loaded from the data received from the DUT.

† Valid for Target Speed only, not for transient speeds in between.

Static Parameters

The Low Frequency Parameter

Name	Value	Description
LF	8	The value which corresponds to the minimum differential value that can be generated by the transmitter.

Parameters being controlled by the Sequencer

These parameters are required for testing, but are controlled by the Sequencer. So they are not direct input parameters of the LTSSM.

Name	Possible Values	Description
DUT Test Pattern	Compliance Test Pattern Modified Compliance Pattern	Specified the test pattern being sent out by the Sequencer when the link is up (i.e. LTSSM is in loopback state).

LTSSM States

LTSSM transits through various states and sub states during link initialization, training, and management. The entry and exit of each of these states and the exchange of packets (Ordered sets) between the two devices during each state are as per the PCIe specifications. Following is a list of these states as per the PCIe specifications.

Detect.Active
Polling.Active
Polling.Configuration
Configuration.Linkwidth.Start
Configuration.Linkwidth.Accept
Configuration.Lanenum.Wait
Configuration.Lanenum.Accept
Configuration.Complete
Configuration.Idle
L0
Loopback.Active
Recovery.Equalization.Phase0
Recovery.Equalization.Phase1
Recovery.Equalization.Phase2
Recovery.Equalization.Phase3
Recovery.Speed

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 28 on page -74. When the sequence has been downloaded, this button turns green.

Figure 28 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[:SET]:VALue[?] on page 555.

3 Programming Examples

KEYSIGHT TECHNOLOGIES

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 M8070B 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”

M8070B Jitter Tolerance Measurement CSV Output

The following python script generates the results from all open jitter tolerance measurements to different CSV files. This script can be easily modified to generate the output of the highest-passing point at each jitter modulation frequency value measured. This script can be run from the M8070B Script Editor or can also be run externally.

```
# M8020A jtol CSV output
#
# Run from the M8070B script editor

def WaitForCompletion():
    res = M8000.Scpi.Query(":syst:err?")
    if (res != '0,"No error"'):
        print res

WaitForCompletion()
M8000.Scpi.Timeout = 120000

# loop through all jitter tolerance measurements
for id in M8000.Scpi.Query(":PLUG:JTOL:CAT?").Split(','):
    print "Measurement: " + id + ":"

    results = M8000.Scpi.Query(":PLUG:JTOL:FETC:DATA? " + id).Trim('(',')').Split('
    #print results

    # size of results
    print " " + str(len(results)) + " data items, " + str(int(len(results)/6)) + " rows;",

    # if there are any rows (and not just a blank measurement), then
    #  use the name of the jtol measurement as the filename
    #  this will overwrite the file if it already exists!
    #  the file will be located in \Program Files (x86)\Keysight\M8070B\bin\, mayb
    if (int(len(results)/6) == 0):
        # nothing to write to file
        print "no file created"
    else:
        with open(id.Trim('"').Replace(' ','_') + ".csv", 'w') as f:

            # output header to file
            f.write("Modulation Frequency (GHz),Sinusoidal Jitter (UI),Number of E
Ratio,Measurement results\n")
```

```
        # iterate over the data
        #  first element ([0]) is a module identifier, discard
        #  items are then grouped in bundles of six, per header above
        #  we need to insert a comma or LF after each item, depending on pos
        for idx,data in enumerate(results):
            if (idx==0):
                # do nothing, this is the module identifier
                pass
            elif (idx%6==0):
                # output ending in a newline
                #print data
                f.write(data + "\n")
            else:
                # output ending in a comma
                #print data + ",",
                f.write(data + ",")

    # close file and move on!
    f.close()
    print "written to file"
```

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 118.

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/M8040A/M8050A 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/M8040A/M8050A 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/M8040A/M8050A as either

ON | OFF or 0 | 1. The M8020A/M8040A/M8050A 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
- Modules Supported by a Command

Instrument Model

SCPI guidelines require that the M8020A/M8040A/M8050A 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 144.

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.

Modules Supported by a Command

The SCPI command description lists all the modules which are supported by that particular command/query.

Here is an example:

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

This SCPI is applicable for M8041A, M8042A, M8045A, M8051A, M8054A, M8062A, M8194A, M8195A, M8196A and M8009A modules.

Sending Commands to the M8020A/M8040A/M8050A

Sending Commands to the M8020A/M8040A/M8050A - Overview

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/M8040A/M8050A 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 112 for details about /Socket, /Telnet, /Inst, /AutoID, /NoAutoID, /FallBack).

Table 17 Command line arguments

Argument	Description
/AutoId	Start in auto ID mode (this is the default) [optional]
/ChassisIdentifier <chassis id>	Defines a chassis identifier 1,2,3, ...
/EnvironmentVariable	Defines an environment variable for the M8070B software
/FallBack	Use auto ID mode if the communication parameters from the command line don't work [optional]
/IgnoreAwg	M8070B software don't grab M8194A, M8195A and M8196A modules
/IgnoreRts	M8070B software don't grab supported real-time scopes
/IgnoreIfs	M8070B software don't grab supported interference source
/IgnoreDca	M8070B software don't grab supported DCA
/Inst <instrument number>	Set the instrument number for VXI-11.3 and HiSLIP SCPI communication [optional]

Argument	Description
/Maintenance	Start the M8070B software in maintenance mode
/Minimize	Launches the M8070B GUI in minimized form **NOTE - The user need to decide whether to update the M8070B built in settings or not whenever the new version is updated.**
/NoAutoId	Do not start in auto ID mode - use communication parameters from command line [optional]
/NoDrivers	Do not load any module drivers. Only affects individually installable drivers.
/NoPlugins	Do not load any (graphical user interface)
/Normal	Start the M8070B software in normal mode (default)
/NoSplash	Do not show the splash screen
/rcl <RecallLastSetting>	Recall last used setting [optional]
/rst <FactoryPresetSetting>	Reset to factory default [optional]
/SafeMode	Do not load any plugins or individually installable module drivers
/SlotNumber	Defines a slot number; of the first module to be built an instrument
/Socket <socket port>	Set the socket port for the SCPI communication (only used with /NoAutoId) [optional]
/StartupConfiguration	Defines startup configuration for a specific instrument controlled by M8070B
/SystemConfigurationFile	Path to system configuration file (.m8kc) For details on how to use the system configuration file, refer to the *M8000 Series User Guide* and *Help*.
/Telnet <telnet port>	Set the telnet port for the SCPI communication (only used with /NoAutoId) [optional]
/UnprotectFactoryData	Disables the write protection of the factory data in the workspace being used
/VisaResource	Pass a visa resource to be used
/WorkspaceName	Addresses a workspace with the specified name - if it does no exist a new one is created

Communication

Depending on the command line arguments /Socket, /Telnet, /Inst, /AutoID, /NoAutoID, /FallBack, the M8070B 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 M8070B 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 M8070B 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 M8070B 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 18 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.
CLOCk	The CLOCk subsystem is used to detect clock source or to set the clock frequency applied to input or output channel.
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.
ACQuisition	The acquisition subsystem controls how a real-time oscilloscope is being configured in order to capture the signal for BER measurements.
PLUGin	The PLUGin subsystem is a plugin interface for integrating C# assemblies into the M8070B 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.
M8047B	The M8047B subsystem is used to control M8047B re-driver using M8070B software.

IEEE Commands - Reference

Mandatory Commands

The following mandatory IEEE 488.2 commands are implemented:

Table 19 IEEE commands

Name	Description under
*CLS	*CLS on page 119
*CAL?	*CAL? on page 119
*LRN?	*LRN? on page 119
*ESE[?]	*ESE[?] on page 119
*ESR?	*ESR? on page 119
*IDN?	*IDN? on page 120
*OPC	*OPC on page 120
*OPC?	*OPC? on page 120
*OPT?	*OPT? on page 121
*RCL	*RCL on page 121
*RST	*RST on page 121
*SAV	*SAV on page 121
*SRE[?]	*SRE[?] on page 122
*STB?	*STB? on page 122
*TST?	*TST? on page 122
*WAI	*WAI on page 123

*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 20 IDN field definitions

Field	Value
Manufacturer	Keysight 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-M8045A,G32,G64,0G2,0G3,0G4,0G6,0P3,0P6,0G9,M8070ISIB,0G8,U64,UG2,UG3,UG4,UG6,UP3,UP6,UG9)(M2-M8046A,A32,A64,0A3,0A4,0A5,0P3,0P6,0S1,0S2,0S3,0S4,0S6,0N1,0N2,U64,UA3,UA4,UA5,UP3,UP6,US1,US2,US3,US4,US6,UN1,UN2,0G6)

*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 21 SCPI standard commands

Name	Description under
:SYSTem:ERRor[:NEXT]?	:SYSTem:ERRor[:NEXT]? on page 124
:SYSTem:HELP:HEADers?	:SYSTem:HELP:HEADers? on page 124
:SYSTem:VERSion?	:SYSTem:VERSion? on page 124

: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 22 Miscellaneous commands

Name	Description under
:TEST:RESults?	:TEST:RESults? on page 125
:ISET?	:ISET? on page 125
:VSET?	:VSET? on page 126

:TEST:RESults?

Syntax :TEST:RESults?

Description This query returns the results of self test.

:ISET?

Syntax :ISET?

Description ISET is short for "identifier set".

This query returns a list of locations where this SCPI is usable with.

Example Here is a usage example:

The regular SCPI might be:

-> :DATA:IDLepattern:VALue? 'M1.DataOut1'

<- PRBS

:ISET? is then usable like this:

-> :DATA:IDLepattern:VALue:ISET?

<- "M1.DataOut1,M1.DataOut2"

NOTE Note that the :ISET? sub-mnemonic is not supported by all SCPI definitions.

:VSET?

Syntax :VSET?

.Description VSET is short for "value set".

This query returns the intersection of possible values of enums existing at the individually selected location.

Usage

- M1.DataOut1 has enum Type1 {A, B}
- M2.Clock has enum Type2 {A, C}

:VSET? "M1.DataOut1" will return A, B

:VSET? "M2.Clock" will return A, C

:VSET? ".*" will return A.

If :VSET? across the selected locations does not intersect NVAL (for "no value") will be returned.

Example Here is a usage example:

The regular SCPI might be:

-> :DATA:IDLepattern:VALue? 'M1.DataOut1'

<- PRBS

:VSET? will then be usable like this:

-> :DATA:IDLepattern:VALue:VSET? 'M1.DataOut1'

<- "PRBS,STATIC0"

M8042A Pattern Generator Module Commands

The Keysight M8042A Pattern Generator module supports the following commands:

Table 23 M8042A pattern generator commands

Name	Description under
[:SOURce]:VOLTage[:AMPLitude][?]	[:SOURce]:VOLTage[:AMPLitude][?] on page 188
[:SOURce]:VOLTage:OFFSet[?]	[:SOURce]:VOLTage:OFFSet[?] on page 189
[:SOURce]:VOLTage:HIGH[?]	[:SOURce]:VOLTage:HIGH[?] on page 190
[:SOURce]:VOLTage:LOW[?]	[:SOURce]:VOLTage:LOW[?] on page 190
[:SOURce]:VOLTage:RANGe[:SELect][?]	[:SOURce]:VOLTage:RANGe[:SELect][?] on page 191
[:SOURce]:VOLTage:RANGe:AUTO[?]	[:SOURce]:VOLTage:RANGe:AUTO[?] on page 191
[:SOURce]:JITTer[:GLOBal][:STATe][?]	[:SOURce]:JITTer[:GLOBal][:STATe][?] on page 194
[:SOURce]:JITTer:HFRequency:UNIT[?]	[:SOURce]:JITTer:HFRequency:UNIT[?] on page 197
[:SOURce]:JITTer:CONFigure[:DELay][?]	[:SOURce]:JITTer:CONFigure[:DELay][?] on page 194
[:SOURce]:JITTer:HFRequency:BUNCorrelate[:STATe][?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate[:STATe][?] on page 203
[:SOURce]:JITTer:HFRequency:BUNCorrelate:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:AMPLitude[?] on page 203
[:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect][?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect][?] on page 205
[:SOURce]:JITTer:HFRequency:BUNCorrelate:DRATe[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:DRATe[?] on page 204
[:SOURce]:JITTer:HFRequency:BUNCorrelate:FILTer[:SELect][?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:FILTer[:SELect][?] on page 204
[:SOURce]:JITTer:HFRequency:RANDom[:STATe]	[:SOURce]:JITTer:HFRequency:RANDom[:STATe][?] on page 207
[:SOURce]:JITTer:HFRequency:RANDom:AMPLitude	[:SOURce]:JITTer:HFRequency:RANDom:AMPLitude[?] on page 207
[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs][?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs][?] on page 208

Name	Description under
[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs]:VALue[?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs]:VALue[?] on page 208
[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs[?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs[?] on page 209
[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe][?]	[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe][?] on page 211
[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*)[?]	[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*)[?] on page 211
[:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs][:STATe][?]	[:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs][:STATe][?] on page 212
[:SOURce]:JITTer:LFRequency:UNIT[?]	[:SOURce]:JITTer:LFRequency:UNIT[?] on page 214
[:SOURce]:JITTer:LFRequency:PERiodic[:STATe][?]	[:SOURce]:JITTer:LFRequency:PERiodic1[:STATe][?] on page 216
[:SOURce]:JITTer:LFRequency:PERiodic:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:PERiodic1:AMPLitude[?] on page 216
[:SOURce]:JITTer:LFRequency:PERiodic:FREQuency[?]	[:SOURce]:JITTer:LFRequency:PERiodic1:FREQuency[?] on page 217
[:SOURce]:JITTer:LFRequency:RSSClocking[:STATe][?]	[:SOURce]:JITTer:LFRequency:RSSClocking[:STATe][?] on page 219
[:SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude[?] on page 220
[:SOURce]:JITTer:LFRequency:RSSClocking:FREQuency[?]	[:SOURce]:JITTer:LFRequency:RSSClocking:FREQuency[?] on page 220
:OUTPut[:STATe][?]	:OUTPut[:STATe][?] on page 281
:OUTPut:CONFig:SOURce	:OUTPut:CONFig:SOURce[?] on page 282
:OUTPut:DRATe?	:OUTPut:DRATe? on page 283
:OUTPut:SRATe?	:OUTPut:DRATe? on page 283
:OUTPut:POLarity[?]	:OUTPut:POLarity[?] on page 283
:OUTPut:TVOLtage[?]	:OUTPut:TVOLtage[?] on page 283
:OUTPut:TCONfig[?]	:OUTPut:TCONfig[?] on page 284
:OUTput:COUPling[?]	:OUTPut:COUPling[?] on page 285
:OUTPut:FREQuency[?]	:OUTPut:FREQuency[?] on page 286

Name	Description under
:OUTPut:DATA:F2Jitter:STATe[?]	:OUTPut:DATA:F2Jitter:STATe[?] on page 289
:OUTPut:DATA:F2Jitter[?]	:OUTPut:DATA:F2Jitter[?] on page 289
:OUTPut:DATA:F2Jitter:UNIT[?]	
:OUTPut:CALibration:SKEW[?]	:OUTPut:CALibration:SKEW[?] on page 287
:OUTPut:DEEMphasis:UNIT[?]	:OUTPut:DEEMphasis:UNIT[?] on page 293
:OUTPut:DEEMphasis:PRESet[?]	:OUTPut:DEEMphasis:PRESet[?] on page 293
:OUTPut:DEEMphasis:PRESet:FILE[?]	:OUTPut:DEEMphasis:PRESet:FILE[?] on page 294
:OUTPut:DEEMphasis:PRESet:ENABle[?]	:OUTPut:DEEMphasis:PRESet:ENABle[?] on page 294
:OUTPut:DEEMphasis:SELect:PRESet[?]	:OUTPut:DEEMphasis:SELect:PRESet[?] on page 295
:OUTPut:DEEMphasis:POSTcursor(*)[?]	:OUTPut:DEEMphasis:POSTcursor(*)[?] on page 296
:OUTPut:DEEMphasis:CURSor:MAGNitude(0|1|2|3|4|5|6)[?]	:OUTPut:DEEMphasis:CURSor:MAGNitude(0|1|2|3|4|5|6|7)[?] on page 298
:OUTPut:DEEMphasis:CURSor:MAIN:AUTO[?]	:OUTPut:DEEMphasis:CURSor:MAIN:AUTO[?] on page 298
:OUTPut:DEEMphasis:CURSor:MAIN:SUFFix[?]	:OUTPut:DEEMphasis:CURSor:MAIN:SUFFix[?] on page 299
:OUTPut:DEEMphasis:CURSor:NPRecursor[?]	:OUTPut:DEEMphasis:CURSor:NPRecursor[?] on page 299
:OUTPut:DEEMphasis:CURSor:NTAPs[?]	:OUTPut:DEEMphasis:CURSor:NTAPs[?] on page 300
:OUTPut:EINSertion[:STATe]	:OUTPut:EINSertion[:STATe][?] on page 306
:OUTPut:EINSertion:RATio	:OUTPut:EINSertion:RATio[?] on page 306
:OUTPut:EINSertion:MODE	:OUTPut:EINSertion:MODE[?] on page 307
:OUTPut:EINSertion:ONCE	:OUTPut:EINSertion:ONCE on page 307
:OUTPut:TRIGger:MODE[?]	:OUTPut:TRIGger:MODE[?] on page 310
:OUTPut:TRIGger:DIVider[?]	:OUTPut:TRIGger:DIVider[?] on page 311
:INPut[:VOLTage][:NRZ]:THReshold[?]	:INPut[:VOLTage][:NRZ]:THReshold[?] on page 335
:INPut:TVOLtage	:INPut:TVOLtage[?] on page 331
:DATA:SEQuence:BREak	:DATA:SEQuence:BREak on page 557
:DATA:SEQuence:BLOCk:IDENtifier?	:DATA:SEQuence:BLOCk:IDENtifier? on page 559
:DATA:SEQuence:GRANularity:IDENtifier?	:DATA:SEQuence:GRANularity:IDENtifier? on page 560

Name	Description under
:DATA:SEQuence:RESTart:IDENtifier	:DATA:SEQuence:RESTart:IDENtifier on page 558
:DATA:LINecoding[:VALue][?]	:DATA:LINecoding[:VALue][?] on page 565
:DATA:LINecoding:PAM3:PRECoding:VALue[?]	:DATA:LINecoding:PAM3:PRECoding:VALue[?] on page 566
:DATA:LINecoding:PAM3[:SYMBol]:LEVel(1)[?]	:DATA:LINecoding:PAM3[:SYMBol]:LEVel(1)[?] on page 567
:DATA:LINecoding:PAM3:MAPPing[?]	:DATA:LINecoding:PAM3:MAPPing[?] on page 567
:DATA:LINecoding:PAM3:MAPPing:CUSTom[?]	:DATA:LINecoding:PAM3:MAPPing:CUSTom[?] on page 568
:DATA:LINecoding:PAM4:MAPPing[?]	:DATA:LINecoding:PAM4:MAPPing[?] on page 569
:DATA:LINecoding:PAM4:PRECoding[?]	:DATA:LINecoding:PAM4:PRECoding[?] on page 571
:DATA:LINecoding:PAM4[:SYMBol]:LEVel(0|3)?	:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2)[?] on page 571
:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2)[?]	:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2)[?] on page 571
:DATA:LINecoding:PAM6:MAPPing[?]	:DATA:LINecoding:PAM6:MAPPing[?] on page 572
:DATA:LINecoding:PAM6[:SYMBol]:LEVel(1|2|3|4)[?]	:DATA:LINecoding:PAM6[:SYMBol]:LEVel(1|2|3|4)[?] on page 575
:DATA:LINecoding:PAM8:MAPPing[?]	:DATA:LINecoding:PAM8:MAPPing[?] on page 574
:DATA:LINecoding:PAM8[:SYMBol]:LEVel(1|2|3|4|5|6)[?]	:DATA:LINecoding:PAM8[:SYMBol]:LEVel(1|2|3|4|5|6)[?] on page 576
:TRIGger[:SOURce]:DELay[?]	:TRIGger[:SOURce][?] on page 168
:TRIGger[:SOURce]:FREQuency[?]	:TRIGger[:SOURce]:FREQuency? on page 169

M8043A Error Analyzer Module Commands

The Keysight M8043A Error Analyzer module supports the following commands:

Table 24 M8043A error analyzer commands

Name	Description under
:CLOCk[:SOURce][?]	:CLOCk[:SOURce][?] on page 408
:CLOCk:TRACk[:STATe][?]	:CLOCk:TRACk[:STATe][?] on page 410
:FETCh:BCOunter?	:FETCh:BCOunter? on page 421
:FETCh:BCOunter:DETailed?	:FETCh:BCOunter:DETailed? on page 424
:FETCh:IBERate?	:FETCh:IBERate? on page 425
:FETCh:HISTogram:BIN[:DATA]?	:FETCh:HISTogram:BIN[:DATA]? on page 426
:FETCh:EDIagram:SINGle	:FETCh:EDIagram:SINGle on page 427
:FETCh:EDIagram:COLumns?	:FETCh:EDIagram:COLumns? on page 428
:FETCh:EDIagram:[DATA]?	:FETCh:EDIagram[:DATA]? on page 428
:FETCh:EDIagram:ROWS?	:FETCh:EDIagram:ROWS? on page 429
:FETCh:EDIagram:XINCrement?	:FETCh:EDIagram:XINCrement? on page 430
:FETCh:EDIagram:XORigin?	:FETCh:EDIagram:XORigin? on page 430
:FETCh:EDIagram:YINCrement?	:FETCh:EDIagram:YINCrement? on page 430
:FETCh:EDIagram:YORigin?	:FETCh:EDIagram:YORigin? on page 431
:INPut:EQUalization:STATe[?]	:INPut:EQUalization:STATe[?] on page 378
:INPut:EQUalization[:PRESet]:TYPE[?]	:INPut:EQUalization[:PRESet]:TYPE[?] on page 374
:INPut:EQUalization[:PRESet]:GAIN[?]	:INPut:EQUalization[:PRESet]:GAIN[?] on page 374
:INPut:EQUalization:CURSor:DETect	:INPut:EQUalization:CURSor:DETect on page 372
:INPut:EQUalization:CURSor:MAGNitude[0|1|2|3|...|15][?]	:INPut:EQUalization:CURSor:MAGNitude[0|1|2|3|...|15][?] on page 371
:INPut:COUPling[?]	:INPut:COUPling[?] on page 332
:INPut:TCONfig[?]	:INPut:TCONfig[?] on page 331
:INPut:CMODe[?]	:INPut:CMODe[?] on page 331

Name	Description under
:INPut:TVOLtage[?]	:INPut:TVOLtage[?] on page 331
:INPut[:VOLTage][:NRZ]:THReshold[?]	:INPut[:VOLTage][:NRZ]:THReshold[?] on page 335
:INPut[:VOLTage]:PAM4[:SYMBol]:THReshold(1|2|3)[?]	:INPut[:VOLTage]:PAM4[:SYMBol]:THReshold(1|2|3)[?] on page 336
:INPut:POLarity[?]	:INPut:POLarity[?] on page 329
:INPut[:VOLTage]:WINDow[?]	:INPut[:VOLTage]:WINDow[?] on page 333
:INPut[:VOLTage]:WINDow:DETect	:INPut[:VOLTage]:WINDow:DETect on page 334
:INPut[:VOLTage]:WINDow:HISTogram?	:INPut[:VOLTage]:WINDow:HISTogram? on page 335
:INPut:CMVoltage[?]	:INPut:CMVoltage[?] on page 329
:INPut:CMVoltage:DETect	:INPut:CMVoltage:DETect on page 330
:INPut:CDR:CTRL[?]	:INPut:CDR[:JTF]:CTRL[?] on page 347
:INPut:CDR[:JTF]:AUTO[?]	:INPut:CDR[:JTF]:AUTO[?] on page 348
:INPut:CDR[:JTF]:RELOck	:INPut:CDR[:JTF]:RELOck on page 349
:INPut:CDR[:JTF]:FIRSt:TDENsity[?]	:INPut:CDR[:JTF]:FIRSt:TDENsity[?] on page 350
:INPut:CDR[:JTF]:FIRSt:LBANdwidth[?]	:INPut:CDR[:JTF]:FIRSt:LBANdwidth[?] on page 350
:INPut:DRATe?	:INPut:DRATe? on page 328
:INPut:DELay[?]	:INPut:DELay[?] on page 328
:INPut:ALIGnment:STATus[:VALue]?	:INPut:ALIGnment:STATus[:VALue]? on page 366
:INPut:ALIGnment:EYE:ABORt	:INPut:ALIGnment:EYE:ABORt on page 358
:INPut:ALIGnment:EYE[:AUTO]	:INPut:ALIGnment:EYE[:AUTO] on page 357
:INPut:ALIGnment:EYE:TCENter	:INPut:ALIGnment:EYE:TCENter on page 358
:INPut:ALIGnment:EYE:ACENter	:INPut:ALIGnment:EYE:ACENter on page 358
:INPut:ALIGnment:EYE:THReshold[?]	:INPut:ALIGnment:EYE:THReshold[?] on page 359
:INPut:ALIGnment:EYE:RESult:WIDTh?	:INPut:ALIGnment:EYE:RESult:WIDTh? on page 359
:INPut:ALIGnment:EYE:RESult:DELay?	:INPut:ALIGnment:EYE:RESult:DELay? on page 363
:INPut:ALIGnment:EYE:RESult[:NRZ]:THReshold?	:INPut:ALIGnment:EYE:RESult[:NRZ]:THReshold? on page 360

Name	Description under
:INPut:ALIGnment:EYE:RESult:PAM4:HEIGht(1|2|3)?	:INPut:ALIGnment:EYE:RESult:PAM4:HEIGht(1|2|3)? on page 361
:INPut:ALIGnment:EYE:RESult:PAM4:THReshold(1|2|3)?	:INPut:ALIGnment:EYE:RESult:PAM4:THReshold(1|2|3)? on page 361
:INPut:ALIGnment:EYE:RESult:POLarity?	:INPut:ALIGnment:EYE:RESult:POLarity? on page 364
:INPut:ALIGnment:EYE:RESult[:NRZ][:HEIGht]?	:INPut:ALIGnment:EYE:RESult[:NRZ][:HEIGht]? on page 359
:INPut:DEMBedding:ISYMbol:CLOSs[?]	:INPut:DEMBedding:ISYMbol:CLOSs[?] on page 398
:INPut:DEMBedding:ISYMbol:EXTernal[?]	:INPut:DEMBedding:ISYMbol:EXTernal[?] on page 398
:INPut:DEMBedding:ISYMbol:FREQuency[?]	:INPut:DEMBedding:ISYMbol:FREQuency[?] on page 399
:INPut:DEMBedding:ISYMbol:STATe[?]	:INPut:DEMBedding:ISYMbol:STATe[?] on page 399
:INPut:DEMBedding[1|2]:FILE[?]	:INPut:DEMBedding[1|2]:FILE[?] on page 400
:INPut:DEMBedding[1|2]:OPERator[?]	:INPut:DEMBedding[1|2]:OPERator[?] on page 400
:INPut:DEMBedding[1|2]:PORT[?]	:INPut:DEMBedding[1|2]:PORT[?] on page 401
:INPut:DEMBedding[1|2]1:STATe[?]	:INPut:DEMBedding[1|2]1:STATe[?] on page 401
:INPut:DEMBedding[1|2]1:WEIGht[?]	:INPut:DEMBedding[1|2]1:WEIGht[?] on page 402
:INPut:TVOLtage[?]	:INPut:TVOLtage[?] on page 331
:INPut[:VOLTage][:NRZ]:THReshold[?]	:INPut[:VOLTage][:NRZ]:THReshold[?] on page 335
:DATA:LINecoding[:VALue][?]	:DATA:LINecoding[:VALue][?] on page 565
:DATA:LINecoding:PAM4:MAPPing[?]	:DATA:LINecoding:PAM4:MAPPing[?] on page 569
:DATA:LINecoding:PAM4:MAPPing:CUSTom[?]	:DATA:LINecoding:PAM4:MAPPing:CUSTom[?] on page 570
:DATA:SYNC:TYPE[?]	:DATA:SYNC:TYPE[?] on page 562
:DATA:SYNC:THReshold[?]	:DATA:SYNC:THReshold[?] on page 562
:DATA:SYNC[:ONCE]	:DATA:SYNC[:ONCE] on page 561
:DATA:COMPare:EVENt:ERRor:MODE[?]	:DATA:COMPare:EVENt:ERRor:MODE[?] on page 584
:DATA:PATTern:BLENgth?	:DATA:PATTern:BLENgth? on page 546
:DATA:PATTern:CATalog?	:DATA:PATTern:CATalog? on page 546
:DATA:PATTern:CINFormation?	:DATA:PATTern:CINFormation? on page 546
:DATA:PATTern:DELete	:DATA:PATTern:DELete on page 547

Name	Description under
:DATA:PATTern:DESCription[?]	:DATA:PATTern:DESCription[?] on page 547
:DATA:PATTern:EXPort	:DATA:PATTern:EXPort on page 549
:DATA:PATTern:IDATa[?]	:DATA:PATTern:IDATa[?] on page 545
:DATA:PATTern:IMPort	:DATA:PATTern:IMPort on page 548
:DATA:PATTern:PACKed[?]	:DATA:PATTern:PACKed[?] on page 547
:DATA:PATTern:USE[?]	:DATA:PATTern:USE[?] on page 544
:DATA:SEQuence:BIND[?]	:DATA:SEQuence:BIND[?] on page 556
:DATA:SEQuence:BLOCk?	:DATA:SEQuence:BLOCk? on page 559
:DATA:SEQuence:BLOCk:IDENtifier?	:DATA:SEQuence:BLOCk:IDENtifier? on page 559
:DATA:SEQuence:DALL	:DATA:SEQuence:DALL on page 557
:DATA:SEQuence:DELete	:DATA:SEQuence:DELete on page 557
:DATA:SEQuence:GRANularity?	:DATA:SEQuence:GRANularity? on page 559
:DATA:SEQuence:GRANularity:IDENtifier?	:DATA:SEQuence:GRANularity:IDENtifier? on page 560
:DATA:SEQuence:NAMes?	:DATA:SEQuence:NAMes? on page 556
:DATA:SEQuence:NEW	:DATA:SEQuence:NEW on page 556
:DATA:SEQuence:RESTart	:DATA:SEQuence:RESTart on page 558
:DATA:SEQuence:RESTart:IDENtifier	:DATA:SEQuence:RESTart:IDENtifier on page 558
:DATA:SEQuence[:SET][?]	:DATA:SEQuence[:SET][?] on page 553
:DATA:SEQuence[:SET]:VALue[?]	:DATA:SEQuence[:SET]:VALue[?] on page 555
:OUTPut[:STATe][?]	:OUTPut[:STATe][?] on page 281
:OUTPut:POLarity[?]	:OUTPut:POLarity[?] on page 283
:SENSe:SRATe[?]	:SENSe:SRATe[?] on page 405
[:SOURce]:VOLTage[:AMPLitude][?]	[:SOURce]:VOLTage[:AMPLitude][?] on page 188
[:SOURce]:VOLTage:OFFSet[?]	[:SOURce]:VOLTage:OFFSet[?] on page 189
[:SOURce]:VOLTage:HIGH[?]	[:SOURce]:VOLTage:HIGH[?] on page 190
[SOURce]:VOLTage:LOW[?]	[:SOURce]:VOLTage:LOW[?] on page 190

M8009A Clock Module Commands

The Keysight M8009A clock module supports the following commands:

Table 25 M8009A clock module commands

Name	Description under
ClkGen	
:TRIGger[:SOURce][?]	:TRIGger[:SOURce][?] on page 168
:TRIGger:INTernal[:SOURce][?]	:TRIGger:INTernal[:SOURce][?] on page 169
[:SOURce]:FREQuency[?]	[:SOURce]:FREQuency[?] on page 177
[:SOURce]:PERiod[?]	[:SOURce]:PERiod[?] on page 179
:TRIGger:REFerence[:FREQuency][?]	:TRIGger:REFerence[:FREQuency][?] on page 170
:TRIGger:DIRect:FREQuency[?]	:TRIGger:DIRect[:FREQuency][?] on page 170
:TRIGger:CMULtiplier:FREQuency[?]	:TRIGger:CMULtiplier:FREQuency[?] on page 172
:TRIGger:CMULtiplier:LBWidth[?]	:TRIGger:CMULtiplier:LBWidth[?] on page 171
:TRIGger:CMULtiplier:FREQuency:MULTiplier[?]	:TRIGger:CMULtiplier:FREQuency:MULTiplier[?] on page 171
:TRIGger:CMULtiplier:FREQuency:DIVider[?]	:TRIGger:CMULtiplier:FREQuency:DIVider[?] on page 172
ClkOut16G	
:OUTPut:STATe[?]	:OUTPut[:STATe][?] on page 281
:OUTPut:CLock:DIVider[?]	:OUTPut:CLock:DIVider[?] on page 286
:SOURce:VOLTage:TERMination[?]	[:SOURce]:VOLTage:TERMination[?] on page 192
:SOURce:VOLTage:AMPLitude[?]	[:SOURce]:VOLTage[:AMPLitude][?] on page 188
:SOURce:VOLTage:OFFSet[?]	[:SOURce]:VOLTage:OFFSet[?] on page 189
:SOURce:VOLTage:HIGH[?]	[:SOURce]:VOLTage:HIGH[?] on page 190
:SOURce:VOLTage:LOW[?]	[:SOURce]:VOLTage:LOW[?] on page 190
:SOURce:JITTer:LFRequency:UNIT[?]	[:SOURce]:JITTer:LFRequency:UNIT[?] on page 214
:SOURce:JITTer:LFRequency:PERiodic1:STATe[?]	[:SOURce]:JITTer:LFRequency:PERiodic1[:STATe][?] on page 216
:SOURce:JITTer:LFRequency:PERiodic1:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:PERiodic1:AMPLitude[?] on page 216

Name	Description under
:SOURce:JITTer:LFRequency:PERiodic1:FREQuency[?]	[:SOURce]:JITTer:LFRequency:PERiodic1:FREQuency[?] on page 217
:SOURce:JITTer:LFRequency:PERiodic2:STATe[?]	[:SOURce]:JITTer:LFRequency:PERiodic2[:STATe][?] on page 217
:SOURce:JITTer:LFRequency:PERiodic2:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:PERiodic2:AMPLitude[?] on page 218
:SOURce:JITTer:LFRequency:PERiodic2:FREQuency[?]	[:SOURce]:JITTer:LFRequency:PERiodic2:FREQuency[?] on page 218
[:SOURce]:JITTer:LFRequency:RSSClocking[:STATe][?]	[:SOURce]:JITTer:LFRequency:RSSClocking[:STATe][?] on page 219
[:SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude[?] on page 220
[:SOURce]:JITTer:LFRequency:RSSClocking:FREQuency[?]	[:SOURce]:JITTer:LFRequency:RSSClocking:FREQuency[?] on page 220
:SOURce:JITTer:HFRequency:PERiodic(*):STATe	[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?] on page 199
:SOURce:JITTer:HFRequency:BUNCorrelate:STATe	[:SOURce]:JITTer:HFRequency:BUNCorrelate[:STATe][?] on page 203
:SOURce:JITTer:HFRequency:RANDom:STATe	[:SOURce]:JITTer:HFRequency:RANDom[:STATe][?] on page 207
:SOURce:JITTer:HFRequency:SPECtrally:STATe	[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe][?] on page 211
ClkOut32G	
:OUTPut:STATe[?]	:OUTPut[:STATe][?] on page 281
:SOURce:VOLTage:AMPLitude	[:SOURce]:VOLTage[:AMPLitude][?] on page 188
:SOURce:PULSe:DELay	[:SOURce]:PULSe:DELay[?] on page 184
:SOURce:JITTer:CONFigure:DELay	[:SOURce]:JITTer:CONFigure[:DELay][?] on page 194
:OUTPut:CALibration:SKEW	:OUTPut:CALibration:SKEW[?] on page 287
:OUTPut:DIVider	:OUTPut:DIVider[?] on page 285
:OUTPut:FREQuency	:OUTPut:FREQuency[?] on page 286
:OUTPut:JITTer	:OUTPut:JITTer [?] on page 287

Name	Description under
Ref ClkOut	
:OUTPut:FREQuency[?]	:OUTPut:FREQuency[?] on page 286
Ref ClkOut 16G	
:OUTPut:STATe[?]	:OUTPut[:STATe][?] on page 281
:OUTPut:FREQuency?	:OUTPut:FREQuency[?] on page 286
Channel ClkOut (ChClkOut1, ChClkOut2)	
:OUTPut:FREQuency[?]	:OUTPut:FREQuency[?] on page 286
ChClkOut2	
:OUTPut:STATe[?]	:OUTPut[:STATe][?] on page 281
[:SOURce]:POWer[?]	[:SOURce]:POWer[?] on page 179
:OUTPut:DIVider[?]	:OUTPut:DIVider[?] on page 285
:SOURce:PULSe:DELay[?]	[:SOURce]:PULSe:DELay[?] on page 184
:SOURce:JITTer:CONFigure:DELay[?]	[:SOURce]:JITTer:CONFigure[:DELay][?] on page 194
:OUTPut:CALibration:SKEW[?]	:OUTPut:CALibration:SKEW[?] on page 287
:SOURce:JITTer:LFRequency:UNIT[?]	[:SOURce]:JITTer:LFRequency:UNIT[?] on page 214
:SOURce:JITTer:LFRequency:PERiodic1:STATe[?]	[:SOURce]:JITTer:LFRequency:PERiodic1[:STATe][?] on page 216
:SOURce:JITTer:LFRequency:PERiodic1:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:PERiodic1:AMPLitude[?] on page 216
:SOURce:JITTer:LFRequency:PERiodic1:FREQuency[?]	[:SOURce]:JITTer:LFRequency:PERiodic1:FREQuency[?] on page 217
:SOURce:JITTer:LFRequency:PERiodic2:STATe[?]	[:SOURce]:JITTer:LFRequency:PERiodic2[:STATe][?] on page 217
:SOURce:JITTer:LFRequency:PERiodic2:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:PERiodic2:AMPLitude[?] on page 218
:SOURce:JITTer:LFRequency:PERiodic2:FREQuency[?]	[:SOURce]:JITTer:LFRequency:PERiodic2:FREQuency[?] on page 218
[:SOURce]:JITTer:LFRequency:RSSClocking[:STATe][?]	[:SOURce]:JITTer:LFRequency:RSSClocking[:STATe][?] on page 219
[:SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude[?] on page 220

Name	Description under
[:SOURce]:JITTer:LFRequency:RSSClocking:FREQuency[?]	[:SOURce]:JITTer:LFRequency:RSSClocking:FREQuency[?] on page 220
:SOURce:JITTer:HFRequency:UNIT[?]	[:SOURce]:JITTer:HFRequency:UNIT[?] on page 197
:SOURce:JITTer:HFRequency:PERiodic1:STATe[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?] on page 199
:SOURce:JITTer:HFRequency:PERiodic1:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude[?] on page 200
:SOURce:JITTer:HFRequency:PERiodic1:FREQuency[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*):FREQuency[?] on page 200
:SOURce:JITTer:HFRequency:PERiodic2:STATe[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?] on page 199
:SOURce:JITTer:HFRequency:PERiodic2:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude[?] on page 200
:SOURce:JITTer:HFRequency:PERiodic2:FREQuency[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*):FREQuency[?] on page 200
:SOURce:JITTer:HFRequency:BUNCorrelate:STATe[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate[:STATe][?] on page 203
:SOURce:JITTer:HFRequency:BUNCorrelate:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:AMPLitude[?] on page 203
:SOURce:JITTer:HFRequency:BUNCorrelate:PRBSequence[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect][?] on page 205
:SOURce:JITTer:HFRequency:BUNCorrelate:DRATe[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:DRATe[?] on page 204
:SOURce:JITTer:HFRequency:BUNCorrelate:FILTer:SELect[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:FILTer[:SELect][?] on page 204
:SOURce:JITTer:HFRequency:RANDom:STATe[?]	[:SOURce]:JITTer:HFRequency:RANDom[:STATe][?] on page 207
:SOURce:JITTer:HFRequency:RANDom:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:RANDom:AMPLitude[?] on page 207
:SOURce:JITTer:HFRequency:RANDom:FILTer:LPASs[?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs][?] on page 208
:SOURce:JITTer:HFRequency:RANDom:FILTer:HPASs[?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs[?] on page 209
:SOURce:JITTer:HFRequency:SPECtrally:STATe[?]	[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe][?] on page 211

Name	Description under
:SOURce:JITTer:HFRequency:SPECtrally:AMPLitude1|AMPLitude[?]	[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*)[?] on page 211
:SOURce:JITTer:HFRequency:SPECtrally:AMPLitude2[?]	[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*)[?] on page 211
:SOURce:JITTer:HFRequency:SPECtrally:FILTer:LPASs:STATe[?]	[:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs][:STATe][?] on page 212
:OUTPut:MODE[?]	:OUTPut:MODE[?] on page 282
SSC	
:SOURce:SSCLocking:STATe?	[:SOURce]:SSCLocking[:STATe][?] on page 228
:SOURce:SSCLocking:DEViation:VALue[?]	[:SOURce]:SSCLocking:DEViation[:VALue][?] on page 229
:SOURce:SSCLocking:FREQuency[?]	[:SOURce]:SSCLocking:FREQuency[?] on page 232
:SOURce:SSCLocking:TYPe[?]	[:SOURce]:SSCLocking:TYPE[?] on page 230
:SOURce:SSCLocking:PROFile[?]	[:SOURce]:SSCLocking:PROFile[?] on page 231
:SOURce:SSCLocking:SHAPe[?]	[:SOURce]:SSCLocking:SHAPe[?] on page 232

Segmented SSC Subnode	
[:SOURce]:SSCLocking:SEGMented:ADVance	[:SOURce]:SSCLocking:SEGMented:ADVance on page 268
[:SOURce]:SSCLocking:SEGMented:CUSTom:FILE1|2|3|4[?]	[:SOURce]:SSCLocking:SEGMented:CUSTom:FILE1|2|3|4[?] on page 268
[:SOURce]:SSCLocking:SEGMented:CUSTom:ONCE1|2|3|4[?]	[:SOURce]:SSCLocking:SEGMented:CUSTom:ONCE1|2|3|4[?] on page 269
[:SOURce]:SSCLocking:SEGMented:CUSTom:SEGMents[?]	[:SOURce]:SSCLocking:SEGMented:CUSTom:SEGMents[?] on page 269
[:SOURce]:SSCLocking:SEGMented:DELTa1|2[?]	[:SOURce]:SSCLocking:SEGMented:DELTa1|2[?] on page 270
[:SOURce]:SSCLocking:SEGMented:DEViation[?]	[:SOURce]:SSCLocking:SEGMented:DEViation[?] on page 270
[:SOURce]:SSCLocking:SEGMented:INTDeviation[?]	[:SOURce]:SSCLocking:SEGMented:INTDeviation[?] on page 271
[:SOURce]:SSCLocking:SEGMented:MINDeviation[?]	[:SOURce]:SSCLocking:SEGMented:MINDeviation[?] on page 271
[:SOURce]:SSCLocking:SEGMented:LOOP:STATe[?]	[:SOURce]:SSCLocking:SEGMented:LOOP:STATe[?] on page 272

[:SOURce]:SSCLocking:SEGMented:OVERshoot[?]	[:SOURce]:SSCLocking:SEGMented:OVERshoot[?] on page 272
[:SOURce]:SSCLocking:SEGMented:RESTart	[:SOURce]:SSCLocking:SEGMented:RESTart on page 273
[:SOURce]:SSCLocking:SEGMented:SEGMent?	[:SOURce]:SSCLocking:SEGMented:SEGMent? on page 273
[:SOURce]:SSCLocking:SEGMented:SHAPe[?]	[:SOURce]:SSCLocking:SEGMented:SHAPe[?] on page 273
[:SOURce]:SSCLocking:SEGMented:SSCFrequency[?]	[:SOURce]:SSCLocking:SEGMented:SSCFrequency[?] on page 274
[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:ACTTime[?]	[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:ACTTime[?] on page 274
[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:STATe[?]	[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:STATe[?] on page 275
[:SOURce]:SSCLocking:SEGMented:TP[?]	[:SOURce]:SSCLocking:SEGMented:TP[?] on page 275
[:SOURce]:SSCLocking:SEGMented:TRIGger:SEGMent1|2|3|4:STATe[?]	[:SOURce]:SSCLocking:SEGMented:TRIGger:SEGMent1|2|3|4:STATe[?] on page 276
[:SOURce]:SSCLocking:SEGMented:TRIGger:STATe[?]	[:SOURce]:SSCLocking:SEGMented:TRIGger:STATe[?] on page 276

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

Table 26 lists locations and their respective modules.

Table 26 Location wise mapping of modules

Location	Module
Data Out	M8041A, M8042A, M8051A, M8062A, M8195A and M8045A
Data In	M8041A, M8051A, M8062A, M8043A and M8046A
System	M8041A, M8042A
Clk Gen	M8009A, M8041A, M8062A, M8195A and M8045A
Ref Clk Out	M8009A, M8041A and M8045A
Ref Clk Out 16G	M8009A
Clk Out	M8041A and M8045A
Clk Out 16G	M8009A
Clk Out 32G	M8009A
Cln Clk Out	M8062A
Trig Out	M8041A, M8042A and M8045A
Sys Out A	M8041A and M8045A
Sys Out B	M8041A and M8045A
Ctrl Out A	M8041A, M8042A, M8051A, M8045A, M8043A and M8046A
Ctrl Out B	M8042A
Sys In A	M8041A and M8045A
Sys In B	M8041A and M8045A
Ctrl In A	M8041A, M8042A, M8051A, M8045A, M8043A and M8046A
Ctrl In B	M8041A, M8042A and M8051A
Elect Idle In	M8062A
Clk In	M8043A, M8046A
Ch Clk Out 1	M8009A
Ch Clk Out 2	M8009A

NOTE The “System” location is only available for few SCPIs. It is not visible on the M8070B user interface.

STATus Subsystem

This subsystem has the following SCPI structure:

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

This subsystem has the following commands and subnodes:

Table 27

Name	Description under
Subnodes	
:QUEStionable	:STATus:QUEStionable Subnode on page 146
:OPERation	:STATus:OPERation Subnode on page 154
INSTrument	:STATus:INSTrument Subnode on page 159

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 28

License	Description	Modules
M8009A-0G6	Reference Clock Multiplier	M8009A
M8041A-0G6	Reference Clock Input with Multiplying PLL, Clockgroup-wide License	M8041A
M8041A-C08	BERT one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8042A-G32	Pattern generation up to 32 GBd for NRZ and PAM4	M8042A
M8043A-A32	Error analysis up to 32.4 GBd for NRZ and PAM4	M8043A
M8045A-G32	32 GBd Pattern Generator	M8045A
M8046A-0A4	Clock Recovery for 32 GBd	M8046A
M8046A-A32	32GBd Error Detector	M8045A
M8062A-0A4	Clock Recovery up to 32 Gb/s	M8062A
M8062A-C32	32 Gb/s BERT front-end	M8062A
M8070ADVB	Advanced Measurement Package for M8000 Series BERT Test Solutions	M8041A, M8045A, M8009A

: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 29

Name	Description under
:CONDition?	:STATus:QUEStionable:CONDition? on page 148
:ENABle[?]	:STATus:QUEStionable:ENABle[?] on page 148
[:EVENt]?	:STATus:QUEStionable[:EVENt]? on page 148
:NTRansition[?]	:STATus:QUEStionable:NTRansition[?] on page 148
:PTRansition[?]	:STATus:QUEStionable:PTRansition[?] on page 149
:INSTrument[?]	:STATus:QUEStionable:INSTrument on page 149
:INSTrument:CONDition?	:STATus:QUEStionable:INSTrument:CONDition? on page 150
:INSTrument:ENABle[?]	:STATus:QUEStionable:INSTrument:ENABle[?] on page 150
:INSTrument[:EVENt]?	:STATus:QUEStionable:INSTrument[:EVENt]? on page 150
:INSTrument:NTRansition[?]	:STATus:QUEStionable:INSTrument:NTRansition[?] on page 150
:INSTrument:PTRansition[?]	:STATus:QUEStionable:INSTrument:PTRansition[?] on page 151
:PROTection	:STATus:QUEStionable:PROTection on page 151
:PROTection:CONDition?	:STATus:QUEStionable:PROTection:CONDition? on page 151
:PROTection:ENABle[?]	:STATus:QUEStionable:PROTection:ENABle[?] on page 151
:PROTection[:EVENt]?	:STATus:QUEStionable:PROTection[:EVENt]? on page 152
:PROTection:NTRansition[?]	:STATus:QUEStionable:PROTection:NTRansition[?] on page 152
:PROTection:PTRansition[?]	:STATus:QUEStionable:PROTection:PTRansition[?] on page 152
:SYMBol	:STATus:QUEStionable:SYMBol on page 152
:SYMBol:CONDition?	:STATus:QUEStionable:SYMBol:CONDition? on page 153
:SYMBol:ENABle[?]	:STATus:QUEStionable:SYMBol:ENABle[?] on page 153
:SYMBol[:EVENt]?	:STATus:QUEStionable:SYMBol[:EVENt]? on page 153
:SYMBol:NTRansition[?]	:STATus:QUEStionable:SYMBol:NTRansition[?] on page 153
:SYMBol:PTRansition[?]	:STATus:QUEStionable:SYMBol:PTRansition[?] on page 154

: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>

:STATus:QUEStionable:INSTrument: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:INSTrument:PTRansition[?]

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

:STATus:QUEStionable:INSTrument: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: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>

:STATus:QUEStionable:SYMBol: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: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 30

Name	Description under
:CONDition?	:STATus:OPERation:CONDition? on page 155
:ENABle[?]	:STATus:OPERation:ENABle[?] on page 155
[:EVENt]?	:STATus:OPERation[:EVENt]? on page 155

Name	Description under
:NTRansition[?]	:STATus:OPERation:NTRansition[?] on page 156
:PTRansition[?]	:STATus:OPERation:PTRansition[?] on page 156
:RUN	:STATus:OPERation:RUN on page 156
:RUN:CONDition?	:STATus:OPERation:RUN:CONDition? on page 157
:RUN:ENABle[?]	:STATus:OPERation:RUN:ENABle[?] on page 157
:RUN[:EVENt]?	:STATus:OPERation:RUN[:EVENt]? on page 157
:RUN:NTRansition[?]	:STATus:OPERation:RUN:NTRansition[?] on page 157
:RUN:PTRansition[?]	:STATus:OPERation:RUN:PTRansition[?] on page 158

: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 Operation 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 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: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 Operation 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?
  │ └:WAIT?
  └:CDR
    └:ULOCKed?
```

This subnode has the following commands:

Table 31

Name	Description under
:CLOSs?	:STATus:INSTrument:CLOSs? on page 160
:DLOSs?	:STATus:INSTrument:DLOSs? on page 160
:SLOSs?	:STATus:INSTrument:SLOSs? on page 160
:SALoss?	:STATus:INSTrument:SALoss? on page 161
:ERRor?	:STATus:INSTrument:ERRor? on page 161
:OVERload?	:STATus:INSTrument:OVERload? on page 161
:RCINput?	:STATus:INSTrument:RCINput? on page 162

Name	Description under
:RUN?	:STATus:INSTrument:RUN? on page 162
:RUN:WAIT?	:STATus:INSTrument:RUN:WAIT? on page 162
:CDR:ULOCked?	:STATus:INSTrument:CDR:ULOCked? on page 163

: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 addressed channel is in running state. A true (1) indicates the input is in running state.
Example	:STAT:INST:RUN? 'M1.DataIn1'

:STATus:INSTrument:RUN:WAIT?

Syntax	:STATus:INSTrument:RUN:WAIT? 'identifier'[,timeout]
Input Parameters	'identifier': 'M*.DataOut1' or 'M*.DataOut2'
Description	This query waits until the addressed channel is in running state. Using this query is recommended before starting any measurement that relies on the operation of the addressed channel. The optional timeout argument is having a default value of 90 and the accepted value range is 2 ... 300. The timeout is specified in terms of seconds.

NOTE

- The VISA timeout must be set to at least the timeout value that is used for :STAT:INST:RUN:WAIT?
- *OPC? does not wait until the individual channels are running, since this cannot be ensured for channels that are clocked by externally provided signals, or for error detector channels that operate on a recovered clock signal.

Example — Change the system frequency and wait until the pattern generator channels are running again

:SOUR:FREQ 'M1.ClkGen',16e9;:STAT:INST:RUN:WAIT? 'M1.DataOut1';
:STAT:INST:RUN:WAIT? 'M1.DataOut2'

: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:ULOC? 'M1.DataIn1'

:STATus:INSTrument:ALIGnment Subnode

This subsystem is reporting alignment specific status information, and is offers an alternative way of retrieving the same status information that is available via the :INPut:ALIGnment:STATus Subnode on page 366.

This subsystem has the following SCPI structure:

```
:STATus
└:INSTrument
  └:ALIGnment[:VALue]?
```

Table 32

Name	Description under
:STATus:INSTrument:ALIGnment[:VALue]?	:STATus:INSTrument:ALIGnment[:VALue]? on page 164

:STATus:INSTrument:ALIGnment[:VALue]?

Syntax :STATus:INSTrument:ALIGnment[:VALue]? 'identifier',

Input Parameters 'identifier': Any error detector location and group name. E.g. 'M1.DataIn1', 'M2.DataIn', 'INF1.DataIn1'

Description This command retrieves alignment status information for an error detector or error detector group.

This command can be used to retrieve information about the alignment procedure that is working on the addressed error detector channel, or a group of error detectors. This command is reporting the status independently of the executing alignment procedure. There is no differentiation in respect to different types of alignments.

This command returns the following values:

- INPROGRESS - At least one addressed error detector is executing an alignment procedure.
- ABORTED - The alignment procedure has been aborted by the user before all aligning error detector channels finished the procedure.

- FAILED - All addressed error detector channels have finished the alignment procedure, but at least one was unable to finish successfully.
- SUCCESS - The addressed error detector channels have successfully finished the alignment procedure. Error detector channels that are addressed by the command but return UNKNOWN state, do not contribute to the result.
- UNKNOWN - None of the addressed error detectors has executed an alignment procedure yet.

NOTE This command cannot be used in the same compound SCPI statement that is starting the alignment procedure. If this command is sent to the instrument right after the command that is initiating the alignment procedure, then the returned value is undefined and can reflect the state that was valid before the alignment. This can be overcome by reading any value from the instrument (e.g. *ESE?), except for *OPC? as this will block until the alignment procedure is finished.

Example :INP:INST:ALIGN? '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 33

Name	Description under
[:SOURce]	:TRIGger[:SOURce][?] on page 168
[:SOURce]:FREQuency?	:TRIGger[:SOURce]:FREQuency? on page 169
:INTernal[:SOURce][?]	:TRIGger:INTernal[:SOURce][?] on page 169
:REFerence[:FREQuency][?]	:TRIGger:REFerence[:FREQuency][?] on page 170
:DIRect[:FREQuency][?]	:TRIGger:DIRect[:FREQuency][?] on page 170
:DIRect[:FREQuency]:DETect	:TRIGger:DIRect[:FREQuency][?] on page 170
:CMULtiplier:LBWidth[?]	:TRIGger:CMULtiplier:LBWidth[?] on page 171

Name	Description under
:CMULtiplier:FREQuency[?]	:TRIGger:CMULtiplier:FREQuency:DETect on page 173
:CMULtiplier:FREQuency:DETect	:TRIGger:CMULtiplier:FREQuency:DETect on page 173
:CMULtiplier:FREQuency:MULTiplier[?]	:TRIGger:CMULtiplier:FREQuency:MULTiplier[?] on page 171
:CMULtiplier:FREQuency:DIVider[?]	:TRIGger:CMULtiplier:FREQuency:DIVider[?] on page 172
:CMULtiplier:FREQuency[?]	:TRIGger:CMULtiplier:FREQuency[?] on page 172
:CMULtiplier:FREQuency:DETect	:TRIGger:CMULtiplier:FREQuency:DETect on page 173

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 34

License	Description	Modules
M8009A-062	Clock Generator two channel 60 GHz	M8009A
M8009A-0G6	Reference Clock Multiplier	M8009A
M8041A-0G6	Reference Clock Input with Multiplying PLL, Clockgroup-wide License	M8041A
M8041A-G08	Generator one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8042A-G32	Pattern generation up to 32 GBd for NRZ and PAM4	M8042A
M8045A- G32	32 GBd Pattern Generator	M8045A
M8045A-0G6	Reference Clock Input with Multiplying PLL	M8045A
M8062A-G32	32 Gb/s Pattern generator front-end	M8062A

:TRIGger[:SOURce][?]

For Modules: M8041A, M8062A, M8045A, M8009A, M8053A, DSO634A, DSA634A, DSAX96204A and DSOX96204A

Syntax

:TRIGger[:SOURce] 'identifier', INTernal|REFerence|DIRect|CMULtiplier

:TRIGger[:SOURce]? 'identifier'

Input Parameters

'identifier': 'M*.ClkGen', 'INF*.Common'

Return Range

INT|REF|DIR|CMUL

Description

This command 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

For M8053A, only INTernal and REFerence clock source are supported.

This SCPI is applicable for M8041A, M8062A, M8045A, M8009A, M8053A, DSO634A, DSA634A, DSAX96204A and DSOX96204A.

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 applicable for M8041A and M8042A.
Example	:TRIG:FREQ? 'M1.TrigOut'

:TRIGger:INTernal[:SOURce][?]

Syntax	:TRIGger:INTernal[:SOURce] 'identifier', <AXIFrame|INTernal> :TRIGger:INTernal[:SOURce]? 'identifier'
Input Parameters	'identifier': M*.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 applicable for M8041A, M8045A and M8009A.
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': 'M*.ClkGen', 'INF*.Common'
	<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 applicable for M8041A, M8045A, M8009A, M8053A, DSO634A, DSA634A, DSAX96204A and DSOX96204A.
Example	:TRIG:REF:FREQ '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 applicable for M8041A, M8045A and M8009A.
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 applicable for M8041A and M8045A.
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 applicable for M8041A and M8009A.
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 141.

Description	This command specifies the external reference clock frequency multiplier.
	This SCPI is applicable for M8041A, M8045A and M8009A.
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 applicable for M8041A, M8045A and M8009A.
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 141.
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 applicable for M8041A, M8045A and M8009A.
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 applicable for M8041A and M8045A.
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][?]
│  └:COUPling[?]
├ :IMPairments:DCDistortion
│  ├:STATe [?]
│  └[:VALue][?]
├ :PULSe
│  └...
├ :VOLTage
│  └...
├ :JITTer
│  └...
│  ├ :HFRequency
│  │  └...
│  │  ├ :PERiodic(*)[?]
│  │  │  └...
│  │  ├ :BUNCorrelate
│  │  │  └...
│  │  ├ :RANDom
│  │  │  └...
│  │  └ :SPECtrally
│  │     └...
│  ├ :LFRequency
│  │  └...
│  │  ├ :PERiodic
│  │  │  └...
│  │  └ :RSSClocking
│  │     └...
│  └ :SWEep
│     └...
├:SSCLocking
│  └...
├:INTerference
│  └...
├:CONFigure
│  └...
├:MODulated:CLOCk
│  └...
└ :SSCLocking:SEGMented
   └...
```

This subsystem has the following commands and subnodes:

Table 35

Name	Description under
:FREQuency[?]	[:SOURce]:FREQuency[?] on page 177
:FREQuency:RASTer?	[:SOURce]:FREQuency:RASTer? on page 178
:ATTenuation[?]	[:SOURce]:ATTenuation[?] on page 178
:PERiod[?]	[:SOURce]:PERiod[?] on page 179
:DRATe[?]	[:SOURce]:DRATe[?] on page 180
:MEMory:CONFigure[:VALue][?]	[:SOURce]:MEMory:CONFigure[:VALue][?] on page 180
:MEMory:CONFigure:COUPling[?]	[:SOURce]:MEMory:CONFigure:COUPling[?] on page 181
:IMPairments:DCDistortion:STATe[?]	[:SOURce]:IMPairments:DCDistortion:STATe[?] on page 182
:IMPairments:DCDistortion[:VALue][?]	[:SOURce]:IMPairments:DCDistortion[:VALue][?] on page 182
Subnodes	
:PULse	[:SOURce]:PULSe Subnode on page 183
:VOLTage	[:SOURce]:VOLTage Subnode on page 187
:JITTer	[:SOURce]:JITTer Subnode on page 193
:JITTer:HFRequency	[:SOURce]:JITTer:HFRequency Subnode on page 196
:JITTer:LFRequency	[:SOURce]:JITTer:LFRequency Subnode on page 213
:JITTer:SWEep	[:SOURce]:JITTer:SWEep Subnode on page 221
:SSCLocking	[:SOURce]:SSCLocking Subnode on page 227
:INTerference	[:SOURce]:INTerference Subnode on page 233
:CONFigure	[:SOURce]:CONFigure Subnode on page 255
:MODulated:CLOCk	[:SOURce]:MODulated:CLOCk Subnode on page 256
SSCLocking:SEGMented	[:SOURce]:SSCLocking:SEGMented Subnode on page 266

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 36

License	Description	Modules
AWG License	Required for INTerference:RANDom, INTerference:SINUsoidal, MODulated:CLOCk subnode commands	M8194A, M8195A. M8196A
M8009A-062	Clock Generator two channel 60 GHz	M8009A
M8009A-0G3	Advanced Jitter Modulation for up to two Channels	M8009A
M8009A-0G6	Reference Clock Multiplier	M8009A
M8041A-0A2	Second Channel for Analyzer	M8041A
M8041A-0G2	Second Channel for Generator	M8041A
M8041A-0G3	Advanced Jitter Sources for Receiver Characterization, Module-wide License	M8041A
M8041A-0G5	Adjustable ISI, Module-wide License	M8041A
M8041A-0G6	Reference Clock Input with Multiplying PLL, Clockgroup-wide License	M8041A
M8041A-C08	BERT one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8041A-G08	Generator one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8041A-G16	Generator one Channel, Data Rate up to 16 Gb/s	M8041A
M8042A-0G4	Deemphasis	M8042A
M8042A-G32	Pattern generation up to 32 GBd for NRZ and PAM4	M8042A
M8045A-0G3	Advanced jitter sources for receiver characterization	M8045A
M8045A-0G4	Deemphasis	M8045A
M8045A-0G6	Reference Clock Input with Multiplying PLL	M8045A
M8045A-0P3	PAM4 encoding up to 32 GBd	M8045A
M8045A-0P6	Extension to PAM4 encoding up to 64 GBd	M8045A
M8045A-G32	Pattern generator one channel NRZ, data rate up to 32 GBd (requires remote head, M8057A/B)	M8045A
M8062A-0G5	Adjustable Intersymbol Interference (ISI) License	M8062A

License	Description	Modules
M8062A-C32	32 Gb/s BERT front-end	M8062A
M8062A-G32	32 Gb/s Pattern generator front-end	M8062A
M8070ISIB	Adjustable ISI Package for M8000 Series User Guide	M8042A, M8045A

[:SOURce]:FREQuency[?]

Syntax: [:SOURce]:FREQuency 'identifier', <NRf>

[:SOURce]:FREQuency? 'identifier'

Input Parameters: 'identifier': 'M*.ClkGen'

<NRf>: Set system frequency.

Return Range: For M8041A: 256 MHz to 16.207 GHz

For M8194A: 256 MHz to 113 GHz

For M8195A: 256 MHz to 32.5 GHz

For M8045A: 2.025 GHz to 64.828 GHz

Description: Sets the frequency of the synthesizer in the M8020A/M8040A/M8050A. Acceptable units include kHz, MHz, GHz and exponents (for example, 10E9 is the same as 10 GHz).

This SCPI is applicable for M8041A, M8194A, M8195A, M8062A, M8196A, M8045A and M8009A.

The SCPI is read only for M8062A.

Example: :FREQ 'M1.ClkGen', 10e9

[:SOURce]:FREQuency:RASTer?

Syntax [:SOURce]:FREQuency:RASTer? 'identifier'

Input Parameters 'identifier': 'M1.ClkGen'

Description This query returns the sample frequency of the output DAC for the currently selected reference clock and its range.

This SCPI is applicable for M8053A.

Example -> :FREQ:RAST? 'M1.ClkGen'

<- 2.16000000000000E+11

[:SOURce]:ATTenuation[?]

Syntax [:SOURce]:ATTenuation 'identifier', <Value>

[:SOURce]:ATTenuation? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1', 'M*.DataOut2'

Description This command sets the external frequency-independent attenuation in dB. It scales the source amplitudes such that they are calibrated after the attenuation rather at the output of the instrument.

This SCPI is applicable for M8053A.

Example :SOURce:ATTenuation 'M1.DataOut1', 2

:SOURce:ATTenuation? 'M1.DataOut1', 2 -> 2e-12

[:SOURce]:PERiod[?]

Syntax

[:SOURce]:PERiod 'identifier', <NRf>

[:SOURce]:PERiod? 'identifier'

Input Parameters

'identifier': 'M*.ClkGen'

<NRf>: Set system period.

Return Range

For M8041A: 61.702 ps to 3.906249 ns

For M8194A: 8.8 ps to 3.90625 ns

For M8195A: 30.77 ps to 3.90625 ns

For M8045A: 15.426 ps to 493.827

Description

Sets the M8020A/M8040A/M8050A 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 applicable for M8041A, M8194A, M8195A, M8196A, M8045A and M8009A.

Example

:PER 'M1.ClkGen', 61.702ps

[:SOURce]:POWer[?]

Syntax

[:SOURce]:POWer 'identifier', <NRf>

[:SOURce]:POWer? 'identifier'

Input Parameters

'identifier': 'M1.ChClkOut2'

<NRf>: Power levels

Return Range

-5 dBm to 10 dBm

Description

This commands sets the power levels on the clock modules.

This query returns the current power levels.

This SCPI is applicable for M8009A clock module.

Example

:POW 'M1.ChClkOut2',-1.5

:POW? 'M1.ChClkOut2' -> -1.5

[: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 applicable for M8062A.
Example	:DRAT 'M2.ClkGen', 10e9

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

Syntax

[:SOURce]:MEMory:CONFigure 'identifier', <SINGle | DUAL | FOUR | INTerference>

[:SOURce]:MEMory:CONFigure? 'identifier'

Input Parameters

'identifier': 'M1'

SINGle | DUAL | FOUR | INTerference

These input parameters are applicable for M8194A and M8195A.
IFOur | INTerference

These input parameters are applicable for M8196A.

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.
- INTerference: The Data Out locations of all channels are used to generate interference.

NOTE

The following modes changes are done:

- RANDom | SINusoidal: Not available (Obsolete)
- INTerference: Added (New)

NOTE

If the user sets the SINusoidal configuration mode (obsolete) using the above SCPI, then the following steps will be done:

1. Interference Source will automatically be selected.
2. RI State will be disabled on all channels.
3. SI1 State will be enabled on all channels
4. SI2 State will be disabled on all channels.

If the user sets the RANDom configuration mode (obsolete) using the above SCPI, then the following steps will be done:

1. Interference Source will automatically be selected.
2. RI State will be enabled on all channels.
3. SI1 State and the SI2 State will be disabled on all channels.

[:SOURce]:MEMory:CONFigure:COUPling[?]

Syntax

[:SOURce]:MEMory:CONFigure:COUPling 'Identifier', NONe|ONETwo|ONEFour|THReeandfour

[:SOURce]:MEMory:CONFigure:COUPling? 'Identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

NONe | ONETwo | ONEFour | THReeandfour

Description

This command sets the Channel Coupling for Channel One & Two and Channel Three & Four. In case of two channel license module, Channel One and Four will be coupled.

This query returns the current setting.

This SCPI is applicable for M8194A, M8195A, M8196A and M8053A.

Example :SOUR:MEM:CONF:COUP "M1.DataOut1", ONET

:SOUR:MEM:CONF:COUP? 'M1'

ONET

[:SOURce]:IMPairments:DCDistortion:STATe[?]

Syntax [:SOURce]:IMPairments:DCDistortion:STATe 'Identifier', 0 | 1 | OFF | ON

[:SOURce]:IMPairments:DCDistortion:STATe? 'Identifier'

Input Parameters 'identifier': 'M*.TrigOut'

0 | 1 | OFF | ON

Description This command enables or disables the effect of DCD on the Trigger Out.

This query returns the current setting.

This SCPI is applicable for M8045A.

Example :SOUR:IMP:DCD:STAT 'M1.TrigOut', ON

:SOUR:IMP:DCD:STAT? 'M1.TrigOut' -> 1

[:SOURce]:IMPairments:DCDistortion[:VALue][?]

Syntax [:SOURce]:IMPairments:DCDistortion[:VALue] 'Identifier', <NRf>

[:SOURce]:IMPairments:DCDistortion[:VALue]? 'Identifier'

Input Parameters 'identifier': 'M*.TrigOut'

<NRf> - DCD value in the range of -20 ps to +20 ps.

Description This command sets the DCD value in picoseconds of the subrate clock that is generated at Trigger Out.

This query returns the current setting.

This SCPI is applicable for M8045A.

Example :SOUR:IMP:DCD:VAL 'M1.TrigOut', -20E-12

:SOUR:IMP:DCD:VAL? 'M1.TrigOut' -> -20E-12

[:SOURce]:PULSe Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:PULSe
 ├:FILTer
 │└:TYPE[?]
 ├:ROFFfactor[?]
 ├:DELay[?]
 └:TRANsition
  ├:FIXed[?]
  └[:LEADing][?]
```

This subnode has the following commands:

Table 37

Name	Description under
:PULSe:FILTer:TYPE[?]	[:SOURce]:PULSe:FILTer:TYPE[?] on page 183
:PULSe:ROFFfactor[?]	[:SOURce]:PULSe:ROFFfactor[?] on page 184
:PULSe:DELay[?]	[:SOURce]:PULSe:DELay[?] on page 184
:PULSe::TRANsition:FIXed[?]	[:SOURce]:PULSe:TRANsition:FIXed[?] on page 185
:PULSe::TRANsition[:LEADing][?]	[:SOURce]:PULSe:TRANsition[:LEADing][?] on page 186

[:SOURce]:PULSe:FILTer:TYPE[?]

Syntax [:SOURce]:PULSe:FILTer:TYPE 'identifier', <GAUS | RCOS>

[:SOURce]:PULSe:FILTer:TYPE? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'

<GAUS | RCOS| RRCO> (For M8194A)

<GAUS | RCOS|> (For M8196A)

Return Range	GAUS | RCOS | RRCO
Description	This command specifies filter type used on the waveform. Depending upon the specific waveform used, the waveform gets properties like transition time or bandwidth limits. This query returns the present setting. This SCPI is applicable for M8194A, and M8196A.
Example	:PULS:FILT:TYPE 'M1.DataOut2', GAUS

[:SOURce]:PULSe:ROFFfactor[?]

Syntax	[:SOURce]:PULSe:ROFFfactor 'identifier', <NRf> [:SOURce]:PULSe:ROFFfactor? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4' <NRf>: Sets the roll-off-factor value. The value should be between 0 to 1.
Description	This command sets the roll-off factor for the raised cosine pulse shaping filter. It is a measure for the excess bandwidth the waveform contains beyond Nyquist bandwidth. This query returns the present setting. This SCPI is applicable for M8194A and M8196A.
Example	:PULS:ROFF 'M1.DataOut2', 0.50

[:SOURce]:PULSe:DELay[?]

Syntax	[:SOURce]:PULSe:DELay 'identifier', <NRf> [:SOURce]:PULSe:DELay? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', 'M*.DataOut4', 'M*.ChClkOut2', or 'M*.ClkOut32G' <NRf>: Set the pulse delay time.
Return Range	For M8041A: 0.0 ps to 100 ns For M8194A: -1ns to 1ns For M8195A: -1ns to 1ns For M8196A: -1ns to 1ns

For M8062A: 0 ps to 100 ns

For M8045A: 0 ps to 100 ns

For M8009A: 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 applicable for M8041A, M8051A, M8194A, M8195A, M8196A, M8062A, M8045A and M8009A.

Example — :PULS:DEL 'M1.DataOut2', 2.5E-11

[: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 applicable 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 applicable for M8194A, M8195A, M8196A.

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][?]
  ├:CFAMplitude[?]
  ├:OFFSet[?]
  ├:HIGH[?]
  ├:LOW[?]
  └:RANGe
    ├[:SELect][?]
    └:AUTO[?]
```

This subnode has the following commands:

Table 38

Name	Description under
[:AMPLitude][?]	[:SOURce]:VOLTage[:AMPLitude][?] on page 188
CFAMplitude[?]	[:SOURce]:VOLTage:CFAMplitude[?] on page 188
:OFFSet[?]	[:SOURce]:VOLTage:OFFSet[?] on page 189
:HIGH[?]	[:SOURce]:VOLTage:HIGH[?] on page 190
:LOW[?]	[:SOURce]:VOLTage:LOW[?] on page 190
:RANGe[:SELect][?]	[:SOURce]:VOLTage:RANGe[:SELect][?] on page 191
:RANGe:AUTO[?]	[:SOURce]:VOLTage:RANGe:AUTO[?] on page 191

[: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', 'M*.CtrlOutA', 'M*.SyncMrkA', 'M*.SyncMrkB', 'M*.SampleMrk', 'M*.ClkOut16G', 'M*.ChClkOut2' or 'M*.ClkOut32G'

<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 applicable for M8041A, M8042A, M8043A, M8045A, M8051A, M8053A, M8054A, M8062A, M8194A, M8195A, M8196A and M8009A.

For M8053A, this SCPI is applicable as a query only.

NOTE The following information is applicable only for AWG modules (M8194A, M8195A, M8196A, and M8054A):

If only one interference source is enabled, only then the overall amplitude can be set, and the respective source amplitude will be adjusted.

If more than one interference source is enabled, then an error message will be returned.

Example :VOLT 'M1.DataOut1', 500mv

:VOLT? 'M1.DataOut1'

[:SOURce]:VOLTage:CFAMplitude[?]

Syntax [:SOURce]:VOLTage:CFAMplitude 'Identifier', <NRf>

[:SOURce]:VOLTage:CFAMplitude? 'Identifier'

Input Parameters 'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

Description This command sets the amplitude correction factor for each channel. Amplitude correction factor parameter will be functional only in channel coupled mode. For details on how to set channel coupling mode, see [:SOURce]:MEMory:CONFigure:COUPling[?] on page 181.

This SCPI is applicable for M8194A, M8195A and M8196A.

Example :SOUR:VOLT:CFAM 'M1.DataOut1',1

:SOUR:VOLT:CFAM? 'M1.DataOut1'l

1

[: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', 'M*.CtrlOutA', 'M*.SyncMrkA', 'M*.SyncMrkB' or 'M*.SampleMrk'

<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 applicable for M8041A, M8042A, M8043A. M8045A, M8051A, M8062A, M8194A, M8195A, M8196A, M8009A and M8053A.

For M8053A, this SCPI is applicable as a query only.

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', 'M*.CtrlOutA', 'M*.SyncMrkA', 'M*.SyncMrkB' or 'M*.SampleMrk'
	<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 applicable for M8041A, M8042A, M8043A, M8045A, M8051A, M8062A, M8194A, M8195A, M8196A and M8009A.
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', 'M*.CtrlOutA', 'M*.SyncMrkA', 'M*.SyncMrkB' or 'M*.SampleMrk'
	<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 applicable for M8041A, M8042A, M8043A, M8045A, M8051A, M8062A, M8194A, M8195A, M8196A and M8009A.
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 applicable for M8041A, M8042A, M8051A 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 applicable for M8041A, M8042A, M8051A and M8062A.
Example	:VOLT:RANG:AUTO 'M2.DataOut2', off

[:SOURce]:VOLTage:TERMination[?]

Syntax	[:SOURce]:VOLTage:TERMination 'identifier', <NRf> [:SOURce]:VOLTage:TERMination? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', 'M*.DataOut4', 'M*.DataOut' or 'M*.SampleMrk' <NRf>: Set termination voltage. The acceptable range is -750 mV to 1.25 V.
Return Range	<NRf>
Description	This command sets the termination 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 applicable for M8009A.
Example	:VOLT:TERM 'M1.DataOut1', 500 mV :VOLT:TERM? 'M1.DataOut1' -> 5.000000000E-01

[: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 39

Name	Description under
[:GLOBal][:STATe][?]	[:SOURce]:JITTer[:GLOBal][:STATe][?] on page 194
:CONFigure[:DELay][?]	[:SOURce]:JITTer:CONFigure[:DELay][?] on page 194
:CONFigure:HMODe[?]	[:SOURce]:JITTer:CONFigure:HMODe[?] on page 195
:HFRequency	[:SOURce]:JITTer:HFRequency Subnode on page 196
:LFRequency	[:SOURce]:JITTer:LFRequency Subnode on page 213

[: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 and work for all modules (except M8054A).
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', 'M*.DataOut', 'M*.ChClkOut2', or 'M*.ClkOut32G'
	<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 applicable for M8041A, M8042A, M8051A, M8062A, M8045A and M8009A.
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 applicable 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 40

Name	Description under
:UNIT[?]	[:SOURce]:JITTer:HFRequency:UNIT[?] on page 197
:EXTernal[:STATe][?]	[:SOURce]:JITTer:HFRequency:EXTernal[:STATe][?] on page 197
:PMMode[?]	[:SOURce]:JITTer:HFRequency:PMMode[?] on page 198
:PERiodic(*)[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?] on page 199
:BUNCorrelate	[:SOURce]:JITTer:HFRequency:BUNCorrelate Subnode on page 202
:RANDom	[:SOURce]:JITTer:HFRequency:RANDom Subnode on page 206
:SPECtrally	[:SOURce]:JITTer:HFRequency:SPECtrally Subnode on page 210

[: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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A, M8194A, M8196A and M8009A.
	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 applicable for M8041A, M8051A and M8045A.
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 applicable for M8041A, M8051A and M8045A.
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 41

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?] on page 199
:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude[?] on page 200
:FREQuency[?]	[:SOURce]:JITTer:HFRequency:PERiodic(*):FREQuency[?] on page 200

[: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', 'M*.DataOut' or 'M*.ClkOut16G'

<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 applicable for M8041A, M8045A, M8051A, M8062A, M8194A, M8196A and M8009A.

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

Example :JITT:HFR:PER1 '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 141.

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 applicable for M8041A, M8045A, M8051A, M8062A, M8194A, M8196A and M8009A.

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

Example :JITT:HFR:PER:AMPL 'M2.DataOut2', 100e-12

[:SOURce]: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 applicable for M8041A, M8045A, M8051A, M8062A, M8194A, M8196A and M8009A.

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 42

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate[:STATe][?] on page 203
:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:AMPLitude[?] on page 203
:DRATe[?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:DRATe[?] on page 204
:FILTer[:SELect][?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:FILTer[:SELect][?] on page 204
:PRBSequence[:SELect][?]	[:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect][?] on page 205

[: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', 'M*.DataOut' or 'M*.ClkOut16G'
	<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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
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|LP150_20DB|LP300_20DB>: Select the low-pass filter.
Return Range	LP50|LP100|LP200|LP150_20DB|LP300_20DB
Description	This command selects the low-pass filter for bounded uncorrelated jitter (50, 100, or 200 MHz). This SCPI is applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
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 43

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:HFRequency:RANDom[:STATe][?] on page 207
:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:RANDom:AMPLitude[?] on page 207
:FILTer[:LPASs][?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs][?] on page 208
:FILTer[:LPASs]:VALue[?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer[:LPASs]:VALue[?] on page 208
:FILTer:HPASs[?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs[?] on page 209
:FILTer:HPASs:VALue[?]	[:SOURce]:JITTer:HFRequency:RANDom:FILTer:HPASs:VALue[?] on page 209

[: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', 'M*.DataOut' or 'M*.ClkOut16G'
	<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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A, M8194A, M8196A, and M8009A.
	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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A, M8194A, M8196A and M8009A.
	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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.

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 applicable for M8042A, M8194A, M8195A and M8196A.

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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
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 applicable for M8194A, M8195A and M8196A. 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 44

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe][?] on page 211
:AMPLitude[?]	[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*)[?] on page 211
:FILTer[:LPASs][:STATe][?]	[:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs][:STATe][?] on page 212

[: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', 'M*.DataOut' or 'M*.ClkOut16G'
	<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 applicable for M8041A, M8042A, M8051A, M8062A and M8009A.
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 applicable for M8041A, M8042A, M8051A, M8062A and M8009A.
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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
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 45

Name	Description under
:UNIT[?]	[:SOURce]:JITTer:LFRequency:UNIT[?] on page 214
:PERiodic	[:SOURce]:JITTer:LFRequency:PERiodic Subnode on page 215
:RSSClocking	[:SOURce]:JITTer:LFRequency:RSSClocking Subnode on page 219

[: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', 'M*.DataOut', 'M*.ClkOut16G'
	<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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
Example	:JITT:LFR:UNIT 'M1.DataOut2',TIME

[:SOURce]:JITTer:LFRequency:PERiodic Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:JITTer
  └:LFRequency
    ├:PERiodic1
    │ ├[:STATe][?]
    │ ├:AMPLitude[?]
    │ └:FREQuency[?]
    └:PERiodic2
      ├[:STATe][?]
      ├:AMPLitude[?]
      └:FREQuency[?]
```

This subnode has the following commands:

Table 46

Name	Description under
PERiodic1:[:STATe][?]	[:SOURce]:JITTer:LFRequency:PERiodic1[:STATe][?] on page 216
PERiodic1:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:PERiodic1:AMPLitude[?] on page 216
PERiodic1:FREQuency[?]	[:SOURce]:JITTer:LFRequency:PERiodic1:FREQuency[?] on page 217
PERiodic2:[:STATe][?]	[:SOURce]:JITTer:LFRequency:PERiodic2[:STATe][?] on page 217
PERiodic2:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:PERiodic2:AMPLitude[?] on page 218
PERiodic2:FREQuency[?]	[:SOURce]:JITTer:LFRequency:PERiodic2:FREQuency[?] on page 218

[:SOURce]:JITTer:LFRequency:PERiodic1[:STATe][?]

Syntax [:SOURce]:JITTer:LFRequency:PERiodic1[:STATe] 'identifier', <ON|OFF|1|0>

[:SOURce]:JITTer:LFRequency:PERiodic1[:STATe]? 'identifier'

Input Parameters 'identifier': 'M1.ClkOut', 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut', or 'M*.ClkOut16G'

<ON|OFF|1|0>: Enable/disable the low frequency periodic 1 jitter source.

Return Range 1|0

Description This command enables/disables the low frequency periodic 1 jitter source.

This SCPI is applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.

Example :JITT:LFR:PER1 'M1.DataOut', ON

[:SOURce]:JITTer:LFRequency:PERiodic1:AMPLitude[?]

Syntax [:SOURce]:JITTer:LFRequency:PERiodic1:AMPLitude 'identifier', <NRf>

[:SOURce]:JITTer:LFRequency:PERiodic1:AMPLitude? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', 'M*.DataOut' or 'M*.ClkOut16G'

<NRf>: Set periodic 1 low frequency jitter amplitude.

Return Range NRf

Description This command sets the amplitude of the periodic 1 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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.

Example :JITT:LFR:PER1:AMPL 'M2.DataOut2', 1

[:SOURce]:JITTer:LFRequency:PERiodic1:FREQuency[?]

Syntax: [:SOURce]:JITTer:LFRequency:PERiodic1:FREQuency 'identifier', <NRf>

[:SOURce]:JITTer:LFRequency:PERiodic1:FREQuency? 'identifier'

Input Parameters: 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', 'M*.DataOut' or 'M*.ClkOut16G'

<NRf>: Set periodic 1 jitter frequency.

Return Range: NRf

Description: This command sets the frequency of the periodic 1 jitter. Acceptable units include Hz, kHz, MHz and exponents (for example, 1E3 is the same as 1,000 Hz).

This SCPI is applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.

Example: :JITT:LFR:PER1:FREQ 'M2.DataOut2', 1e6

[:SOURce]:JITTer:LFRequency:PERiodic2[:STATe][?]

Syntax: [:SOURce]:JITTer:LFRequency:PERiodic2[:STATe] 'identifier', <ON|OFF|1|0>

[:SOURce]:JITTer:LFRequency:PERiodic2[:STATe]? 'identifier'

Input Parameters: 'identifier': 'M1.ClkOut', 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut' or 'M*.ClkOut16G'

<ON|OFF|1|0>: Enable/disable the low frequency periodic 2 jitter source.

Return Range: 1|0

Description: This command enables/disables the low frequency periodic 2 jitter source.

This SCPI is applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.

Example: :JITT:LFR:PER2 'M1.DataOut', ON

[:SOURce]:JITTer:LFRequency:PERiodic2:AMPLitude[?]

Syntax	[:SOURce]:JITTer:LFRequency:PERiodic2:AMPLitude 'identifier', <NRf>
	[:SOURce]:JITTer:LFRequency:PERiodic2:AMPLitude? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', 'M*.DataOut' or 'M*.ClkOut16G'
	<NRf>: Set periodic 2 low frequency jitter amplitude.
Return Range	NRf
Description	This command sets the amplitude of the periodic 2 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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
Example	:JITT:LFR:PER2:AMPL 'M2.DataOut2', 1

[:SOURce]:JITTer:LFRequency:PERiodic2:FREQuency[?]

Syntax	[:SOURce]:JITTer:LFRequency:PERiodic2:FREQuency 'identifier', <NRf>
	[:SOURce]:JITTer:LFRequency:PERiodic2:FREQuency? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', 'M*.DataOut' or 'M*.ClkOut16G'
	<NRf>: Set periodic 2 jitter frequency.
Return Range	NRf
Description	This command sets the frequency of the periodic 2 jitter. Acceptable units include Hz, kHz, MHz and exponents (for example, 1E3 is the same as 1,000 Hz).
	This SCPI is applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
Example	:JITT:LFR:PER2: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 47

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:LFRequency:RSSClocking[:STATe][?] on page 219
:AMPLitude[?]	[:SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude[?] on page 220
:FREQuency[?]	[:SOURce]:JITTer:LFRequency:RSSClocking:FREQuency[?] on page 220

[: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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.

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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
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 applicable for M8041A, M8042A, M8045A, M8051A, M8062A and M8009A.
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 48

Name	Description under
[:STATe][?]	[:SOURce]:JITTer:SWEep[:STATe][?] on page 222
:FREQuency:STARt[?]	[:SOURce]:JITTer:SWEep:FREQuency:STARt[?] on page 222
:FREQuency:STOP[?]	[:SOURce]:JITTer:SWEep:FREQuency:STOP[?] on page 223
:TIME[?]	[:SOURce]:JITTer:SWEep:TIME[?] on page 223
:STEP:[VALue][?]	[:SOURce]:JITTer:SWEep:STEP:[VALue][?] on page 224
:STEP:DISTance[?]	[:SOURce]:JITTer:SWEep:STEP:DISTance[?] on page 224

Name	Description under
:AMPLitude[:VALue]?	[:SOURce]:JITTer:SWEep:AMPLitude[:VALue]? on page 224
:AMPLitude:MODE[?]	[:SOURce]:JITTer:SWEep:AMPLitude:MODE[?] on page 225
:AMPLitude:[VALue]?	[:SOURce]:JITTer:SWEep:AMPLitude[:VALue]? on page 224
:PLOT[:VALue]?	[:SOURce]:JITTer:SWEep:PLOT[:VALue]? on page 225
:PLOT:INPut?	[:SOURce]:JITTer:SWEep:PLOT:INPut? on page 226
:DATA:FILE[?]	[:SOURce]:JITTer:SWEep:DATA:FILE[?] on page 226

[: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 applicable 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 applicable 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 applicable 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 applicable 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 applicable 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 applicable 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 applicable 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 applicable 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 applicable 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 applicable 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 applicable 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 49

Name	Description under
[:STATe][?]	[:SOURce]:SSCLocking[:STATe][?] on page 228
GLOBal[:STATe][?]	[:SOURce]:SSCLocking:GLOBal[:STATe][?] on page 228
:DEViation[?]	[:SOURce]:SSCLocking:DEViation[:VALue][?] on page 229
:TYPE[?]	[:SOURce]:SSCLocking:TYPE[?] on page 230
:PROFile[?]	[:SOURce]:SSCLocking:PROFile[?] on page 231
:SHAPe[?]	[:SOURce]:SSCLocking:SHAPe[?] on page 232
:FREQuency[?]	[:SOURce]:SSCLocking:FREQuency[?] on page 232

[: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.

The SSC mode can be only enabled if Global SSC state is activated.

This SCPI is applicable for M8041A, M8194A, M8195A, M8196A, M8045A and M8009A.

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[:VALue][?]

Syntax	[:SOURce]:SSCLocking:DEViation[:VALue] 'identifier', <NRf> [:SOURce]:SSCLocking:DEViation[:VALue]? '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 applicable for M8041A, M8194A, M8195A, M8196A, M8045A and M8009A.
Example	The below mentioned SCPI command sets the spread spectrum clocking (SSC) deviation to '0.2' percent for M8041A/M8045A: SSCL:DEV 'M1.ClkGen', 0.2 The below mentioned SCPI command queries the spread spectrum clocking (SSC) deviation for M8041A/M8045A: SSCL:DEV? 'M1.ClkGen' 0.2

[:SOURce]:SSCLocking:DEViation:UP[?]

Syntax	[:SOURce]:SSCLocking:DEViation:UP 'identifier', <NRf> [:SOURce]:SSCLocking:DEViation:UP? 'identifier'
Input Parameters	'identifier': 'M1.ClkGen' <NRf>: Set peak up spread spectrum clocking deviation.
Description	This command sets the up deviation from the currently set clock rate. Value range is from 0.0 % to +1.0 %. Sending this command will automatically change the SSC type to ASYMmetric. This SCPI is applicable for M8041A and M8045A.
Example	The below mentioned SCPI command sets the up deviation to '0.25' percent for M8041A/M8045A: SSCL:DEV:UP 'M1.ClkGen', 0.25 The below mentioned SCPI command queries the up deviation for M8041A/M8045A:

SSCL:DEV:UP? 'M1.ClkGen'

0.25

[:SOURce]:SSCLocking:DEViation:DOWN[?]

Syntax

[:SOURce]:SSCLocking:DEViation:DOWN 'identifier', <NRf>

[:SOURce]:SSCLocking:DEViation:DOWN? 'identifier'

Input Parameters

'identifier': 'M1.ClkGen'

<NRf>: Set spread spectrum clocking deviation.

Description

This command sets the down deviation from the currently set clock rate. Value range is from -1.0 % to 0.0 %. Sending this command will automatically change the SSC type to ASYMmetric.

This SCPI is applicable for M8041A and M8045A.

Example

The below mentioned SCPI command sets down deviation to '-0.45' percent for M8041A/M8045A:

SSCL:DEV 'M1.ClkGen', -0.45

The below mentioned SCPI command queries the down deviation for M8041A/M8045A:

SSCL:DEV? 'M1.ClkGen'

-0.45

[:SOURce]:SSCLocking:TYPE[?]

Syntax

[:SOURce]:SSCLocking:TYPE 'identifier', <DOWNspread|UPSPread|CENTerspread|ASYMmetric>

[:SOURce]:SSCLocking:TYPE? 'identifier'

Input Parameters

'identifier': 'M1.ClkGen'

<DOWNspread|UPSPread|CENTerspread|ASYMmetric>: Set spread spectrum clocking type.

Return Range

DOWN|UPSP|CENT|ASYM

Description

This command sets method used to modulate the clock frequency and data. Choose between center spread, down spread, up spread or asymmetric. For SSC type as ASYMmetric, you have to individually specify up and down deviation. For details, see

[:SOURce]:SSCLocking:DEViation:UP[?] on page 229 and [:SOURce]:SSCLocking:DEViation:DOWN[?] on page 230.

This SCPI is applicable for M8041A, M8094A, M8095A, M8096A, M8045A and M8009A.

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/M8045A:

:SSCL:TYPE 'M1.ClkGen', DOWN

The below mentioned SCPI command queries the method used to modulate clock frequency and data for M8041A/M8045A:

:SSCL:TYPE? 'M1.ClkGen'

DOWN

[:SOURce]:SSCLocking:PROFile[?]

Syntax	[:SOURce]:SSCLocking:PROFile 'identifier', <TRIangular|ARBitrary|SEGMented>
	[:SOURce]:SSCLocking:PROFile? 'identifier'
Input Parameters	'identifier': 'M1.ClkGen'
	<TRIangular|ARBitrary|SEGMented>: Set spread spectrum clocking profile.
Return Range	TRI|ARB|SEGM
Description	This command sets up the spread spectrum clocking (SSC) using a triangular, arbitrary or segmented arbitrary profile.
	This SCPI is applicable for M8194A, M8196A, M8041A, M8045A and M8009A.
	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 applicable for M8041A, M8045A and M8009A.
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 applicable for M8041A, M8194A, M8195A, M8196A, M8045A and M8009A.
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[?]
  ├─:DEMBedding
  │ └─:NSTAtes[?]
  └─:DEMBedding[1|2|3|4|5|6]
    ├─:STATe[?]
    ├─:FILE[?]
    ├─:FLIP[?]
    ├─:PORT[?]
    ├─:OPERator[?]
    └─:WEIGht[?]
```

This subnode has the following commands:

Table 50

Name	Description under
[:LEVel]:HFRequency[:SOURce][?]	[:SOURce]:INTerference[:LEVel]:HFRequency[:SOURce][?] on page 235
[:LEVel]:LFRequency[:SOURce][?]	[:SOURce]:INTerference[:LEVel]:LFRequency[:SOURce][?] on page 235
[:LEVel]:CMODe[:STATe][?]	[:SOURce]:INTerference[:LEVel]:CMODe[:STATe][?] on page 235
[:LEVel]:CMODe:AMPLitude[?]	[:SOURce]:INTerference[:LEVel]:CMODe:AMPLitude[?] on page 236
[:LEVel]:CMODe:SOURce[?]	[:SOURce]:INTerference[:LEVel]:CMODe:SOURce[?] on page 236
[:LEVel]:CMODe:GAIN[?]	[:SOURce]:INTerference[:LEVel]:CMODe:GAIN[?] on page 237
[:LEVel]:DMODe[:STATe][?]	[:SOURce]:INTerference[:LEVel]:DMODe[:STATe][?] on page 237
[:LEVel]:DMODe:AMPLitude[?]	[:SOURce]:INTerference[:LEVel]:DMODe:AMPLitude[?] on page 237
[:LEVel]:DMODe:SOURce[?]	[:SOURce]:INTerference[:LEVel]:DMODe:SOURce[?] on page 238
[:LEVel]:DMODe:GAIN[?]	[:SOURce]:INTerference[:LEVel]:DMODe:GAIN[?] on page 238
:ISYMbol[:STATe][?]	[:SOURce]:INTerference:ISYMbol[:STATe][?] on page 239
:ISYMbol:MODE[?]	[:SOURce]:INTerference:ISYMbol:MODE[?] on page 239
:ISYMbol:FREQuency{1:2}[?]	[:SOURce]:INTerference:ISYMbol:FREQuency{1:2}[?] on page 239
:ISYMbol:ILOSs{1:2}[?]	[:SOURce]:INTerference:ISYMbol:ILOSs{1:2}[?] on page 240
:ISYMbol:PRESet[?]	[:SOURce]:INTerference:ISYMbol:PRESet[?] on page 240
:DEMBedding:NSTAtes[?]	[:SOURce]:INTerference:DEMBedding:NSTAtes[?] on page 243
:DEMBedding[1|2|3|4|5|6]:STATe[?]	[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:STATe[?] on page 243
:DEMBedding[1|2|3|4|5|6]:FILE[?]	[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:FILE[?] on page 244
:DEMBedding[1|2|3|4|5|6]:FLIP[?]	[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:FLIP[?] on page 244
:DEMBedding[1|2|3|4|5|6]:PORT[?]	[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:PORT[?] on page 245
:DEMBedding[1|2|3|4|5|6]:OPERator[?]	[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:OPERator[?] on page 245
:DEMBedding[1|2|3|4|5|6]:WEIGht[?]	[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:WEIGht[?] on page 246

[: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 applicable 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 applicable 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 applicable for M8041A, M8051A 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 applicable 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 applicable 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 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 applicable for M8041A, M8051A 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 applicable 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 applicable 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 applicable for 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 applicable 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 applicable 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 applicable 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 applicable 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 applicable for M8041A, M8051A and M8062A.

For M8062A the presets are available up to P11 only; please refer to the Table 52 on page -242.

The following table describes the list of presets for selected standards and traces.

Table 51 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 52 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]:INTerference:DEMBedding:NSTAtes[?]

Syntax	[:SOURce]:INTerference:DEMBedding:NSTAtes 'Identifier', <NRf> [:SOURce]:INTerference:DEMBedding:NSTAtes? 'Identifier'
Input Parameters	Identifier: 'M*.DataOut', 'M*.DataOut1', or 'M*.DataOut2' <NRf>: 1 - 6
Description	This command creates the S-Parameter profiles. You can create up to six S-Parameter profiles. This SCPI is only applicable for M8042A.
Range	1 - 6
Example	:SOUR:INT:DEMB:NSTA 'M2.DataOut1',6 :SOUR:INT:DEMB:NSTA? 'M2.DataOut1'

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:STATe[?]

Syntax	[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:STATe 'Identifier', <0|1|ON|OFF> [:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:STATe? 'Identifier'
Input Parameters	Identifier: 'M*.DataOut', 'M*.DataOut1', or 'M*.DataOut2' 0|1|ON|OFF
Description	This command enables/disables the S-Parameter profiles. The M8042A generator module supports up to six S-Parameter profiles while M8045A generator module supports only two S-Parameter profiles. This SCPI is applicable for M8042A and M8045A.
Range	0|1
Example	Command :SOUR:INT:DEMB2:STAT 'M2.DataOut1',2 :SOUR:INT:DEMB6:STAT 'M2.DataOut1',2 Query :SOUR:INT:DEMB2:STAT? 'M2.DataOut1' :SOUR:INT:DEMB6:STAT? 'M2.DataOut1'

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:FILE[?]

Syntax

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:FILE 'Identifier', <File_Path>

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:FILE? 'Identifier'

Input Parameters

Identifier: 'M*.DataOut', 'M*.DataOut1', or 'M*.DataOut2'

<File_Path>: Path of the S-Parameter file

Description

This command selects the S-Parameter file of de-embedding profile 1 to 6 to adjust ISI. The M8042A generator module supports up to six S-Parameter profiles while M8045A generator module supports only two S-Parameter profiles.

This SCPI is applicable for M8042A and M8045A.

Example

Command

:SOUR:INT:DEMB2:FILE 'M2.DataOut1','Factory/ISIBoard/M8049A_001_TRACE1.s2p'

:SOUR:INT:DEMB6:FILE 'M2.DataOut1','Factory/ISIBoard/M8049A_001_TRACE1.s2p'

Query

:SOUR:INT:DEMB2:FILE? 'M2.DataOut1'

:SOUR:INT:DEMB6:FILE? 'M2.DataOut1'

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:FLIP[?]

Syntax

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:FLIP 'Identifier', <0|1>

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:FLIP? 'Identifier'

Input Parameters

Identifier: 'M*.DataOut', 'M*.DataOut1', or 'M*.DataOut2'

<0|1>: Enable/disable flip

Description

This command flips the S-Parameter ports for de-embedding profile 1 to 6. The M8042A supports up to six S-Parameter profiles while M8045A supports only two S-Parameter profiles.

This SCPI is applicable for M8042A and M8045A.

Example

Command

:SOUR:INT:DEMB2:FLIP 'M2.DataOut1',1

:SOUR:INT:DEMB6:FLIP 'M2.DataOut1',1

Query

:SOUR:INT:DEMB2:FLIP? 'M2.DataOut1'

:SOUR:INT:DEMB6:FLIP? 'M2.DataOut1'

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:PORT[?]

Syntax

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:PORT 'Identifier', <Value>

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:PORT? 'Identifier'

Input Parameters

Identifier: 'M*.DataOut', 'M*.DataOut1', or 'M*.DataOut2'

<Value>: 'PORT12' or 'PORT13'

Description

This command selects port for S-Parameter profile for external ISI board.

This parameter will only be available if you load S4P file from the "S-Parameter Profile" using the command [:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:FILE[?] on page 244.

This SCPI is applicable for M8042A and M8045A.

Example

Command

:SOUR:INT:DEMB2:PORT 'M2.DataOut1',PORT12

:SOUR:INT:DEMB6:PORT 'M2.DataOut1',PORT12

Query

:SOUR:INT:DEMB2:PORT? 'M2.DataOut1'

:SOUR:INT:DEMB6:PORT? 'M2.DataOut1'

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:OPERator[?]

Syntax

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:OPERator 'Identifier', <DIFFerential | SEComplement| SENormal>

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:OPERator? 'Identifier'

Input Parameters

Identifier: 'M*.DataOut', 'M*.DataOut1', or 'M*.DataOut2'

<DIFFerential | SEComplement| SENormal>

Description

This command selects operator output for S-Parameter profile.

This parameter will only be available if you load S4P file from the "S-Parameter Profile" using the command [:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:FILE[?] on page 244

This SCPI is applicable for M8042A and M8045A.

Example

Command

:SOUR:INT:DEMB2:OPER 'M2.DataOut1',DIFF

:SOUR:INT:DEMB6:OPER 'M2.DataOut1',DIFF

Query

:SOUR:INT:DEMB2:OPER? 'M2.DataOut1'

:SOUR:INT:DEMB6:OPER? 'M2.DataOut1'

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:WEIGht[?]

Syntax

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:WEIGht 'Identifier', <Num>

[:SOURce]:INTerference:DEMBedding[1|2|3|4|5|6]:OPERator? 'Identifier'

Input Parameters

Identifier: 'M*.DataOut', 'M*.DataOut1', or 'M*.DataOut2'

<Num> - From 0 to 10

Description

This command sets the weight of the Intersymbol Interference (ISI).

This SCPI is applicable for M8042A and M8045A.

Example

Command

:SOUR:INT:DEMB2:WEIG 'M2.DataOut1',1

:SOUR:INT:DEMB6:WEIG? 'M2.DataOut1

Query

:SOUR:INT:DEMB2:WEIG? 'M2.DataOut1'

:SOUR:INT:DEMB6:WEIG? 'M2.DataOut1'

[:SOURce]:INTerference:RANDom Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
└:INTerference
  └:RANDom
    ├:HFRequency[?]
    ├:LFRequency[?]
    ├:STATe[?]
    ├:MODE[?]
    ├:AMPLitude[?]
    └:CREStfactor[?]
      └:UNIT[?]
```

This subnode has the following commands:

Table 53

Name	Description under
[:SOURce]:INTerference:RANDom:STATe[?]	[:SOURce]:INTerference:RANDom:STATe[?] on page 248
[:SOURce]:INTerference:RANDom:MODE[?]	[:SOURce]:INTerference:RANDom:MODE[?] on page 248
[:SOURce]:INTerference:RANDom:AMPLitude[?]	[:SOURce]:INTerference:RANDom:AMPLitude[?] on page 249
[:SOURce]:INTerference:RANDom:CREStfactor:UNIT[?]	[:SOURce]:INTerference:RANDom:CREStfactor:UNIT[?] on page 249
[:SOURce]:INTerference:RANDom:CREStfactor[?]	[:SOURce]:INTerference:RANDom:CREStfactor[?] on page 250
[:SOURce]:INTerference::RANDom:HFRequency[?]	[:SOURce]:INTerference:RANDom:HFRequency[?] on page 250
[:SOURce]:INTerference:RANDom:LFRequency[?]	[:SOURce]:INTerference:RANDom:LFRequency[?] on page 250

[:SOURce]:INTerference:RANDom:STATe[?]

Syntax

[:SOURce]:INTerference:RANDom:STATe 'identifier', <ON | OFF | 1 | 0>

[:SOURce]:INTerference:RANDom:STATe? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

<ON | OFF | 1 | 0>: Switch an RI state ON or OFF.

Description

This command switches the state ON or OFF.

This SCPI is applicable for M8194A, M8195A, M8196A, M8053A, and M8054A.

Example

The following example shows how to use this command:

:SOUR:INT:RAND:STAT 'M1.DataOut1',ON

:SOUR:INT:RAND:STAT? 'M1.DataOut1'

ON

[:SOURce]:INTerference:RANDom:MODE[?]

Syntax

[:SOURce]:INTerference:RANDom:MODE 'identifier', COMMon | DIFFerential

[:SOURce]:INTerference:SINusoidal:MODE? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

Description

This command can be used to set the Tone Mode for RI to either Common mode or Differential mode.

For M8053A, this option is only valid if channel coupling is enabled in the M8053A user interface.

This SCPI is applicable for M8194A, M8195A, M8196A, M8053A and M8054A.

Example

The following example shows how to use this command to set the Tone Mode to Differential Mode:

:SOUR:INT:RAND:MODE 'M1.DataOut1',DIFFerential

:SOUR:INT:SIN:MODE? 'M1.DataOut1'

DIFFerential

[:SOURce]:INTerference:RANDom:AMPLitude[?]

Syntax: [:SOURce]:INTerference:RANDom:AMPLitude 'identifier', <NRf>
[:SOURce]:INTerference:SINusoidal:AMPLitude? 'identifier'

Input Parameters: 'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

Description: This command can be used to set the RI Amplitude (Vrms) parameters value.

This SCPI is applicable for M8194A, M8195A, M8196A, M8053A, and M8054A.

Example: The following example shows how to use this command to set the RI Amplitude (Vrms) parameter value:

:SOUR:INT:RAND:AMPL 'M1.DataOut1',0.2

:SOUR:INT:RAND:AMPL? 'M1.DataOut1'

0.2

[:SOURce]:INTerference:RANDom:CREStfactor:UNIT[?]

Syntax: [:SOURce]:INTerference:RANDom:CREStfactor:UNIT 'identifier', < LINear | LOGarithmic >

[:SOURce]:INTerference:RANDom:CREStfactor:UNIT? 'identifier'

Input Parameters: 'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'

< LINear | LOGarithmic >

Description: This command sets the random interference crest factor unit type. Following unit types are allowed:

- LINear - This is the default setting. Select this unit to specify the crest factor as voltage ratio.
- LOGarithmic - Select this unit to specify the crest factor in dB.

This SCPI is applicable for M8194A, M8195A, M8196A, and M8054A.

Example: :INT:RAND:CRES:UNIT 'M2.DataOut1', LOG

[:SOURce]:INTerference:RANDom:CREStfactor[?]

Syntax	[:SOURce]:INTerference:RANDom:CREStfactor 'identifier', <NRf>
	[:SOURce]:INTerference:RANDom:CREStfactor? 'identifier'
Input Parameters	'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'
	<NRf>: Set random interference crest factor value
Description	This command sets the random interference crest factor value in dB or as voltage ratio depending on the currently selected crest factor unit type.
	This SCPI is applicable for M8194A, M8195A, M8196A, and M8054A.
Example	:INT:RAND:CRES 'M2.DataOut1', 12

[:SOURce]:INTerference:RANDom:HFRequency[?]

Syntax	[:SOURce]:INTerference:RANDom:HFRequency 'identifier', <NRf>
	[:SOURce]:INTerference:RANDom:HFRequency? 'identifier'
Input Parameters	'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'
	<NRf>: Set random interference high frequency
Description	This command sets the random interference high frequency in Hz.
	This SCPI is applicable for M8194A, M8195A, M8196A, M8053A, and M8054A.
Example	:INT:RAND:HFR 'M2.DataOut1', 1.0e9

[:SOURce]:INTerference:RANDom:LFRequency[?]

Syntax	[:SOURce]:INTerference:RANDom:LFRequency 'identifier', <NRf>
	[:SOURce]:INTerference:RANDom:LFRequency? 'identifier'
Input Parameters	'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'
	<NRf>: Set random interference low frequency
Description	This command sets the random interference low frequency in Hz.
	This SCPI is applicable for M8194A, M8195A, M8196A, M8053A, and M8054A.
Example	:INT:RAND:LFR 'M2.DataOut1', 1.0e9

[:SOURce]:INTerference:SINUsoidal Subnode

This subnode controls the configuration when AWG is used as sinusoidal interference source. It has the following SCPI structure:

```
[:SOURce]
└:INTerference
  └:SINusoidal
    ├:FRQuency[?]
    ├:AMPLitude{1:2}[?]
    ├:PHASe{1:2}[?]
    ├:TFRequency{1:2}[?]
    ├:TONemode{1:2}[?]
    └:STATe{1:2}[?]
```

This subnode has the following commands:

Table 54

Name	Description under
[:SOURce]INTerference:SINusoidal:STATe{1:2}[?]	[:SOURce]INTerference:SINusoidal:STATe{1:2}[?] on page 252
[:SOURce]INTerference:SINusoidal:TONemode{1:2}[?]	[:SOURce]:INTerference:SINusoidal:TONemode{1:2}[?] on page 252
[:SOURce]INTerference:SINusoidal:FREQuency{1:2}[?]	[:SOURce]:INTerference:SINusoidal:FREQuency{1:2}[?] on page 253
[:SOURce]INTerference:SINusoidal:AMPLitude{1:2}[?]	[:SOURce]:INTerference:SINusoidal:AMPLitude{1:2}[?] on page 253
[:SOURce]INTerference:SINusoidal:PHASe{1:2}[?]	[:SOURce]:INTerference:SINusoidal:PHASe{1:2}[?] on page 254
[:SOURce]INTerference:SINusoidal:TFRequency{1:2}[?]	[:SOURce]:INTerference:SINusoidal:TFRequency{1:2}[?] (Obsolete) on page 254

[:SOURce]INTerference:SINusoidal:STATe{1:2}[?]

Syntax

[:SOURce]:INTerference:SINusoidal:STATe{1:2} 'identifier', <ON | OFF | 1 | 0>

[:SOURce]:INTerference:SINusoidal:STATe{1:2}? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

<ON | OFF | 1 | 0>: Switch the SI {1:2} state ON or OFF.

Description

This command switches the state ON or OFF.

This SCPI is applicable for M8194A, M8195A, M8196A, M8053A, and M8054A.

Example

The following example shows how to use this command:

:SOUR:INT:SIN:STAT 'M1.DataOut1',ON

:SOUR:INT:SIN:STAT? 'M1.DataOut1'

ON

[:SOURce]:INTerference:SINusoidal:TONemode{1:2}[?]

Syntax

[:SOURce]:INTerference:SINusoidal:TONemode{1:2} 'identifier', COMMon | DIFFerential

[:SOURce]:INTerference:SINusoidal:TONemode{1:2}? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

Description

This command can be used to set the Tone Mode 1 and Tone Mode 2 to either Common mode or Differential mode.

For M8053A, this option is only valid if channel coupling is enabled in the M8053A user interface.

This SCPI is applicable for M8194A, M8195A, M8196A, M8053A and M8054A.

Example

The following example shows how to use this command to set the Tone Mode 1 to Differential Mode:

:SOUR:INT:SIN:TON1 'M1.DataOut1',DIFFerential

:SOUR:INT:SIN:TON1? 'M1.DataOut1'

DIFFerential

NOTE

The following modes changes are done:

- OFF: Not available (Obsolete)

If user sets ToneMode (SI1/SI2) to OFF in channel coupling mode, the respective State will be disabled.

[:SOURce]:INTerference:SINusoidal:FREQuency{1:2}[?]

Syntax	[:SOURce]:INTerference:SINusoidal:FREQuency{1:2} 'identifier', <NRf> [:SOURce]:INTerference:SINusoidal:FREQuency{1:2}? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' or 'M*.DataOut2'
Description	This command sets the sinusoidal interference frequency in Hz. This SCPI is applicable for M8194A, M8195A, M8196A, M8053A, and M8054A.
Example	:INT:SIN:FREQ1 'M2.DataOut1',1e9

[:SOURce]:INTerference:SINusoidal:AMPLitude{1:2}[?]

Syntax	[:SOURce]:INTerference:SINusoidal:AMPLitude{1:2} 'identifier', <NRf> [:SOURce]:INTerference:SINusoidal:AMPLitude{1:2}? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'
Description	This command can be used to set the Multitone Amplitude 1 and Multitone Amplitude 2 parameters value. This SCPI is applicable for M8194A, M8195A, M8196A, M8053A, and M8054A.
Example	The following example shows how to use this command to set the Multitone Amplitude 1 parameter value: :SOUR:INT:SIN:AMPL1 'M1.DataOut1',0.2 :SOUR:INT:SIN:AMPL1? 'M1.DataOut1' 0.2

[:SOURce]:INTerference:SINusoidal:PHASe{1:2}[?]

Syntax
[:SOURce]:INTerference:SINusoidal:PHASe{1:2} 'identifier', <NRf>
[:SOURce:]INTerference:SINusoidal:PHASe{1:2}? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

Description
This command can be used to set the Multitone Phase 1 and Phase 2 parameters value.

This SCPI is applicable for M8194A, M8195A, M8196A, M8053A and M8054A.

Example
The following example shows how to use this command to set the Multitone Phase 1 parameter value:

:SOUR:INT:SIN:PHAS1 'M1.DataOut1',180

:SOUR:INT:SIN:PHAS1? 'M1.DataOut1'

180

[:SOURce]:INTerference:SINusoidal:TFRequency{1:2}[?] (Obsolete)

NOTE

As this command is obsolete now, use command [:SOURce]:INTerference:SINusoidal:FREQuency{1:2}[?] for this.

For more information, refer to [:SOURce]:INTerference:SINusoidal:FREQuency{1:2}[?] **on page 253.**

Syntax
[:SOURce]:INTerference:SINusoidal:TFRequency{1:2} 'identifier', <NRf>
[:SOURce]:INTerference:SINusoidal:TFRequency{1:2}? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

Description
This command can be used to set the Multitone Frequency 1 and Multitone Frequency 2 parameters value.

This SCPI is applicable for M8194A, M8195A, M8196A, and M8054A.

Example
The following example shows how to use this command to set the Multitone Frequency 1 parameter value:

:SOUR:INT:SIN:TFR1 'M1.DataOut1',100000000

:SOUR:INT:SIN:TFR1? 'M1.DataOut1' -> 100000000

[: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 applicable for M8062A.

Example

:CONF:MINT 'M2.MuxMode', BOTH

[:SOURce]:MODulated:CLOCk Subnode

This subnode controls the configuration when the M8195A AWG is used as Modulated Clock and Interference Source.

It has the following SCPI structure:

```
[:SOURce]
└:MODulated
  └:CLOCk
    ├[:STATe][?]
    ├:SHAPe[?]
    ├:SSCProfile
    │└:TRIangular
    │  ├:STATe[?]
    │  └:ACTTime[?]
    ├:TP[?]
    ├:CFRequency[?]
    ├:SSCFrequency[?]
    ├:DEViation[?]
    ├:MINDeviation[?]
    ├:INITdeviation [?]
    ├:OVERshoot [?]
    ├:DELTa1 [?]
    ├:DELTa2 [?]
    ├:LOOP
    │└[:STATe][?]
    ├:CUSTom
    │├:SEGMents[?]
    │├:ONCE*[?]
    │└:FILE*[?]
    ├:EVENt
    │└:ADVance
    └:DELAy[?]
```

This subnode has the following commands:

Table 55

Name	Description under
[:SOURce]:MODulated:CLOCk[:STATe][?]	[:SOURce]:MODulated:CLOCk[:STATe][?] on page 258
[:SOURce]:MODulated:CLOCk:SHAPe[?]	[:SOURce]:MODulated:CLOCk:SHAPe[?] on page 258
[:SOURce]:MODulated:CLOCk:SSCProfile:TRIangular:STATe[?]	[:SOURce]:MODulated:CLOCk:SSCProfile:TRIangular:STATe[?] on page 259
[:SOURce]:MODulated:CLOCk:SSCProfile:TRIangular:STATe[?]	[:SOURce]:MODulated:CLOCk:SSCProfile:TRIangular:ACTTime[?] on page 259
[:SOURce]:MODulated:CLOCk:TP[?]	[:SOURce]:MODulated:CLOCk:TP[?] on page 260
[:SOURce]:MODulated:CLOCk:CFRequency[?]	[:SOURce]:MODulated:CLOCk:CFRequency[?] on page 260
[:SOURce]:MODulated:CLOCk:SSCFrequency[?]	[:SOURce]:MODulated:CLOCk:SSCFrequency[?] on page 260
[:SOURce]:MODulated:CLOCk:DEViation[?]	[:SOURce]:MODulated:CLOCk:DEViation[?] on page 261
[:SOURce]:MODulated:CLOCk:MINDeviation[?]	[:SOURce]:MODulated:CLOCk:MINDeviation[?] on page 261
[:SOURce]:MODulated:CLOCk:INITdeviation[?]	[:SOURce]:MODulated:CLOCk:INITdeviation[?] on page 262
[:SOURce]:MODulated:CLOCk:OVERshoot[?]	[:SOURce]:MODulated:CLOCk:OVERshoot[?] on page 262
[:SOURce]:MODulated:CLOCk:DELTa1[?]	[:SOURce]:MODulated:CLOCk:DELTa1[?] on page 262
[:SOURce]:MODulated:CLOCk:DELTa2[?]	[:SOURce]:MODulated:CLOCk:DELTa2[?] on page 263
[:SOURce]:MODulated:CLOCk:LOOP[:STATe][?]	[:SOURce]:MODulated:CLOCk:LOOP[:STATe][?] on page 263
[:SOURce]:MODulated:CLOCk:CUSTom:SEGMents[?]	[:SOURce]:MODulated:CLOCk:CUSTom:SEGMents[?] on page 264
[:SOURce]:MODulated:CLOCk:CUSTom:ONCE*[?]	[:SOURce]:MODulated:CLOCk:CUSTom:ONCE*[?] on page 264
[:SOURce]:MODulated:CLOCk:CUSTom:FILE*[?]	[:SOURce]:MODulated:CLOCk:CUSTom:FILE*[?] on page 265
[:SOURce]:MODulated:CLOCk:EVENt:ADVance	[:SOURce]:MODulated:CLOCk:EVENt:ADVance on page 265
[:SOURce]:MODulated:CLOCk:DELAy[?]	[:SOURce]:MODulated:CLOCk:DELAy[?] on page 265

[:SOURce]:MODulated:CLOCk[:STATe][?]

Syntax	[:SOURce]:MODulated:CLOCk[:STATe] 'identifier', ON | OFF | 1 | 0 [:SOURce]:MODulated:CLOCk[:STATe]? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' ON | OFF | 1 | 0
Description	This command enables or disables the modulated reference clock generation. This query returns the current settings. This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:STAT 'M1.DataOut1', ON :SOUR:MOD:CLOC:STAT? 'M1.DataOut1' -> 1

[:SOURce]:MODulated:CLOCk:SHAPe[?]

Syntax	[:SOURce]:MODulated:CLOCk:SHAPe 'identifier', USBG2|USBG3|USBG4|RBR|HBR|HBR2|HBR3|UBR10|UHBR13|UHBR20|USER|CUSTom [:SOURce]:MODulated:CLOCk:SHAPe? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' USBG2|USBG3|USBG4|RBR|HBR|HBR2|HBR3|UBR10|UHBR13|UHBR20|USER|CUSTom
Description	This command sets the deviation shape to be used for the reference clock. This query returns the current settings. This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:SHAP 'M1.DataOut1', USBG2 :SOUR:MOD:CLOC:SHAP? 'M1.DataOut1' -> USBG2

[:SOURce]:MODulated:CLOCk:SSCProfile:TRIangular:STATe[?]

Syntax	[:SOURce]:MODulated:CLOCk:SSCProfile:TRIangular:STATe 'identifier', 0|1|ON|OFF [:SOURce]:MODulated:CLOCk:SSCProfile:TRIangular:STATe? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' 0|1|ON|OFF
Description	This command enables and disables the triangular SSC modulation after the clock switch. This query returns the current settings. This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:SSC:TRI:STAT 'M1.DataOut1', ON :SOUR:MOD:CLOC:SSC:TRI:STAT? 'M1.DataOut1' -> 1

[:SOURce]:MODulated:CLOCk:SSCProfile:TRIangular:ACTTime[?]

Syntax	[:SOURce]:MODulated:CLOCk:SSCProfile:TRIangular:ACTTime 'identifier', <Activation Time> [:SOURce]:MODulated:CLOCk:SSCProfile:TRIangular:ACTTime? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' <Activation Time>: 0 to 20 us
Description	This command sets the activation time before the SSC is enabled. The activation time range is 0 to 20 us. This allows delaying the SSC profile activation by specified amount of time. Currently the SSC is enabled immediately. This query returns the current settings. This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:SSC:TRI:ACTT 'M1.DataOut1', 10 :SOUR:MOD:CLOC:SSC:TRI:ACTT? 'M1.DataOut1' -> 10

[:SOURce]:MODulated:CLOCk:TP[?]

Syntax	[:SOURce]:MODulated:CLOCk:TP 'identifier', TP3P | TP3 [:SOURce]:MODulated:CLOCk:TP? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' TP3P | TP3
Description	This command is used for selecting the test point at the receiver. This query returns the current settings. This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:TP 'M1.DataOut1', TP3P :SOUR:MOD:CLOC:TP? 'M1.DataOut1' -> TP3P

[:SOURce]:MODulated:CLOCk:CFRequency[?]

Syntax	[:SOURce]:MODulated:CLOCk:CFRequency 'identifier', <NRf> [:SOURce]:MODulated:CLOCk:CFRequency? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' <NRf>: Carrier Frequency in Hz
Description	This command sets the Carrier Frequency of the Modulated Reference Clock in Hz. This query returns the current settings. This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:CFR 'M1.DataOut1', 100000000 :SOUR:MOD:CLOC:CFR? 'M1.DataOut1' -> 100000000

[:SOURce]:MODulated:CLOCk:SSCFrequency[?]

Syntax	[:SOURce]:MODulated:CLOCk:SSCFrequency 'identifier', <NRf> [:SOURce]:MODulated:CLOCk:SSCFrequency? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' <NRf>: SSC Frequency in Hz

Description This command sets the SSC Frequency of the Modulated Reference Clock in Hz.

This query returns the current settings.

This SCPI is only applicable for M8195A AWG.

Example :SOUR:MOD:CLOC:SSCF 'M1.DataOut1', 32000

:SOUR:MOD:CLOC:SSCF? 'M1.DataOut1' -> 32000

[:SOURce]:MODulated:CLOCk:DEViation[?]

Syntax [:SOURce]:MODulated:CLOCk:DEViation 'identifier', <NRf>

[:SOURce]:MODulated:CLOCk:DEViation? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1'

<NRf>: SSC frequency in ppm

Description This command sets the SSC frequency deviation in ppm.

This query returns the current settings.

This SCPI is only applicable for M8195A AWG.

Example :SOUR:MOD:CLOC:DEV 'M1.DataOut1', 5600

:SOUR:MOD:CLOC:DEV? 'M1.DataOut1' -> 5600

[:SOURce]:MODulated:CLOCk:MINDeviation[?]

Syntax [:SOURce]:MODulated:CLOCk:MINDeviation 'identifier', <NRf>

[:SOURce]:MODulated:CLOCk:MINDeviation? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1'

<NRf>: SSC frequency in ppm

Description This command sets the minimum SSC frequency deviation in ppm.

This query returns the current settings.

This SCPI is only applicable for M8195A AWG.

Example :SOUR:MOD:CLOC:MIND 'M1.DataOut1', -5300

:SOUR:MOD:CLOC:MIND? 'M1.DataOut1' -> -5300

[:SOURce]:MODulated:CLOCk:INITdeviation[?]

Syntax	[:SOURce]:MODulated:CLOCk:INITdeviation 'identifier', <NRf>
	[:SOURce]:MODulated:CLOCk:INITdeviation? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1'
	<NRf>: SSC frequency in ppm
Description	This command sets the initial non-modulated SSC frequency deviation in ppm.
	This query returns the current settings.
	This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:STAT 'M1.DataOut1', ON
	:SOUR:MOD:CLOC:STAT? 'M1.DataOut1' -> 1

[:SOURce]:MODulated:CLOCk:OVERshoot[?]

Syntax	[:SOURce]:MODulated:CLOCk:OVERshoot 'identifier', <NRf>
	[:SOURce]:MODulated:CLOCk:OVERshoot? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1'
	<NRf>: Frequency in ppm
Description	This command sets the maximum frequency deviation offset in ppm.
	This query returns the current settings.
	This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:OVER 'M1.DataOut1', 1600
	:SOUR:MOD:CLOC:OVER? 'M1.DataOut1' -> 1600

[:SOURce]:MODulated:CLOCk:DELTa1[?]

Syntax	[:SOURce]:MODulated:CLOCk:DELTa1 'identifier', <NRf>
	[:SOURce]:MODulated:CLOCk:DELTa1? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1'
	<NRf>: Frequency in ppm
Description	This command sets the frequency deviation over 200 ns measurement

window in ppm.

This query returns the current settings.

This SCPI is only applicable for M8195A AWG.

Example :SOUR:MOD:CLOC:DELT1 'M1.DataOut1', -1400

:SOUR:MOD:CLOC:DELT1? 'M1.DataOut1' -> -1400

[:SOURce]:MODulated:CLOCk:DELTa2[?]

Syntax [:SOURce]:MODulated:CLOCk:DELTa2 'identifier', <NRf>

[:SOURce]:MODulated:CLOCk:DELTa2? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1'

<NRf>: Frequency in ppm

Description This command sets the frequency deviation over 1 us measurement window in Hz.

This query returns the current settings.

This SCPI is only applicable for M8195A AWG.

Example :SOUR:MOD:CLOC:DELT2 'M1.DataOut1', -2200

:SOUR:MOD:CLOC:DELT2? 'M1.DataOut1' -> -2200

[:SOURce]:MODulated:CLOCk:LOOP[:STATe][?]

Syntax [:SOURce]:MODulated:CLOCk:LOOP[:STATe] 'identifier', ON | OFF | 1 | 0

[:SOURce]:MODulated:CLOCk:LOOP[:STATe]? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1'

ON | OFF | 1 | 0

Description This command enables or disables the segment loop.

This query returns the current settings.

This SCPI is only applicable for M8195A AWG.

Example :SOUR:MOD:CLOC:LOOP:STAT 'M1.DataOut1', ON

:SOUR:MOD:CLOC:LOOP:STAT? 'M1.DataOut1' -> ON

[:SOURce]:MODulated:CLOCk:CUSTom:SEGMents[?]

Syntax
[:SOURce]:MODulated:CLOCk:CUSTom:SEGMents 'identifier', <NRf>

[:SOURce]:MODulated:CLOCk:CUSTom:SEGMents? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1'

<NRf>: Number of segments

Description
This command sets the number of segments for Custom shape. The maximum number of segments is 4.

This query returns the current settings.

This SCPI is only applicable for M8195A AWG.

Example
:SOUR:MOD:CLOC:CUST:SEGM 'M1.DataOut1', 2

:SOUR:MOD:CLOC:CUST:SEGM? 'M1.DataOut1' -> 2

[:SOURce]:MODulated:CLOCk:CUSTom:ONCE*[?]

Syntax
[:SOURce]:MODulated:CLOCk:CUSTom:ONCE* 'identifier', ON | OFF | 1 | 0

[:SOURce]:MODulated:CLOCk:CUSTom:ONCE*? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1'

ON | OFF | 1 | 0

Description
This command enables or disables the 'Play Once' option for a segment for Custom shape. Use numeric suffix 1 to 4 to select the segment for which the option needs to be set. When this option is enabled the segment is played only once.

This query returns the current settings.

This SCPI is only applicable for M8195A AWG.

Example
:SOUR:MOD:CLOC:CUST:ONCE1 'M1.DataOut1', 0

:SOUR:MOD:CLOC:CUST:ONCE1? 'M1.DataOut1' -> 0

[:SOURce]:MODulated:CLOCk:CUSTom:FILE*[?]

Syntax	[:SOURce]:MODulated:CLOCk:CUSTom:FILE* 'identifier', "Filename"
	[:SOURce]:MODulated:CLOCk:CUSTom:FILE*? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1'
	"Filename": File name
Description	This command sets the deviation file for a segment for Custom shape. Use numeric suffix 1 to 4 to select the segment for which the file name needs to be set.
	This query returns the current settings.
	This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:CUST:FILE1 'M1.DataOut1', "Factory/shape1.txt"

[:SOURce]:MODulated:CLOCk:EVENt:ADVance

Syntax	[:SOURce]:MODulated:CLOCk:EVENt:ADVance 'identifier'
Input Parameters	'identifier': 'M*.DataOut1'
Description	This command sends force event for advancing from one segment to the next segment.
	This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:EVE:ADV 'M1.DataOut1'

[:SOURce]:MODulated:CLOCk:DELAy[?]

Syntax	[:SOURce]:MODulated:CLOCk:DELAy 'identifier', <NRf>
	[:SOURce]:MODulated:CLOCk:DELAy? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1'
Description	This command sets the initial delay for the clock switch segment.
	This SCPI is only applicable for M8195A AWG.
Example	:SOUR:MOD:CLOC:DELA 'M1.DataOut1', 2.00000000000000E-06
	:SOUR:MOD:CLOC:DELA? 'M1.DataOut1' -> 2E-06

[:SOURce]:SSCLocking:SEGMented Subnode

This subnode controls the SSC source in segmented arbitrary mode.

It has the following SCPI structure:

```
[:SOURce]
└─:SSCLocking
  └─:SEGMented
    ├─:ADVance
    ├─:CUSTom
    │ ├─:FILE1|2|3|4[?]
    │ ├─:ONCE1|2|3|4[?]
    │ └─:SEGMents[?]
    ├─:DELTa1|2[?]
    ├─:DEViation[?]
    ├─:INTDeviation[?]
    ├─:MINDeviation[?]
    ├─:LOOP:STATe[?]
    ├─:OVERshoot [?]
    ├─:RESTart
    ├─:SEGMent?
    ├─:SHAPe[?]
    ├─:SSCFrequency[?]
    ├─:SSCProfile
    │ └─:TRIangular
    │   ├─:ACTTime[?]
    │   └─:STATe[?]
    ├─:TP[?]
    └─:TRIGger
      ├─:SEGMent1|2|3|4:STATe[?]
      └─:STATe[?]
```

This subnode has the following commands:

Table 56

Name	Description under
[:SOURce]:SSCLocking:SEGMented:ADVance	[:SOURce]:SSCLocking:SEGMented:ADVance on page 268
[:SOURce]:SSCLocking:SEGMented:CUSTom:FILE1|2|3|4[?]	[:SOURce]:SSCLocking:SEGMented:CUSTom:FILE1|2|3|4[?] on page 268
[:SOURce]:SSCLocking:SEGMented:CUSTom:ONCE1|2|3|4[?]	[:SOURce]:SSCLocking:SEGMented:CUSTom:ONCE1|2|3|4[?] on page 269
[:SOURce]:SSCLocking:SEGMented:CUSTom:SEGMents[?]	[:SOURce]:SSCLocking:SEGMented:CUSTom:SEGMents[?] on page 269
[:SOURce]:SSCLocking:SEGMented:DELTa1|2[?]	[:SOURce]:SSCLocking:SEGMented:DELTa1|2[?] on page 270
[:SOURce]:SSCLocking:SEGMented:DEViation[?]	[:SOURce]:SSCLocking:SEGMented:DEViation[?] on page 270
[:SOURce]:SSCLocking:SEGMented:INTDeviation[?]	[:SOURce]:SSCLocking:SEGMented:INTDeviation[?] on page 271
[:SOURce]:SSCLocking:SEGMented:MINDeviation[?]	[:SOURce]:SSCLocking:SEGMented:MINDeviation[?] on page 271
[:SOURce]:SSCLocking:SEGMented:LOOP:STATe[?]	[:SOURce]:SSCLocking:SEGMented:LOOP:STATe[?] on page 272
[:SOURce]:SSCLocking:SEGMented:OVERshoot[?]	[:SOURce]:SSCLocking:SEGMented:OVERshoot[?] on page 272
[:SOURce]:SSCLocking:SEGMented:RESTart	[:SOURce]:SSCLocking:SEGMented:RESTart on page 273
[:SOURce]:SSCLocking:SEGMented:SEGMent?	[:SOURce]:SSCLocking:SEGMented:SEGMent? on page 273
[:SOURce]:SSCLocking:SEGMented:SHAPe[?]	[:SOURce]:SSCLocking:SEGMented:SHAPe[?] on page 273
[:SOURce]:SSCLocking:SEGMented:SSCFrequency[?]	[:SOURce]:SSCLocking:SEGMented:SSCFrequency[?] on page 274
[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:ACTTime[?]	[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:ACTTime[?] on page 274
[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:STATe[?]	[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:STATe[?] on page 275
[:SOURce]:SSCLocking:SEGMented:TP[?]	[:SOURce]:SSCLocking:SEGMented:TP[?] on page 275
[:SOURce]:SSCLocking:SEGMented:TRIGger:SEGMent1|2|3|4:STATe[?]	[:SOURce]:SSCLocking:SEGMented:TRIGger:SEGMent1|2|3|4:STATe[?] on page 276
[:SOURce]:SSCLocking:SEGMented:TRIGger:STATe[?]	[:SOURce]:SSCLocking:SEGMented:TRIGger:STATe[?] on page 276

[:SOURce]:SSCLocking:SEGMented:ADVance

Syntax	[:SOURce]:SSCLocking:SEGMented:ADVance 'identifier'
Input Parameters	'identifier': 'M*.ClkGen'
Description	This command sends force event for advancing from one segment to the next segment in the sequence. This SCPI is applicable for M8009A clock module.
Example	:SOUR:SSCL:SEGM:ADV 'M1.ClkGen'

[:SOURce]:SSCLocking:SEGMented:CUSTom:FILE1|2|3|4[?]

Syntax	[:SOURce]:SSCLocking:SEGMented:CUSTom:FILE1|2|3|4 'identifier', "Filename" [:SOURce]:SSCLocking:SEGMented:CUSTom:FILE1|2|3|4? 'identifier'
Input Parameters	'identifier': 'M*.ClkGen' "Filename": File name
Description	This command sets the deviation waveform file to be loaded into this segment. The file has to contain one decimal number per line that specifies deviation in ppm. The maximum deviation range is ±10000 ppm. The deviation points defined by the file will be re-sampled to fill the 32768 samples log segment. Use numeric suffix 1 to 4 to select the segment for which the file name needs to be set. This query returns the current settings. This SCPI is applicable for M8009A clock module.
Example	:SOUR:SSCL:SEGM:CUST:FILE1 'M1.ClkGen', "Factory/shape1.txt" :SOUR:SSCL:SEGM:CUST:FILE1? 'M1.ClkGen' -> "Factory/shape1.txt

[:SOURce]:SSCLocking:SEGMented:CUSTom:ONCE1|2|3|4[?]

Syntax: [:SOURce]:SSCLocking:SEGMented:CUSTom:ONCE1|2|3|4 'identifier', ON | OFF | 1 | 0

[:SOURce]:SSCLocking:SEGMented:CUSTom:ONCE1|2|3|4? 'identifier'

Input Parameters: 'identifier': 'M*.ClkGen'

ON | OFF | 1 | 0

Description: This command enables or disables the 'Play Once' option for a segment for Custom shape. Use numeric suffix 1 to 4 to select the segment for which the option needs to be set. When this option is enabled the segment is played only once and the segmented SSC automatically advances to the next one.

If disabled, this segment will be played in a loop until clicking the “Advance Segment” button. Only then, the segmented SSC will advance to the next one.

This query returns the current settings.

This SCPI is applicable for M8009A clock module.

Example: :SOUR:SSCL:SEGM:CUST:ONCE1 'M1.ClkGen', ON

:SOUR:SSCL:SEGM:CUST:ONCE1? 'M1.ClkGen' -> 1

[:SOURce]:SSCLocking:SEGMented:CUSTom:SEGMents[?]

Syntax: [:SOURce]:SSCLocking:SEGMented:CUSTom:SEGMents 'identifier', <NRf>

[:SOURce]:SSCLocking:SEGMented:CUSTom:SEGMents? 'identifier'

Input Parameters: 'identifier': 'M*.ClkGen'

<NRf>: Number of segments

Description: This command sets the number of segments for Custom shape in the segmented SSC profile. The maximum number of segments is 4.

This query returns the current settings.

This SCPI is applicable for M8009A clock module.

Example: :SOUR:SSCL:SEGM:CUST:SEGM 'M1.ClkGen', 2

:SOUR:SSCL:SEGM:CUST:SEGM? 'M1.ClkGen' -> 2

[:SOURce]:SSCLocking:SEGMented:DELTa1|2[?]

Syntax	[:SOURce]:SSCLocking:SEGMented:DELTa1|2 'identifier', <NRf> [:SOURce]:SSCLocking:SEGMented:DELTa1|2? 'identifier'
Input Parameters	'identifier': 'M*.ClkGen' <NRf>: Deviation in ppm
Description	This command sets the relative deviation in ppm after the overshoot. Delta 1 is 200 ns and Delta 2 is 1000 ns. The deviation is relative to the overshoot deviation. This SCPI command becomes invalid if the shape is neither selected as “User Defined” nor “Custom”. In such case, this SCPI behaves as a query only. This query returns the current settings. This SCPI is applicable for M8009A clock module.
Example	:SOUR:SSCL:SEGM:DELT1 'M1.ClkGen', -1400 :SOUR:SSCL:SEGM:DELT1? 'M1.ClkGen' -> -1400

[:SOURce]:SSCLocking:SEGMented:DEViation[?]

Syntax	[:SOURce]:SSCLocking:SEGMented:DEViation 'identifier', <NRf> [:SOURce]:SSCLocking:SEGMented:DEViation? 'identifier'
Input Parameters	'identifier': 'M*.ClkGen' <NRf>: SSC deviation in ppm
Description	This command sets the peak-to-peak deviation of the triangular modulation in ppm. This SCPI command becomes invalid if the shape is neither selected as “User Defined” nor “Custom”. In such case, this SCPI behaves as a query only. This query returns the current settings. This SCPI is applicable for M8009A clock module.
Example	:SOUR:SSCL:SEGM:DEV 'M1.ClkGen', 5600 :SOUR:SSCL:SEGM:DEV? 'M1.ClkGen' -> 5600

[:SOURce]:SSCLocking:SEGMented:INTDeviation[?]

Syntax	[:SOURce]:SSCLocking:SEGMented:INTDeviation 'identifier', <NRf>
	[:SOURce]:SSCLocking:SEGMented:DEViation? 'identifier'
Input Parameters	'identifier': 'M*.ClkGen'
	<NRf>: SSC deviation in ppm
Description	This command sets the initial non-modulated deviation in ppm. This SCPI command becomes invalid if the shape is neither selected as “User Defined” nor “Custom”. In such case, this SCPI behaves as a query only.
	This query returns the current settings.
	This SCPI is applicable for M8009A clock module.
Example	:SOUR:SSCL:SEGM:INTD 'M1.ClkGen', 300
	:SOUR:SSCL:SEGM:INTD? 'M1.ClkGen' -> 300

[:SOURce]:SSCLocking:SEGMented:MINDeviation[?]

Syntax	[:SOURce]:SSCLocking:SEGMented:MINDeviation 'identifier', <NRf>
	[:SOURce]:SSCLocking:SEGMented:MINDeviation? 'identifier'
Input Parameters	'identifier': 'M*.ClkGen'
	<NRf>: SSC deviation in ppm
Description	This command sets the minimum deviation of the triangular modulation in ppm. This SCPI command becomes invalid if the shape is neither selected as “User Defined” nor “Custom”. In such case, this SCPI behaves as a query only.
	This query returns the current settings.
	This SCPI is applicable for M8009A clock module.
Example	:SOUR:SSCL:SEGM:MIND 'M1.ClkGen', -5300
	:SOUR:SSCL:SEGM:MIND? 'M1.ClkGen' -> -5300

[:SOURce]:SSCLocking:SEGMented:LOOP:STATe[?]

Syntax [:SOURce]:SSCLocking:SEGMented:LOOP[:STATe] 'identifier', ON | OFF | 1 | 0

[:SOURce]:SSCLocking:SEGMented:LOOP[:STATe]? 'identifier'

Input Parameters 'identifier': 'M*.ClkGen'

ON | OFF | 1 | 0

Description This command enables or disables the segment loop.

If enabled, all the three segments (initial deviation, clock switch, triangular modulation) will be played back continuously one after the other. The initial deviation and triangular modulation segments are repeated 10 times.

If disabled, the initial segment will be played repeatedly. Clicking the “Advance Segment” button plays the clock segment once and then repeats the triangular modulation segment.

Clicking the “Advance Segment” button a second time switches back to the initial deviation segment.

This query returns the current settings.

This SCPI is applicable for M8009A clock module.

Example :SOUR:SSCL:SEGM:LOOP:STAT 'M1.ClkGen', ON

:SOUR:SSCL:SEGM:LOOP:STAT? 'M1.ClkGen' -> 1

[:SOURce]:SSCLocking:SEGMented:OVERshoot[?]

Syntax [:SOURce]:SSCLocking:SEGMented:OVERshoot 'identifier', <NRf>

[:SOURce]:SSCLocking:SEGMented:OVERshoot? 'identifier'

Input Parameters 'identifier': 'M*.ClkGen'

<NRf>: Frequency in ppm

Description This command sets the absolute peak deviation of the overshoot in ppm. This SCPI command becomes invalid if the shape is neither selected as “User Defined” nor “Custom”. In such case, this SCPI behaves as a query only.

This query returns the current settings.

This SCPI is applicable for M8009A clock module.

Example :SOUR:SSCL:SEGM:OVER 'M1.ClkGen', 1600

:SOUR:SSCL:SEGM:OVER? 'M1.ClkGen' -> 1600

[:SOURce]:SSCLocking:SEGMented:RESTart

Syntax [:SOURce]:SSCLocking:SEGMented:RESTart 'identifier',

Input Parameters 'identifier': 'M*.ClkGen'

Description This command stops the segmented SSC and restarts it at the first segment.

This SCPI is applicable for M8009A clock module.

Example :SOUR:SSCL:SEGM:REST 'M1.ClkGen'

[:SOURce]:SSCLocking:SEGMented:SEGMent?

Syntax [:SOURce]:SSCLocking:SEGMented:SEGMent? 'identifier'

Input Parameters 'identifier': 'M*.ClkGen'

Description This query returns the segment number that is being played by the segmented SSC.

This SCPI is applicable for M8009A clock module.

Example :SOUR:SSCL:SEGM:SEGM? 'M1.ClkGen' -> 1

[:SOURce]:SSCLocking:SEGMented:SHAPe[?]

Syntax [:SOURce]:SSCLocking:SEGMented:SHAPe 'identifier', USB10|USB20|USB40|RBR|HBR|HBR2|HBR3|UBR10|UHBR13|UHBR20|USER|CUSTom

[:SOURce]:SSCLocking:SEGMented:SHAPe? 'identifier'

Input Parameters 'identifier': 'M*.ClkGen'

USB10|USB20|USB40|RBR|HBR|HBR2|HBR3|UBR10|UHBR13|UHBR20|USER|CUSTom

Description This command sets the deviation shape to be used for the reference clock.

This query returns the current settings.

This SCPI is applicable for M8009A clock module.

Example :SOUR:SSCL:SEGM:SHAP 'M1.ClkGen', USB10

:SOUR:SSCL:SEGM:SHAP? 'M1.ClkGen' -> USB10

[:SOURce]:SSCLocking:SEGMented:SSCFrequency[?]

Syntax [:SOURce]:SSCLocking:SEGMented:SSCFrequency 'identifier', <NRf>

[:SOURce]:SSCLocking:SEGMented:SHAPe? 'identifier'

Input Parameters 'identifier': 'M*.ClkGen'

<NRf>: SSC Frequency

Description This command sets the frequency of the triangular modulation. In custom shape mode, it specifies the inverse of the duration of each segment. This SCPI command becomes invalid if the shape is neither selected as “User Defined” nor “Custom”. In such case, this SCPI behaves as a query only.

This query returns the current settings.

This SCPI is applicable for M8009A clock module.

Example :SOUR:SSCL:SEGM:SSCF 'M1.ClkGen', 3200

:SOUR:SSCL:SEGM:SSCF? 'M1.ClkGen' -> 3200

[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:ACTTime[?]

Syntax [:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:ACTTime 'identifier', <Activation Time>

[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:ACTTime? 'identifier'

Input Parameters 'identifier': 'M*.ClkGen'

<Activation Time>: 0 to 20 us

Description This command sets the activation time before enabling the triangular modulation. The activation time range is 0 to 20 us. This allows delaying the SSC profile activation by specified amount of time.

This query returns the current settings.

This SCPI is applicable for M8009A clock module.

Example :SOUR:SSCL:SEGM:SSCP:TRI:ACTT 'M1.ClkGen', 0

:SOUR:SSCL:SEGM:SSCP:TRI:ACTT? 'M1.ClkGen' -> 0

[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:STATe[?]

Syntax [:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:STATe 'identifier', 0|1|ON|OFF

[:SOURce]:SSCLocking:SEGMented:SSCProfile:TRIangular:STATe? 'identifier'

Input Parameters 'identifier': 'M*.ClkGen'

0|1|ON|OFF

Description This command enables and disables the triangular SSC modulation after the clock switch.

This query returns the current settings.

This SCPI is applicable for M8009A clock module.

Example :SOUR:SSCL:SEGM:SSCP:TRI:STAT 'M1.ClkGen', 1

:SOUR:SSCL:SEGM:SSCP:TRI:STAT? 'M1.ClkGen' -> 1

[:SOURce]:SSCLocking:SEGMented:TP[?]

Syntax [:SOURce]:SSCLocking:SEGMented:TP 'identifier', TP3P | TP3

[:SOURce]:SSCLocking:SEGMented:TP? 'identifier'

Input Parameters 'identifier': 'M*.ClkGen'

TP3P | TP3

Description This command is used for selecting the test point at the receiver.

This query returns the current settings.

This SCPI is applicable for M8009A clock module.

Example :SOUR:SSCL:SEGM:TP 'M1.ClkGen', TP3P

:SOUR:SSCL:SEGM:TP? 'M1.ClkGen' -> TP3P

[:SOURce]:SSCLocking:SEGMented:TRIGger:SEGMent1|2|3|4:STATe[?]

Syntax	[:SOURce]:SSCLocking:SEGMented:TRIGger:SEGMent1|2|3|4:STATe 'identifier', 1|0|ON|OFF [:SOURce]:SSCLocking:SEGMented:TRIGger:SEGMent1|2|3|4:STATe? 'identifier'
Input Parameters	'identifier': 'M*.ClkGen' 1|0|ON|OFF
Description	This command enables or disables trigger on segment. If enabled, a trigger event is generated when the segmented SSC enters this segment. This query returns the current settings. This SCPI is applicable for M8009A clock module.
Example	:SOUR:SSCL:SEGM:TRIG:SEGM1 'M1.ClkGen', ON :SOUR:SSCL:SEGM:TRIG:SEGM2 'M1.ClkGen', ON :SOUR:SSCL:SEGM:TRIG:SEGM3 'M1.ClkGen', ON :SOUR:SSCL:SEGM:TRIG:SEGM4 'M1.ClkGen', ON :SOUR:SSCL:SEGM:TRIG:SEGM1? 'M1.ClkGen' -> 1

[:SOURce]:SSCLocking:SEGMented:TRIGger:STATe[?]

Syntax	[:SOURce]:SSCLocking:SEGMented:TRIGger:STATe 'identifier', 1|0|ON|OFF [:SOURce]:SSCLocking:SEGMented:TRIGger:STATe? 'identifier'
Input Parameters	'identifier': 'M*.ClkGen' 1|0|ON|OFF
Description	This command enables or disables the state of the segment trigger output. If enabled, the module outputs a 100 MHz square wave at the Sync Out C connector. A single 50 MHz pulse indicates a trigger event. In standard or user-defined shapes, a trigger event is generated at the clock switch event. For custom shapes, a trigger event is generated when entering a segment that has the trigger property set. This query returns the current settings. This SCPI is applicable for M8009A clock module.

Example :SOUR:SSCL:SEGM:TRIG:STAT 'M1.ClkGen', ON

:SOUR:SSCL:SEGM:TRIG:STAT? 'M1.ClkGen' -> 1

OUTPut Subsystem

The OUTPut subsystem controls the output ports of the pattern generator.

This subsystem has the following SCPI structure:

```
:OUTPut
├─:GLOBal
│ └─[:STATe][?]
├─[:STATe][?]
├─:CONFig
│ └─:SOURce[?]
├─:DRATe?
├─:POLarity[?]
├─:TVOLtage[?]
├─:TCONfig[?]
├─:COUPling[?]
├─:DIVider[?]
├─:FREQuency
│ └─:DIVider[?]
├─:CALibration
│ └─:SKEW[?]
├─:EIDLe
│ └─[:STATe][?]
├─:DATA
│ ├─:XOVer[?]
│ └─:F2Jitter[?]
│   └─:STATe[?]
├─:DAC
│ └─:RANDomizer[:STATe][?]
├─:DEEMphasis
│ └─ ...
├─:EINSertion
│ └─ ...
├─:TRIGger
│ └─ ...
├─:DEMBedding
│ └─ ...
├─:EMBedding
│ └─ ...
└─:ADJustment
  └─ ...
```

This subsystem has the following commands and subnodes:

Name	Description under
:GLOBal[:STATe][?]	:OUTPut:GLOBal[:STATe][?] on page 281
[:STATe][?]	:OUTPut[:STATe][?] on page 281
:CONFig:SOURce[?]	:OUTPut:CONFig:SOURce[?] on page 282
:DRATe?	:OUTPut:DRATe? on page 283
:POLarity[?]	:OUTPut:POLarity[?] on page 283
:TVOLtage[?]	:OUTPut:TVOLtage[?] on page 283
:TCONfig[?]	:OUTPut:TCONfig[?] on page 284
:COUPling[?]	:OUTPut:COUPling[?] on page 285
:DIVider[?]	:OUTPut:DIVider[?] on page 285
:CLock:DIVider[?]	:OUTPut:CLock:DIVider[?] on page 286
:FREQuency[?]	:OUTPut:FREQuency[?] on page 286
:FREQuency:DIVider[?]	:OUTPut:FREQuency:DIVider[?] on page 287
:CALibration:SKEW[?]	:OUTPut:CALibration:SKEW[?] on page 287
:EIDLe[:STATe][?]	:OUTPut:EIDLe[:STATe][?] on page 288
:DATA:XOVer[?]	:OUTPut:DATA:XOVer[?] on page 288
:DATA:F2Jitter[?]	:OUTPut:DATA:F2Jitter[?] on page 289
:DATA:F2Jitter:STATe[?]	:OUTPut:DATA:F2Jitter:STATe[?] on page 289
:DAC:RANDomizer[:STATe][?]	:OUTPut:DAC:RANDomizer[:STATe][?] on page 290
Subnodes	
:DEEMphasis	:OUTPut:DEEMphasis Subnode on page 291
:EINSertion	:OUTPut:EINSertion Subnode on page 305
:TRIGger	:OUTPut:TRIGger Subnode on page 310
:DEMBedding	:OUTPut:DEMBedding Subnode on page 312
:EMBedding	:OUTPut:EMBedding Subnode on page 316
:ADJustment	:OUTPut:ADJustment Subnode on page 318

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 57

License	Description	Modules
M8009A-062	Clock Generator two channel 60 GHz	M8009A
M8009A-0G6	Reference Clock Multiplier	M8009A
M8041A-0G4	Multi-tap Deemphasis, Module-wide License	M8041A
M8041A-G08	Generator one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8042A-0G4	Deemphasis	M8042A
M8042A-G32	Pattern generation up to 32 GBd for NRZ and PAM4	M8042A
M8045A-0G4	Deemphasis	M8045A
M8045A-G32	32 GBd Pattern Generator	M8045A
M8051A-0G4	Multi-tap Deemphasis, Module-wide License	M8051A
M8062A-0G4	Multi-tap Deemphasis License	M8062A
M8062A-G32	32 Gb/s Pattern generator front-end	M8062A
M8070ADVB	Advanced Measurement Package for M8000 Series BERT Test Solutions	M8045A
M8195A AWG license	Required for OUTPut:EMBedding subnode commands	M8195A

: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 affects all modules.

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*.DataOut3', 'M*.DataOut4'', M*.CtrlOut', 'M*.CtrlOutA', 'M*.DataOut', 'M*.SyncMrkA', 'M*.SyncMrkB' or 'M*.SampleMrk', M*.RefClkOut, M*.RefClkOut16G, M*.SampleClkOut, 'M*.ClkOut16G', 'M*.ChClkOut2', or 'M*.ClkOut32G'

<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 applicable for M8041A, M8042A, M8045A, M8047B, M8047C, M8051A, M8053A, M8062A, M8195A and M8009A.

Example

:OUTP 'M1.DataOut1', OFF

:OUTPut:MODE[?]

Syntax	:OUTPut:MODE 'identifier', <FCL|MCL>
	:OUTPut:MODE? 'identifier'
Input Parameters	'identifier': 'M1.ChClkOut2'
	FCLock: Forwarded clock
	MClock: Module clock
Description	This command sets the configuration mode to either forwarded clock or module clock.
	This query returns the present state.
	This SCPI is supported by “ChClkOut2” of the M8009A module.
Example	:OUTP:MODE 'M1.ChClkOut2', FCL
	:OUTP:MODE? 'M1.ChClkOut2' -> FCL

:OUTPut:CONFig:SOURce[?]

Syntax	:OUTPut:CONFig:SOURce <Identifier>, <CLEanclock|DATaclock>
	:OUTPut:CONFig:SOURce? <Identifier>
Input Parameters	'identifier': 'M1.ClkOut1' or 'M1.ClkOut2'
Return Range	CLE|DAT
Description	This command selects the signal source for the instrument’s output. This is only valid for certain clock outputs.
	This query returns the present state.
	This SCPI is supported by M8042A & M8045A.
Example	:OUTP:CONF:SOUR 'M1.ClkOut1',CLE

: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 applicable for M8041A, M8042A, M8051A, M8062A, M8194A, M8195A, M8196A, and M8045A.
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*.DataIn', '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 applicable for M8041A, M8042A, M8043A, M8051A, M8062A and M8045A.
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 applicable for M8041A, M8042A, M8051A, M8062A and M8195A.

Example :OUTP:TVOL 'M1.DataOut1', 700mV

:OUTPut:FAMPlitude[?]

Syntax :OUTPut:FAMPlitude 'identifier', <ON|OFF|1|0>

:OUTPut:FAMPlitude? 'identifier'

Input Parameters 'identifier': 'M*.DataOut*'

ON|OFF|1|0: Enables or disables fast amplitude change.

Return Range 1|0

Description This command enables or disables the amplitude change without waveform recalculation and download. This option can be enabled for new optimization and can be disabled to have existing programs behave exactly as now. You can use the download waveform option to download the waveform at any time.

This option is applicable when:

- Amplitude is above 80 mV.
- Only RI or SI (Single-tone) is enabled.

This option is available only when the **Interference Source** option is selected in the **Configuration** dialog box. This SCPI is applicable for M8194A, M8195A, M8196A and M8054A.

Example :OUTP:FAMP 'M3.DataOut1', ON

:OUTP:FAMP? 'M3.DataOut1' -> 1

: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 applicable for M8041A, M8042A, M8051A 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', 'M*.DataOut3', 'M*.DataOut4', 'M*.DataOut' or 'M*.SampleMrk'.
	<AC|DC>: Select output coupling.
Return Range	AC|DC
Description	This command selects DC or AC output coupling.
	This SCPI is applicable for M8041A, M8042A, M8045A, M8051A and M8062A.
Example	:OUTP:COUP 'M1.DataOut1', DC

:OUTPut:DIVider[?]

Syntax	:OUTPut:DIVider 'identifier', <DIV1...DIV640>
	:OUTPut:DIVider? 'identifier'
Input Parameters	'identifier' (for M8009A): 'M1.ClkOut', 'M1.ClkOut1', 'M1.ClkOut2' or 'M1.ClkOut32G'
	Specify a divider (for M8009A only): DIV1 | DIV2 | DIV4 | DIV8 | DIV16 |
	DIV32 | DIV64
Return Range	DIV1 | DIV2 | DIV4 | DIV8 | DIV16 | DIV32 | DIV64
Description	This command selects a specific clock divider.
	Please note that the divider availability depends on identifier, product number and currently used symbol rate.
	This SCPI is applicable for M8041A, M8045A and M8009A.
Example	:OUTP:DIV 'M1.ClkOut',DIV1

:OUTPut:CLock:DIVider[?]

Syntax	:OUTPut:CLock:DIVider 'identifier', <Value>
	:OUTPut:CLock:DIVider? 'identifier'
Input Parameters	'identifier': 'M1.ClkOut16G', 'M*.TrigOut'
	<Value>: Between 2 to 512
Description	This command sets the divider for subrate clock generation.
	This SCPI is applicable for the “Clock Out 16G” port of M8009A.
	This SCPI is applicable for the “Trig Out” port of M8053A.
Example	:OUTP:CL:DIV 'M1.ClkOut16G', 2
	:OUTP:CL:DIV? 'M1.ClkOut16G' -> 2

:OUTPut:FREQuency[?]

Syntax	:OUTPut:FREQuency 'identifier', <NRf>
	:OUTPut:FREQuency? 'identifier'
Input Parameters	'identifier': 'M*.RefClkOut', 'M*.RefClkOut16G', 'M*.ClnClkOut', 'M*.ClkOut16G', 'M*.ChClkOut2', 'M*.ClkOut32G', 'M*.TrigOut'
Description	This command can be used to fetch or set the frequency at RefClockOut.
	Please note that RefClkOut on M8009A can only use 10 MHz or 100 MHz. Other than these two values, you cannot set any other value.
	For RefClkOut on M8009A, this query will only return either 10 MHz or 100 MHz.
	For RefClkOut16G, ClkOut16G, ClkOut32G and ChClkOut1/2 on M8009A, it is only applicable as a query. Frequency for all these options changes itself in accordance with the clock frequency.
	This SCPI is applicable for M8041A, M8042A, M8045A, M8009A and M8053A. For M8062A and M8053A, it is only applicable 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 applicable 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', 'M*.DataOut2', 'M*.ChClkOut2' or 'M*.ClkOut32G'
Return Range	-2ns to 2ns
Description	This command is used as offset to Data Delay and Jitter Delay.
	This SCPI is applicable for M8041A, M8042A, M8051A, M8045A and M8009A.
Example	:OUTPut:CALibration:SKEW 'M1.DataOut1', 0

:OUTPut:JITTer [?]

Syntax	:OUTPut:JITTer 'identifier', <Jitter Source>
	:OUTPut:JITTer? 'identifier'
Input Parameters	'identifier': 'M*.ClkOut32G'
	<Jitter Source>: FC16CLock | CCLock | FC1CLock
Description	This command is used to set either Clean Clock, Follow ClkOut16G or Follow ChClkOut1. In the Clean Clock mode, there is no jitter or SSC. In the Follow ClkOut16G mode, the output has the same jitter or SSC as ClkOut16G. In the Follow ChClkOut1 mode, jitter parameters of ChClkOut1 are added to the ClkOut16G/32G.

This SCPI is applicable for M8009A.

Example :OUTPut:JITTer 'M1.ClkOut32G', FC16CLock

: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 applicable for 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 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 applicable for M8041A, M8051A 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 query returns the present setting. This SCPI is applicable for M8041A, M8042A, M8051A, M8062A and M8045A.
Example	:OUTP:DATA:F2J 'M1.DataOut2', -10e-12

:OUTPut:DATA:F2Jitter:STATe[?]

Syntax	:OUTPut:DATA:F2Jitter:STATe 'identifier', <ON|OFF|1|0> :OUTPut:DATA:F2Jitter:STATe? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
Return Range	ON|OFF|1|0
Description	This command enables/disables f/2 jitter at the output. Enabling this command, causes stop of full clock group (all pattern generator channels and CLK/TRIG). This query returns the present setting. This SCPI is applicable for M8042A and M8045A.
Example	:OUTP:DATA:F2J:STAT 'M1.DataOut', ON

:OUTPut:DAC:RANDomizer[:STATe][?]

Syntax :OUTPut:DAC:RANDomizer[:STATe] 'Identifier', <ON|OFF|1|0>

:OUTPut:DAC:RANDomizer[:STATe]? 'Identifier'

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

Return Range ON|OFF|1|0

Description This command switches the randomizer on or off. If the randomizer is on, non-linear distortions of the signal are minimized, but a higher noise floor is introduced. Switch the randomizer on when the non-linear distortions are dominating SNDR and off when the noise is dominating SNDR. By default, the randomizer is off.

This query returns the present setting.

This SCPI is applicable for M8042A.

Example :OUTP:DAC:RAND:STAT 'M2.DataOut1’, ON

:OUTP:DAC:RAND:STAT? 'M2.DataOut1' -> 1

:OUTPut:DEEMphasis Subnode

The M8070B 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[?]
  │└:FILE[?]
  ├:SELect
  │└:PRESet[?]
  │  └:ENABle[?]
  ├:PRECursor{1:2}[?]
  ├:POSTcursor{1:5}[?]
  ├:CONFigure
  │└:SIGN[?]
  ├:CURSor
  │├:MAGNitude(0|1|2|3|4|5|6|7)[?]
  │├:MAIN
  ││├:AUTO[?]
  ││└:SUFFix[?]
  │├:NPRecursor[?]
  │└:NTAPs[?]
  ├:PCIExpress
  │├:PRECursor
  │├:PRECursor2
  │├:POSTcursor
  │└:PRESet
  │  └:FILE][?]
  └:CORRection[?]
```

This subnode has the following commands:

Table 58

Name	Description under
:UNIT[?]	:OUTPut:DEEMphasis:UNIT[?] on page 293
:PRESet[?]	:OUTPut:DEEMphasis:PRESet[?] on page 293
PRESet:FILE[?]	:OUTPut:DEEMphasis:PRESet:FILE[?] on page 294
:PRESet:ENABle[?]	:OUTPut:DEEMphasis:PRESet:ENABle[?] on page 294
:SELect:PRESet[?]	:OUTPut:DEEMphasis:SELect:PRESet[?] on page 295
:PRECursor{1:2}[?]	:OUTPut:DEEMphasis:PRECursor(*)[?] on page 295
:POSTcursor{1:5}[?]	:OUTPut:DEEMphasis:POSTcursor(*)[?] on page 296
:CONFigure:SIGN[?]	:OUTPut:DEEMphasis:CONFigure:SIGN(*)[?] on page 297
:CURSor:MAGNitude(0|1|2|3|4|5|6|7)[?]	:OUTPut:DEEMphasis:CURSor:MAGNitude(0|1|2|3|4|5|6|7)[?] on page 298
:CURSor:MAIN:AUTO[?]	:OUTPut:DEEMphasis:CURSor:MAIN:AUTO[?] on page 298
:CURSor:MAIN:SUFFix[?]	:OUTPut:DEEMphasis:CURSor:MAIN:SUFFix[?] on page 299
:OUTPut:DEEMphasis:CURSor:NPRecursor[?]	:OUTPut:DEEMphasis:CURSor:NPRecursor[?] on page 299
:OUTPut:DEEMphasis:CURSor:NTAPs[?]	:OUTPut:DEEMphasis:CURSor:NTAPs[?] on page 300
:OUTPut:DEEMphasis:PCIExpress	:OUTPut:DEEMphasis:PCIExpress on page 300
:OUTPut:DEEMphasis:PCIExpress:PRECursor	:OUTPut:DEEMphasis:PCIExpress:PRECursor on page 302
:OUTPut:DEEMphasis:PCIExpress:PRECurcor2	:OUTPut:DEEMphasis:PCIExpress:PRECursor2 on page 303
:OUTPut:DEEMphasis:PCIExpress:POSTcursor	:OUTPut:DEEMphasis:PCIExpress:POSTcursor on page 303
:OUTPut:DEEMphasis:PCIExpress:PRESet:FILE[?]	:OUTPut:DEEMphasis:PCIExpress:PRESet:FILE[?] on page 304
:OUTPut:DEEMphasis:CORRection[?]	:OUTPut:DEEMphasis:CORRection[?] on page 304

: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 applicable for M8041A, M8042A, M8051A, M8062A, M8194A, M8196A, and M8045A.

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 M8062A, 0 is the only preset value.

This SCPI is applicable for M8041A, M8042A, M8051A and M8062A.

Example :OUTP:DEEM:PRES 'M1.DataOut2', 2,1.1,2.2,4.3,6.7,5.3,6,3.3

:OUTPut:DEEMphasis:PRESet:FILE[?]

Syntax :OUTPut:DEEMphasis:PRESet:FILE 'identifier', <workspacepath>

:OUTPut:DEEMphasis:PRESet? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

<workspacepath>: Sets the de-emphasis presets definition file.

Description This command selects the de-emphasis presets definition file.

This query returns the de-emphasis presets definition file.

This SCPI is applicable for M8042A and M8045A.

Example :OUTP:DEEM:PRES:FILE 'M1.DataOut1', 'Factory/DeemphasisPresetProfile.xml'

:OUTPut:DEEMphasis:PRESet:ENABle[?]

Syntax :OUTPut:DEEMphasis:PRESet:ENABle 'identifier', <ON | OFF | 1 | 0>

:OUTPut:DEEMphasis:PRESet? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

<ON | OFF | 1 | 0>

Return Range <ON | OFF | 1 | 0>,

Description This command enables or disables changing the deemphasis preset register number. Using this option, user can predefine separate set of cursor values and can apply it using preset register number.

This query returns the present state.

This SCPI is applicable for M8041A, M8042A, M8042A, M8051A, M8062A and M8045A.

Example :OUTP:DEEM:PRES:ENAB 'M2.DataOut2', ON

: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.

Description

This command sets the current preset register. Depending upon the instrument module, the preset registers available.

Supported preset values for M8041A and M8051A are 0 - 30, for M8062A, 0 is the only preset value and for M8045A are 0 - 49.

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.

This SCPI is applicable for M8041A, M8042A, M8051A, M8062A and M8045A.

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 applicable for M8041A, M8051A and M8062A.
	It is also applicable for M8194A but only in query mode.
	For M8045A, the pre-cursor values cannot be changed. You can set the pre-cursor values by adjusting the coefficient values.
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 applicable for M8041A, M8042A, M8051A and M8062A.
	It is also applicable for M8194A but only in query mode.
	For M8045A, the post-cursor values cannot be changed. You can set the post-cursor values by adjusting the coefficient values.
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 command accepts 3-taps or 7-taps sign and the meaning of the first tap changes depending upon the 3-taps or 7-taps sign sent. If 3-taps sign are sent, PreCursor1, PostCursor1, PostCursor2 will be signed. If 7-taps sign are sent, PreCursor2, PreCursor1, PostCursor1, PostCursor2, PostCursor3, PostCursor4, PostCursor5 will be signed.

This SCPI is applicable for M8041A, M8051A 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) modules.

:OUTPut:DEEMphasis:CURSor:MAGNitude(0|1|2|3|4|5|6|7)[?]

Syntax :OUTPut:DEEMphasis:CURSor:MAGNitude(0|1|2|3|4|5|6|7) 'Identifier',<NRf>

:OUTPut:DEEMphasis:CURSor:MAGNitude(0|1|2|3|4|5|6|7)? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Return Range <NRf>

Description This command sets the cursor magnitude as a signed value. The suffix selects the cursor to be addressed. The number of available cursors depends on the product number.

The M8045A supports cursors 0 to 4, where the main cursor is always cursor 2.

The M8194A supports 0 to 4, where the main cursor is query only.

This query returns the present setting.

This SCPI is supported by M8042A, M8045A, and M8194A.

Example :OUTP:DEEM:CURS:MAGN1 'M1.DataOut1',-0.05

:OUTP:DEEM:CURS:MAGN1? 'M1.DataOut1'

:OUTPut:DEEMphasis:CURSor:MAIN:AUTO[?]

Syntax :OUTPut:DEEMphasis:CURSor:MAIN:AUTO 'Identifier',<ON|OFF|1|0>

:OUTPut:DEEMphasis:CURSor:MAIN:AUTO? 'identifier'

Input Parameters 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Return Range ON|OFF|1|0

Description This command controls whether the main cursor magnitude shall be calculated automatically in order to maintain the overall peak to peak amplitude constant, or whether the main cursor magnitude can be controlled manually.

When this is set to ON, then the main cursor magnitude is calculated to fulfill the sum below.

$$\sum_{i}^{n} abs(CursorMagnitude_i) = 1$$

This query returns the present state.

This SCPI is supported by M8042A, M8045A, M8194A and M8196A.

Example :OUTP:DEEM:CURS:MAIN:AUTO 'M1.DataOut1',OFF

:OUTP:DEEM:CURS:MAIN:AUTO? 'M1.DataOut1'

:OUTPut:DEEMphasis:CURSor:MAIN:SUFFix[?]

Syntax :OUTPut:DEEMphasis:CURSor:MAIN:SUFFix 'Identifier',<NR1>

:OUTPut:DEEMphasis:CURSor:MAIN:SUFFix? 'Identifier'

Input Parameters Identifier: 'M*.DataOut1','M*.DataOut2'

Description This command sets the main cursor's suffix and therefore the number of pre-cursor taps.

This can be changed from 1 to 3. The default value is 2.

This command is equivalent to:

:OUTPut:DEEMphasis:CURSor:NPRecursor[?]

This query returns the main cursor's suffix.

This SCPI is supported by M8042A and M8045A.

Example :OUTP:DEEM:CURS:MAIN:SUFF 'M1.DataOut1',3

:OUTP:DEEM:CURS:MAIN:SUFF? 'M1.DataOut1'

Returns: 3

:OUTPut:DEEMphasis:CURSor:NPRecursor[?]

Syntax :OUTPut:DEEMphasis:CURSor:NPRecursor 'identifier',<NR1>

:OUTPut:DEEMphasis:CURSor:NPRecursor? 'identifier'

Input Parameters Identifier: 'M*.DataOut1','M*.DataOut2'

Description This command sets the number of pre-cursor taps.

This can be changed from 1 to 3. The default value is 2.

This command is equivalent to:

:OUTPut:DEEMphasis:CURSor:MAIN:SUFFix

This query returns the number of pre-cursor taps.

This SCPI is supported by M8042A and M8045A.

Example :OUTP:DEEM:CURS:NPR 'M1.DataOut1',3

:OUTP:DEEM:CURS:NPR? 'M1.DataOut1'

Returns: 3

:OUTPut:DEEMphasis:CURSor:NTAPs[?]

Syntax :OUTPut:DEEMphasis:CURSor:NTAPs 'identifier',<NR1>

:OUTPut:DEEMphasis:CURSor:NTAPs? 'identifier'

Input Parameters Identifier: 'M*.DataOut1','M*.DataOut2'

Description This command sets the number of de-emphasis taps.

This query returns the number of de-emphasis taps.

This SCPI is supported by M8042A and M8045A. The number of taps is always five for M8045A

Example :OUTP:DEEM:CURS:NTAP 'M1.DataOut1',5

:OUTP:DEEM:CURS:NTAP? 'M1.DataOut1'

Returns: 5

:OUTPut:DEEMphasis:PCIExpress

Description When using the PCIe LTSSM on a M8045A, then the de-emphasis is controlled by a preset table. The table defines the full-swing value which will be advertised to the device under test and contains one de-emphasis definition for each of the presets that result from the full-swing selection.

Each preset consists of a coefficient value for a pre-, main-, and post cursor.

The presets are addressed by the nominals pre-cursor and post-cursor value with a resolution given by the full-swing value. This leads to pre- and post-cursor values being integer values, with a value range that depends on the full-swing value and the respective other cursor's value.

Example For a full-swing value of 63, the absolute range for the pre-cursor is 0...15 and for the post cursor it is 0...21.

The resulting nominal pre- and post-cursor de-emphasis coefficients are 0/63...15/63 and 0/63...21/63.

The preset table therefore provides a means of calibrating the pre-, main-, and post-cursor for each of the valid nominal cursor settings of the PCIe LTSSM.

Import preset definitions into the workspace

Use the following SCPI commands to import user defined preset tables into the M8070B workspace

```
:MMEMory:WORKspace:IMPort[:USER]:CATalog? PCIDeemphasis

:MMEMory:WORKspace:IMPort[:USER]
PCIDeemphasis,<"SourcePath">,<"RelativeTargetPath">

:MMEMory:WORKspace:IMPort[:USER]:DELete
PCIDeemphasis,<"RelativeTargetPath">
```

M8070B contains default preset tables for all supported full-swing values (24 .. 63). These can be used as a starting point for external calibration of the de-emphasis presets.

The naming scheme for these default preset-tables is as follows:

Factory/FullSwing-XX.xml

where XX is the full-swing value (24 .. 63).

The calibration code will have to find the de-emphasis coefficient for the pre-cursor, main-cursor and post-cursor that results in the pre-shoot and de-emphasis given by Pre/FullSwing and Post/FullSwing for each of the entries in the Presets list.

The content of the preset definition is defined like this:

```
<?xml version="1.0" encoding="utf-16"?>

<PCIeLtssmDeemphasisPresets
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Version>1.0</Version>

  <Description>De-emphasis tap gain presets for the PCIe
LTSSM</Description>

  <FullSwing>24</FullSwing>
```

```
  <Presets>
    <P>
      <Pre>0</Pre>
      <Post>0</Post>
      <Coefficients>0,1,0</Coefficients>
    </P>
<!-- entries removed for clarity -->
<P>
      <Pre>6</Pre>
      <Post>2</Post>

<Coefficients>-0.25,0.666666666666667,-0.0833333333333333</C
oefficients>
    </P>
  </Presets>
</PCIeLtssmDeemphasisPresets>
```

:OUTPut:DEEMphasis:PCIExpress:PRECursor

Syntax :OUTPut:DEEMphasis:PCIExpress:PRECursor 'Identifier',<NRf>

Input Parameters 'identifier': 'M*.DataOut1' or 'M*.DataOut2'
<NRf>: Pre-cursor value

Description This command sets the pre-cursor to the given value, with a nominal cursor coefficient of pre-cursor/full-scale, e.g. 7/63 for a full-scale value of 63.

The PCIe de-emphasis presets are only available when the pattern sequence is configured for the PHY protocol PCIe3, PCIe4 or PCIe5. When the sequence is not using the PCIe LTSSM, then the de-emphasis is controlled via the cursor magnitudes directly.

This SCPI is only supported by M8045A.

Example :OUTP:DEEM:PCIE:PREC 'M1.DataOut1',7

:OUTPut:DEEMphasis:PCIExpress:PRECursor2

Syntax :OUTPut:DEEMphasis:PCIExpress:PRECursor2 'Identifier',<NRf>

Input Parameters 'identifier': 'M*.DataOut1' or 'M*.DataOut2'
<NRf>: Pre-cursor value

Description This command sets the pre-cursor 2 to the given value, with a nominal cursor coefficient of pre-cursor/full-scale, e.g. 7/63 for a full-scale value of 63.

The PCIe de-emphasis presets are only available when the pattern sequence is configured for the PHY protocol PCIe3, PCIe4 or PCIe5. When the sequence is not using the PCIe LTSSM, then the de-emphasis is controlled via the cursor magnitudes directly. This command is available for the PHY protocol PCIe6 only.

This SCPI is only supported by M8045A.

Example :OUTP:DEEM:PCIE:PREC2 'M1.DataOut1',7

:OUTPut:DEEMphasis:PCIExpress:POSTcursor

Syntax :OUTPut:DEEMphasis:PCIExpress:POSTcursor 'Identifier',<NRf>

Input Parameters 'identifier': 'M*.DataOut1' or 'M*.DataOut2'
<NRf>: Post-cursor value

Description This command sets the post-cursor to the given value, with a nominal cursor coefficient of post-cursor/full-swing, e.g. 11/63 for a full- swing value of 63.

The PCIe de-emphasis presets are only available when the pattern sequence is configured for the PHY protocol PCIe3, PCIe4 or PCIe5. When the sequence is not using the PCIe LTSSM, then the de-emphasis is controlled via the cursor magnitudes directly.

This SCPI is only supported by M8045A.

Example :OUTP:DEEM:PCIE:POST 'M1.DataOut1',11

:OUTPut:DEEMphasis:PCIExpress:PRESet:FILE[?]

Syntax :OUTPut:DEEMphasis:PCIExpress:PRESet:FILE[?] 'Identifier',<filename>

Input Parameters 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Description This command selects the de-emphasis preset definition for the PCIe LTSSM.
When switching the preset definition table, the software will try to maintain the cursor coefficients as constant as possible. The accuracy of this adoption depends on the resolution of the initial and new preset table. Due to the fractional nature of the nominal coefficients, there may be a small variation in the generated de-emphasis.

For example, when changing from a setting with full-swing 46, pre-cursor 5 and post-cursor 11 to a preset table with a full-swing value of 63, the pre-and post-cursor setting will be adapted. But the resulting signal will still vary a bit, because 5/46 cannot be expressed as x/63 with x being an integral value.

The PCIe de-emphasis presets are only available when the pattern sequence is configured for the PHY protocol PCIe3, PCIe4 or PCIe5. When the sequence is not using the PCIe LTSSM, then the de-emphasis is controlled via the cursor magnitudes directly, but the preset- table for the LTSSM can still be selected.

This SCPI is only supported by M8045A.

Example :OUTP:DEEM:PCIE:PRES:FILE 'M1.DataOut1', 'Factory/FullSwing-63.xml'

:OUTPut:DEEMphasis:CORRection[?]

Syntax :OUTPut:DEEMphasis:CORRection 'Identifier',<DEFault | NONE>

:OUTPut:DEEMphasis:CORRection? 'Identifier'

Input Parameters 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

DEFault | NONE

Description This command sets the cursor correction mode. The available cursor correction modes are “DEFault” and “NONE”. If the cursor correction mode is selected as “DEFault” then it uses the factory calibration table, however, if “NONE” is selected, it will turn off the calibration table.This parameter is only available for PAM3 line coding and when the deemphasis profile is selected as “Factory/USB4_80G.xml”.

This query returns the present setting.

This SCPI is supported by M8045A.

Example :OUTP:DEEM:CORR 'M1.DataOut1', DEF

:OUTP:DEEM:CORR? 'M1.DataOut1' -> DEF

:OUTPut:EINSertion Subnode

This subnode has the following SCPI structure:

```
:OUTPut
└:EINSertion
  ├[:STATe][?]
  ├:RATio[?]
  ├:MODE[?]
  ├:ONCE
  ├:SYMBol:ONCE
  └:DISParity
```

This subnode has the following commands:

Table 59

Name	Description under
[:STATe][?]	:OUTPut:EINSertion[:STATe][?] on page 306
:RATio[?]	:OUTPut:EINSertion:RATio[?] on page 306
:MODE[?]	:OUTPut:EINSertion:MODE[?] on page 307
:ONCE	:OUTPut:EINSertion:ONCE on page 307
:SYMBol:ONCE	:OUTPut:EINSertion:SYMBol:ONCE on page 308
:DISParity	:OUTPut:EINSertion:DISParity on page 308

: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 applicable for M8041A, M8051A, M8062A and M8045A.

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 applicable for M8041A, M8042A, M8045A, M8051A 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 applicable for M8041A, M8042A, M8045A, M8051A 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 applicable for M8041A, M8042A, M8045A, M8051A and M8062A.
Example	:OUTP:EINS:ONCE 'M1.DataOut1'

:OUTPut:EINSertion:SYMBol:ONCE

Syntax	:OUTPut:EINSertion:SYMBol:ONCE 'identifier'
Input Parameters	'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
Description	This command generates a single symbol error (single symbol error) on the specified output. This command is only applicable when the M8045A pattern generator module operates in the PAM3 line coding. This SCPI is applicable for M8045A only.
Example	:OUTP:EINS:SYMB: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 applicable for M8041A, M8051A and M8045A.
Example	:OUTP:EINS:DISP 'M1.DataOut2'

:OUTPut:EINSertion:TYPE[?]

Syntax	:OUTPut:EINSertion:TYPE 'identifier' <Error Type>
	:OUTPut:EINSertion:TYPE? 'identifier'
Input Parameters	'identifier': 'M*.DataOut1' or 'M*.DataOut2'
	<Error Type>: BIT | SYMBol
Description	This command generates the bit error ('Bit') or symbol error ('Symbol') on the specified output. It is only applicable when the M8045A pattern generator module operates in PAM3 or PAM4 line coding.
	This query returns the currently selected error type.
	This SCPI command is applicable to the M8045A module.
Example	:OUTP:EINS:TYPE 'M1.DataOut1',BIT
	:OUTP:EINS:TYPE? 'M1.DataOut1' -> BIT

:OUTPut:TRIGger Subnode

This subnode has the following SCPI structure:

```
:OUTPut
└:TRIGger
  ├:MODE[?]
  └:DIVider[?]
```

This subnode has the following commands:

Table 60

Name	Description under
:MODE[?]	:OUTPut:TRIGger:MODE[?] on page 310
:DIVider[?]	:OUTPut:TRIGger:DIVider[?] on page 311

: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 applicable for M8041A M8042A and M8045A.

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 applicable for M8041A, M8042A and M8045A.
Example	:OUTP:TRIG:DIV 'M1.TrigOut', 8

:OUTPut:DEMBedding Subnode

This subnode has the following SCPI structure:

```
:OUTPut
└:DEMBedding
  ├[:SCABle]
  │ └[:STATe][?]
  ├[:CHANnel]
  │ └[:STATe][?]
  └:SPARameter
    ├ [:STATe][?]
    ├ :PROFile[?]
    ├ :IPORt[?]
    └ :OPORt[?]
```

This subnode has the following commands:

Table 61

Name	Description under
:SCABle:STATe[?]	:OUTPut:DEMBedding:SCABle:STATe[?] on page 313
:CHANnel:STATe[?]	:OUTPut:DEMBedding:CHANnel:STATe[?] on page 313
SPARameter:STATe[?]	:OUTPut:DEMBedding:SPARameter:STATe[?] on page 313
:SPARameter:PROFile[?]	:OUTPut:DEMBedding:SPARameter:PROFile[?] on page 314
:SPARameter:IPORt[?]	:OUTPut:DEMBedding:SPARameter:IPORt[?] on page 314
:SPARameter:OPORt[?]	:OUTPut:DEMBedding:SPARameter:OPORt[?] on page 315
:SPARameter:DCLoss?	:OUTPut:DEMBedding:SPARameter:DCLoss? on page 315

: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 applicable for M8194A, M8195A and M8196A.
Example	:OUTP:DEMB:SCAB:STAT 'M1.DataOut1', ON

:OUTPut:DEMBedding:CHANnel:STATe[?]

Syntax	:OUTPut:DEMBedding:CHANnel:STATe 'identifier', <OFF | ON | 0 | 1>
	:OUTPut:DEMBedding:CHANnel: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 channel specific correction of the amplitude and phase using an internal calibration.
	The query returns the present setting.
	This SCPI is applicable for M8053A, M8054A, M8194A, M8195A and M8196A.
Example	:OUTP:DEMB:CHAN: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 applicable for M8053A, M8194A, M8195A and M8196A. For M8045A, only S2P files are supported.
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 applicable for M8194A, M8195A, M8196A and M8053A.
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 applicable for M8194A, M8195A, M8196A and M8053A.
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 applicable for M8194A, M8195A, M8196A and M8053A.
Example	:OUTP:DEMB:SPAR:OPOR 'M1.DataOut1', 2

:OUTPut:DEMBedding:SPARameter:DCLoss?

Syntax	:OUTPut:DEMBedding:SPARameter:DCLoss? 'identifier'
Input Parameters	'identifier':'M*.DataOut1', 'M*.DataOut2'
Description	This query returns the DC loss that is estimated from the loaded S-Parameters.
	This query is applicable for M8053A module.
Example	:OUTPut:DEMBedding:SPARameter:DCLoss? 'M1.DataOut2' -> 0e+8

:OUTPut:EMBedding Subnode

This subnode has the following SCPI structure:

```
:OUTPut
└:EMBedding
  └:SPARameter
    ├ [:STATe][?]
    ├ :PROFile[?]
    ├ :IPORt[?]
    └ :OPORt[?]
```

This subnode has the following commands:

Table 62

Name	Description under
:SPARameter[:STATe][?]	:OUTPut:EMBedding:SPARameter[:STATe][?] on page 316
:SPARameter:PROFile[?]	:OUTPut:EMBedding:SPARameter:PROFile[?] on page 317
:SPARameter:IPORt[?]	:OUTPut:EMBedding:SPARameter:IPORt[?] on page 317
:SPARameter:OPORt[?]	:OUTPut:EMBedding:SPARameter:OPORt[?] on page 317

: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 applicable for M8194A, M8195A and M8196A. For M8045A, only S2P files are supported.

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 applicable for M8194A, M8195A and M8196A.

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 applicable for M8194A, M8195A and M8196A.

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 applicable for M8194A, M8195A and M8196A.

Example OUTP:EMB:SPAR:OPOR 'M1.DataOut1', 2

:OUTPut:ADJustment Subnode

This subnode has the following SCPI structure:

```
:OUTPut
└:ADJustment
  ├:ABORt
  ├:STATus
  │ └[:VALue]?
  ├:BIND
  ├:DEEMphasis
  │ └:AUTO
  ├:APPLY[?]
  ├:ATTenuation[?]
  └:FILTer[?]
```

NOTE M8045A is the only instrument capable of supporting these commands for DataOut locations.

This subnode has the following commands:

Table 63

Name	Description under
:ABORt	:OUTPut:ADJustment:ABORt on page 319
:STATus [:VALue]?	:OUTPut:ADJustment:STATus [:VALue]? on page 319
:BIND	:OUTPut:ADJustment:BIND on page 320
:DEEMphasis:AUTO	:OUTPut:ADJustment:DEEMphasis:AUTO on page 320
APPLY[?]	:OUTPut:ADJustment:APPLY[?] on page 320
ATTenuation[?]	:OUTPut:ADJustment:ATTenuation[?] on page 321
FILTer[?]	:OUTPut:ADJustment:FILTer[?] on page 321

:OUTPut:ADJustment:ABORt

Syntax	:OUTPut:ADJustment:ABORt 'identifier'
Input Parameters	'identifier': 'M1.DataOut1','M2.DataOut2'
Description	This command interrupts the adjustments.
Example	:OUTP:ADJ:ABOR 'M1.DataOut1'

:OUTPut:ADJustment:STATus [:VALue]?

Syntax	:OUTPut:ADJustment:STATus [:VALue]? 'Identifier'
Input Parameters	Any pattern generator location and group name. Example: 'M1.DataOut1', 'M1.DataOut2'
Description	This query retrieves adjustment status information per pattern generator or pattern generator group. This query returns the following values:

INPROGRESS	At least one addressed pattern generator is executing an adjustment procedure.
ABORTED	The adjustment procedure has been aborted by the user before all pattern generator channels finished the procedure.
FAILED	All addressed pattern generator channels have finished the adjustment procedure, but at least one was unable to finish successfully.
SUCCESS	The addressed pattern generator channels have successfully finished the adjustment procedure. Pattern Generator channels that are addressed by the command but return UNKNOWN state, do not contribute to the result.
UNKNOWN	None of the addressed pattern generators has executed an adjustment procedure yet.

NOTE This query cannot be used in the same compound SCPI statement that is starting the adjustment procedure. If this query is sent to the instrument right after the command that is initiating the adjustment procedure, then the returned value is undefined and can reflect the state that was valid before the adjustment. This can be overcome by reading any value from the instrument (e.g. *ESE?), except for *OPC?, as this will block until the adjustment procedure is finished.

Example :OUTP:ADJ:STAT? 'M1.DataOut1'

:OUTPut:ADJustment:BIND

Syntax	:OUTPut:ADJustment:BIND 'Identifier', 'measurement channel identifier'
Input Parameters	Identifier: 'M1.DataOut1', 'M1.DataOut2' or 'INF1.DataIn1' Measurement Channel Identifier: 'DCA*.Slot*.Channel 1A', 'DCA*.Slot*.Channel 2A' or 'UXR*.Slot*.Channel 1A',
Description	This command selects the location of the oscilloscope channel that shall be used by the addressed pattern generator channel. This is required to automatically determine and set the deemphasis coefficients at the end of the cable.
Example	:OUTP:ADJ:BIND 'INF1.DataIn1'

:OUTPut:ADJustment:DEEMphasis:AUTO

Syntax	:OUTPut:ADJustment:DEEMphasis:AUTO 'identifier'
Input Parameters	'M1.DataOut1' or 'M1.DataOut2'
Description	This command automatically determines and sets the deemphasis coefficients at the end of cable.
Example	:OUTP:ADJ:DEEM:AUTO 'M1.DataOut1'

:OUTPut:ADJustment:APPLY[?]

Syntax	:OUTPut:ADJustment:APPLY 'identifier', <1 | 0 | ON | OFF> :OUTPut:ADJustment:APPLY? 'identifier'
Input Parameters	'identifier': 'M2.DataOut1' or 'M2.DataOut2' <1 | 0 | ON | OFF>
Description	This command enables or disables the correction data once the deemphasis adjustment has been completed. Note that the maximum and minimum limits of the amplifier will be adjusted to match the measured amplitude on the reference plane. Consequently, it is possible that the current amplitude may fall outside the allowed range. In such cases, utilize SCPI commands to adjust the amplitude within the permissible range. Additionally, it may be

necessary to turn off the amplifier state beforehand.

This query returns the current settings.

This command is applicable for M8042A.

Example :OUTP:ADJ:APPLY ' M2.DataOut1’, ON

:OUTP:ADJ:APPLY? ' M2.DataOut1’ -> 1

:OUTPut:ADJustment:ATTenuation[?]

Syntax :OUTPut:ADJustment:ATTenuation 'identifier', <Attenuation Value>

:OUTPut:ADJustment:ATTenuation? 'identifier'

Input Parameters 'identifier': 'M2.DataOut1' or 'M2. DataOut2'

<Attenuation Value>: 0 to 120 dB

Description This command sets attenuation if an external attenuator is connected. If no external attenuators are used, set it to 0 dB.

This query returns the current attenuation value.

This SCPI is only valid for M8042A.

Example :OUTP:ADJ:ATT ‘M2.DataOut1’, 0

:OUTP:ADJ:ATT? ‘M2.DataOut1’ -> 0

:OUTPut:ADJustment:FILTer[?]

Syntax :OUTPut:ADJustment:FILTer 'identifier', <Optimization Filter>

:OUTPut:ADJustment:FILTer? 'identifier'

Input Parameters 'identifier': 'M2.DataOut1' or 'M2. DataOut2'

<Optimization Filter>: FFE | TDECq

Description This command sets the optimization method, either through a linear equalizer or TDECQ equalizer.

This query returns the current applied optimization filter.

This SCPI is only valid for M8042A.

Example :OUTP:ADJ:FILT 'M2.DataOut1', TDEC

:OUTP:ADJ:FILT? 'M2.DataOut1' -> TDEC

EVENt Subsystem

This subsystem has the following SCPI structure:

```
:EVENt
└:ADVance
  ├:STATe[?]
  └:SOURce[?]
```

This subsystem has the following commands:

Table 64

Name	Description under
:EVENt:ADVance	:EVENt:ADVance on page 322
:EVENt:ADVance:STATe[?]	:EVENt:ADVance:STATe[?] on page 323
:EVENt:ADVance:SOURce[?]	:EVENt:ADVance:SOURce[?] on page 323

:EVENt:ADVance

Syntax :EVENt:ADVance 'identifier'

Input Parameters 'identifier': 'M1.EventConfig'

Description This command asserts the advance event in the M819xA Soft Front Panel.

This SCPI is applicable for M8195A.

Example :EVEN:ADV:SOUR 'M1.EventConfig', TRIG

:EVENt:ADVance:STATe[?]

Syntax
:EVENt:ADVance:STATe[?] 'identifier', <1 | 0 | ON | OFF>

:EVENt:ADVance:STATe? 'identifier'

Input Parameters
'identifier': 'M1.EventConfig'

1 | 0 | ON | OFF

Description
This command enables/disables the hardware driven advance event in the Soft Front Panel. Triggering this event can also be done via software.

The query returns the present setting.

This SCPI is applicable for M8195A.

Example
:EVEN:ADV:STAT 'M1.EventConfig', 1

:EVEN:ADV:STAT? 'M1.EventConfig'

1

:EVENt:ADVance:SOURce[?]

Syntax
:EVENt:ADVance:SOURce[?] 'identifier', <TRIGger | EVENt>

:EVENt:ADVance:SOURce? 'identifier'

Input Parameters
'identifier': 'M1.EventConfig'

TRIGger | EVENt

Description
This command allows to select whether the Trigger In or the Event In is used as hardware input for advance event.

The query returns the present setting.

This SCPI is applicable for M8195A.

Example
:EVEN:ADV:SOUR 'M1.EventConfig', TRIG

:EVEN:ADV:SOUR? 'M1.EventConfig'

TRIG

INPut Subsystem

The INPut subsystem controls the input ports of the analyzer.

This subsystem has the following SCPI structure:

```
:INPut
├─:ANAlyzer:EYEOptimize:STATe[?]
├─:BRM[?]
├─:DRATe?
├─:DELay[?]
├─:POLarity[?]
├─:CMVoltage[?]
│ └─:DETect
├─:THReshold[?]
├─:TVOLtage[?]
├─:TCONfig[?]
├─:CMODe[?]
├─:COUPling[?]
├─:SENSitivity[?]
├─[:VOLTage]
│ ├─:WINDow[?]
│ │ ├─:ALLocate[?]
│ │ └─:DETect
│ ├─ [:NRZ]:THReshold[?]
│ ├─ :PAM3[:SYMBol]:THReshold(1|2)[?]
│ ├─ :PAM4[:SYMBol]:THReshold(1|2|3)[?]
│ ├─ :PAM6[:SYMBol]:THReshold(1|2|3|4|5)[?]
│ └─ :PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7)[?]
├─:CMPower[?]
├─:POWer
│ ├─:WINDow[?]
│ │ └─:ALLocate[?]
│ ├─[:NRZ]:THReshold[?]
│ ├─:PAM4[:SYMBol]:THReshold(1|2|3)[?]
│ ├─:PAM6[:SYMBol]:THReshold(1|2|3|4|5)[?]
│ └─:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7)[?]
├─:DyRx:State[?]
├─:LTSSm:FILE[?]
├─:CDR
│ └─...
├─:ALIGnment
│ └─...
├─:EQUalization
│ └─...
├─:SQUElch
│ └─...
├─:PROBe
│ └─...
└─:DEMBedding
  └─...
```

The subsystem has the following commands and subnodes:

Table 65

Name	Description under
:ANAlyzer:EYEOptimize:STATe[?]	:INPut:ANAlyzer:EYEOptimize:STATe[?] on page 327
:BRM[?]	:INPut:DATA:BRMode[?] on page 328
:DRATe?	:INPut:DRATe? on page 328
:DELay	:INPut:DELay[?] on page 328
:POLarity	:INPut:POLarity[?] on page 329
:CMVoltage[?]	:INPut:CMVoltage[?] on page 329
:CMVoltage:DETect	:INPut:CMVoltage:DETect on page 330
:THReshold	:INPut:THReshold[?] on page 330
:TVOLtage	:INPut:TVOLtage[?] on page 331
:TCONfig[?]	:INPut:TCONfig[?] on page 331
:CMODe[?]	:INPut:CMODe[?] on page 331
COUPling[?]	:INPut:COUPling[?] on page 332
SENSitivity[?]	:INPut:SENSitivity[?] on page 333
[:VOLTage]:WINDow[?]	:INPut[:VOLTage]:WINDow[?] on page 333
[:VOLTage]:WINDow:ALLocate[?]	:INPut[:VOLTage]:WINDow:ALLocate[?] on page 334
[:VOLTage]:WINDow:DETect	:INPut[:VOLTage]:WINDow:DETect on page 334
:INPut[:VOLTage][:NRZ]:THReshold[?]	:INPut[:VOLTage][:NRZ]:THReshold[?] on page 335
:INPut[:VOLTage]:PAM3[:SYMBol]:THReshold(1|2)[?]	:INPut[:VOLTage]:PAM3[:SYMBol]:THReshold(1|2)[?] on page 335
:INPut[:VOLTage]:PAM4[:SYMBol]:THReshold(1|2|3)[?]	:INPut[:VOLTage]:PAM4[:SYMBol]:THReshold(1|2|3)[?] on page 336
:INPut[:VOLTage]:PAM6[:SYMBol]:THReshold(1|2|3|4|5)[?]	:INPut[:VOLTage]:PAM6[:SYMBol]:THReshold(1|2|3|4|5)[?] on page 337
:INPut[:VOLTage]:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7)[?]	:INPut[:VOLTage]:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7)[?] on page 338
:INPut:CMPower[?]	:INPut:CMPower[?] on page 338
:INPut:POWer:WINDow[?]	:INPut:POWer:WINDow[?] on page 339
:INPut:POWer:WINDow:ALLocate[?]	:INPut:POWer:WINDow:ALLocate[?] on page 339

Name	Description under
:INPut:POWer[:NRZ]:THReshold[?]	:INPut:POWer[:NRZ]:THReshold[?] on page 340
:INPut:POWer:PAM4[:SYMBol]:THReshold(1|2|3)[?]	:INPut:POWer:PAM4[:SYMBol]:THReshold(1|2|3)[?] on page 340
:INPut:POWer:PAM6[:SYMBol]:THReshold(1|2|3|4|5)[?]	:INPut:POWer:PAM6[:SYMBol]:THReshold(1|2|3|4|5)[?] on page 341
:INPut:POWer:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7)[?]	:INPut:POWer:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7)[?] on page 342
:INPut:DyRx:State[?]	:INPut:DyRx:State[?] on page 344
:INPut:LTSSm:FILE[?]	:INPut:LTSSm:FILE[?] on page 344
Subnodes	
:CDR	:INPut:CDR Subnode on page 345
:ALIGnment	:INPut:ALIGnment Subnode on page 356
:EQUalization	:INPut:EQUalization Subnode on page 367
:SQUElch	:INPut:SQUElch Subnode on page 394
:PROBe	:INPut:PROBe Subnode on page 395
:DEMBedding	:INPut:DEMBedding Subnode on page 397

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 66

License	Description	Modules
M8009A-062	Clock Generator two channel 60 GHz	M8009A
M8009A-0G6	Reference Clock Multiplier	M8009A
M8041A-0A2	Second Channel for Analyzer	M8041A
M8041A-0A3	Analyzer Equalization, Module-wide License	M8041A
M8041A-0G2	Second Channel for Generator	M8041A
M8041A-0G6	Reference Clock Input with Multiplying PLL, Clockgroup-wide License	M8041A

License	Description	Modules
M8041A-C08	BERT one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8041A-G08	Generator one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8042A-G32	32 GBd PG for NRZ and PAM4	M8042A
M8043A-0A3	Equalization and De-Embedding capability	M8043A
M8043A-A32	Error analysis up to 32.4 GBd for NRZ and PAM4	M8043A
M8045A-G32	32 GBd Pattern Generator	M8045A
M8046A-0A3	Equalizer	M8046A
M8046A-0A4	Clock Recovery for 32 GBd	M8046A
M8046A-0P3	PAM4 decoding up to 32 GBd	M8046A
M8046A-A32	32GBd Error Detector	M8046A
M8062A-C32	32 Gb/s BERT front-end	M8062A
M8062A-G32	32 Gb/s Pattern generator front-end	M8062A
M8070ADVB	Advanced Measurement Package for M8000 Series BERT Test Solutions	M8041A, M8043A, M8046A

:INPut:ANAlyzer:EYEOptimize:STATe[?]

Syntax :INPut:ANAlyzer:EYEOptimize:STATe 'identifier', < 0 | 1 | ON | OFF>

:INPut:ANAlyzer:EYEOptimize:STATe 'identifier'

Input Parameters 'identifier': 'INF*.DataIn1'

Range 0 | 1 | ON | OFF

Description This command enables/disables the eye optimization process.

This query returns the present state.

Example :INP:ANA:EYEO:STAT 'INF1.DataIn1', ON

:INP:ANA:EYEO:STAT? 'INF1.DataIn1' -> 1

: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 applicable for M8062A, M8043A and M8046A.
Example	:INP:DRAT? 'M1.DataIn1'

:INPut:DELay[?]

Syntax	:INPut:DELay 'identifier', <NRf>
	:INPut:DELay? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn', 'INF*.DataIn1'
	<NRf>: Specify sampling edge delay value.
Return Range	For M8041A, M8051A and M8046A: -6.7 ns to +6.7 ns

	For M8062A: -10 ns to 10 ns
	For M8046A: -6.7 ns to 6.7 ns
Description	This command specifies the delay of the sampling edge.
	This SCPI is applicable for M8041A, M8051A, M8062A, M8043A, M8046A and DSO634A, DSA634A, DSAX96204A and DSOX96204A.
Example	:INP:DEL 'M1.DataIn1', 1e-9

:INPut:POLarity[?]

Syntax	:INPut:POLarity 'identifier', <NORMal | INVerted | EITHer>
	:INPut:POLarity? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*.DataIn', 'INF*.DataIn1'
	<NORMal | INVerted | EITHer>: Set input polarity. The EITHer option only works with the M8195A AWG module.
Return Range	NORM | INV | EITH
Description	This command sets the input polarity of the specified input to either NORMal, INVerted or EITHer.
	This SCPI is applicable for M8041A, M8051A, M8062A, M8043A, M8046A, M8195A, DSO634A, DSA634A, DSAX96204A and DSOX96204A.
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 141.
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 applicable for M8041A, M8051A and M8043A.

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 applicable for M8041A, M8051A and M8043A (command only).

Example :INP:CMV:DET 'M1.DataIn1'

:INPut:THReshold[?]

Syntax :INPut:THReshold 'identifier', <NRf>

:INPut:THReshold? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2', 'M2.ElectIdleIn', 'M*DataIn', 'SysIn A/B' or 'CtrlIn A/B'

<NRf>: Set receiver input threshold voltage.

Return Range For M8041A and M8051A: -1 V to 1 V

For M8062A: -800 mV to 800 mV

For M8046A: -250 mV to 250 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 applicable for M8041A, M8051A, M8062A and M8046A.

Example :INP:THR 'M1.DataIn1', 200 mV

:INPut:TVOLtage[?]

Syntax	:INPut:TVOLtage 'identifier', <NRf> :INPut:TVOLtage? 'identifier'
Input Parameters	'identifier':'M*.DataIn1', 'M*.DataIn2', 'M*.DataIn' 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 141.
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 applicable for M8041A, M8042A, M8043A, M8051A, M8043A and M8062A.
Example	:INP:TVOL 'M1.DataIn1', 300mV

:INPut:TCONfig[?]

Syntax	:INPut:TCONfig 'identifier', <BALanced|UNBalanced> :INPut:TCONfig? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*.DataIn' <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 applicable for M8041A, M8051A, M8043A and M8046A.
Example	:INP:TCON 'M1.DataIn1', BAL

:INPut:CMODe[?]

Syntax :INPut:CMODe 'identifier', <SENormal|SEComplement|DIFFerential|SINGleended>

:INPut:CMODe? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn', 'INF*.DataIn1'

<SENormal|SEComplement|DIFFerential|SINGleended>: Select compare mode.

Return Range SEN|SEC|DIFF|SING

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 or connected to the complement output of the source. This allows to measure a single-ended signal as well as the non-inverted signal of a differential pair.

SEComplement: Data stream is compared at Complement input; Normal is terminated or connected to the normal output of the source. This allows to measure a single-ended signal as well as the inverted signal of a differential pair.

DIFFerential: If differential both input ports need to receive a signal. The actual data signal is measured as the voltage difference between the two incoming signals.

SINGleended: Data stream is provided at either Normal input or Complement input. The other input is either left open or terminated with 50 Ohm.

Note that the availability of the particular selection depends on the currently used hardware.

This SCPI is applicable for M8041A, M8051A, M8043A, M8046A and real-time oscilloscope based error detector channels.

Example :INP:CMOD 'M1.DataIn1', SEC

:INPut:COUPling[?]

Syntax :INPut:COUPling 'identifier', <AC|DC>

:INPut:COUPling? 'identifier'

Input Parameters 'identifier': 'M*.DataIn', '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 applicable for M8041A, M8051A, M8043A and M8046A.

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 applicable for M8041A, M8051A and M8062A.

Example :INP:SENS 'M1.DataIn1', HIGH

:INPut[:VOLTage]:WINDow[?]

Syntax :INPut[:VOLTage]:WINDow 'Identifier',< NRf>

:INPut[:VOLTage]:WINDow? 'identifier'

Input Parameters 'identifier': 'M*.DataIn' or 'INF*.DataIn1'

Description This command defines the input voltage window. The voltage applied at the Data In connector needs to be within the voltage range defined by the common mode voltage and the voltage window.

$$V_{min} = Common\ Mode\ Voltage - \frac{Input\ Voltage\ Window}{2}$$

$$V_{max} = Common\ Mode\ Voltage + \frac{Input\ Voltage\ Window}{2}$$

This SCPI is supported by M8043A, M8046A and all supported real-time oscilloscopes.

Example :INP:WIND 'M2.DataIn1', 0.4

:INPut[:VOLTage]:WINDow:ALLocate[?]

Syntax :INPut[:VOLTage]:WINDow:ALLocate 'Identifier',< NRf>

:INPut[:VOLTage]:WINDow:ALLocate? 'identifier'

Input Parameters 'identifier': 'INF*.DataIn1'

Description This command defines how much of the input voltage window shall be used by the received signal.

This controls how the auto-set algorithm calculates the input voltage window size based on the measured signal amplitude. Normally the input voltage window is chosen larger than the signal amplitude to avoid nonlinear distortions at the peak voltage levels.

When working with small signal amplitudes, the BER results of the real-time oscilloscopes may be better if the received signal is using the input voltage range is used to a larger extent, than what the oscilloscope choses during the built-in vertical auto-scale operation.

Changing this parameter is not directly changing the input voltage range, but has effect on the next vertical auto-scale operation.

This SCPI is only supported by for real-time oscilloscopes.

Example :INP:WIND:ALL 'INF1.DataIn1', 90

:INPut[:VOLTage]:WINDow:DETect

Syntax :INPut[:VOLTage]:WINDow:DETect 'Identifier'

Input Parameters 'identifier': 'M*.DataIn'

Description This command automatically adjust the input voltage window to match the received signal at the Data In connector.

- The execution progress can be queried using the command, see :INPut:ALIGnment:STATus[:VALue]? on page 366.
- The execution can be aborted using the command, see :INPut:ALIGnment:EYE:ABORt on page 358.

This SCPI is applicable for M8043A (command only) and M8046A.

Example :INP:VOLT:WIND:DET 'M2.DataIn'

:INPut[:VOLTage]:WINDow:HISTogram?

Syntax :INPut[:VOLTage]:WINDow:HISTogram? 'Identifier'

Input Parameters 'identifier': 'M*.DataIn'

Description This query returns the input voltage window covered in the histogram measured in the data input signal path after Clock Data Recovery and FFE.

This query is supported by M8043A.

Example :INP:VOLT:WIND:HIST? 'M1.DataIn'

:INPut[:VOLTage][:NRZ]:THReshold[?]

Syntax :INPut[:VOLTage][:NRZ]:THReshold 'identifier', <NRf>

:INPut[:VOLTage][:NRZ]:THReshold? 'identifier'

Input Parameters <identifier>: Any valid data input identifier. M*.DataIn1, M*.DataIn2 or M*.DataIn

Description The command specifies the threshold voltage of a RX input connector.

This query returns the present setting.

This SCPI is applicable for M8043A and M8046A.

Example :INP:THR 'M2.DataIn1', 0.2

:INPut[:VOLTage]:PAM3[:SYMBol]:THReshold(1|2)[?]

Syntax :INPut[:VOLTage]:PAM3[:SYMBol]:THReshold(1|2) 'identifier', <NRf>

:INPut[:VOLTage]:PAM3[:SYMBol]:THReshold(1|2)? 'identifier'

Input Parameters <identifier>: Any valid data input identifier. M*.DataIn

Description The command controls the decision threshold of the PAM3 decoder within the data input's input window. The setting of the PAM3 decision thresholds always refers to the actual input voltage range.

The 2 decision thresholds are applied as described in the following table:

Threshold	Symbol
1	Decides between symbol 0 and symbol 1.
2	Decides between symbol 1 and symbol 2.

This query returns the present state.

This SCPI is applicable for M8046A.

It is best practice to configure the 2 decision thresholds in a combined command transaction (see the example) to minimize hardware re-configuration duration.

Example :INP:PAM3:SYMB:THR1 'M2.DataIn',-100e-3;THR2 'M2.DataIn',-5e-3

:INPut[:VOLTage]:PAM4[:SYMBol]:THReshold(1|2|3)[?]

Syntax :INPut[:VOLTage]:PAM4[:SYMBol]:THReshold(1|2|3) 'identifier', <NRf>

:INPut[:VOLTage]:PAM4[:SYMBol]:THReshold(1|2|3)? 'identifier'

Input Parameters <identifier>: Any valid data input identifier. M*.DataIn

Description The command controls the decision threshold of the PAM4 decoder within the data input's input window. The setting of the PAM4 decision thresholds always refers to the actual input voltage range.

The 3 decision thresholds are applied as described in the following table:

Threshold	Symbol
1	Decides between symbol 0 and symbol 1.
2	Decides between symbol 1 and symbol 2.
3	Decides between symbol 2 and symbol 3.

This query returns the present state.

This SCPI is applicable for M8043A and M8046A.

It is best practice to configure the 3 decision thresholds in a combined command transaction (see the example) to minimize hardware re-configuration duration.

Example :INP:PAM4:SYMB:THR1 'M2.DataIn',-100e-3;THR2 'M2.DataIn',-5e-3;THR3 'M2.DataIn',100e-3

:INPut[:VOLTage]:PAM6[:SYMBol]:THReshold(1|2|3|4|5)[?]

Syntax :INPut[:VOLTage]:PAM6[:SYMBol]:THReshold(1|2|3|4|5) 'identifier', <NRf>

:INPut[:VOLTage]:PAM6[:SYMBol]:THReshold(1|2|3|4|5)? 'identifier'

Input Parameters <identifier>: Any valid data input identifier that supports PAM6 (e.g.:'INF*.DataIn1').

Description The command controls the decision threshold of the PAM-6 decoder within the data input's input window. The setting of the PAM-6 decision thresholds always refers to the actual input voltage range.

The 5 decision thresholds are applied as described in the following table:

Threshold	Symbol
1	Decides between symbol 0 and symbol 1.
2	Decides between symbol 1 and symbol 2.
3	Decides between symbol 2 and symbol 3.
4	Decides between symbol 3 and symbol 4.
5	Decides between symbol 4 and symbol 5.

This query returns the present state.

The SCPI is applicable for UXR real-time oscilloscope that supports PAM6.

It is best practice to configure the 5 decision thresholds in a combined command transaction (see example below) to minimize hardware reconfiguration duration.

Example INP:PAM6:SYMB:THR1 'INF.DataIn',1e-3;THR2 'INF.DataIn',2e-3;THR3 'INF.DataIn',3e-3;THR4 'INF.DataIn',4e-3;THR5 'INF.DataIn',5e-3

:INPut[:VOLTage]:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7)[?]

Syntax :INPut[:VOLTage]:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7) 'identifier', <NRf>

:INPut[:VOLTage]:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7)? 'identifier'

Input Parameters Any valid data input identifier that supports PAM6 (e.g. :'INF*.DataIn1').

Description The command controls the decision threshold of the PAM-8 decoder within the data input's input window. The setting of the PAM-8 decision thresholds always refers to the actual input voltage range.

The 7 decision thresholds are applied as described in the following table:

Threshold	Symbol
1	Decides between symbol 0 and symbol 1.
2	Decides between symbol 1 and symbol 2.
3	Decides between symbol 2 and symbol 3.
4	Decides between symbol 3 and symbol 4.
5	Decides between symbol 4 and symbol 5.
6	Decides between symbol 5 and symbol 6.
7	Decides between symbol 6 and symbol 7.

This query returns the present state.

The SCPI is applicable for UXR real-time oscilloscope that supports PAM8.

It is best practice to configure the 7 decision thresholds in a combined command transaction (see example below) to minimize hardware reconfiguration duration

Example INP:PAM8:SYMB:THR1 'INF.DataIn',1e-3;THR2 'INF.DataIn',2e-3;THR3 'INF.DataIn',3e-3;THR4 'INF.DataIn',4e-3;THR5 'INF.DataIn',5e-3;THR6 'INF.DataIn',6e-3;THR7 'INF.DataIn',7e-3

:INPut:CMPower[?]

Syntax :INPut:CMPower 'Identifier', <NRf>

:INPut:CMPower? 'Identifier'

Input Parameters <identifier>: 'INF*.DataIn1'

Description The command sets the common mode power of the selected data input.

This query returns the present setting.

The SCPI is applicable for UXR real-time oscilloscope.

Example :INP:CMP 'INF1.DataIn1', -0.003

:INPut:POWer:WINDow[?]

Syntax :INPut:POWer:WINDow 'Identifier',<NRf>

:INPut:POWer:WINDow? 'Identifier'

Input Parameters <identifier>: 'INF*.DataIn1'

Description The command defines the input power window. The best resolution is achieved if the power window is just big enough to cover the provided (single ended) amplitude.

This query returns the present setting.

The SCPI is applicable for UXR real-time oscilloscope.

Example :INP:POW:WIND 'INF1.DataIn1'

:INPut:POWer:WINDow:ALLocate[?]

Syntax :INPut:POWer:WINDow:ALLocate 'Identifier',<NRf>

:INPut:POWer:WINDow:ALLocate? 'Identifier'

Input Parameters <identifier>: 'INF*.DataIn1'

Description The command defines how much of the input power window shall be used by the received signal.

This controls how the auto-set algorithm calculates the input power window size based on the measured signal amplitude. Normally the input power window is chosen larger than the signal amplitude to avoid nonlinear distortions at the peak power levels.

When working with small signal amplitudes, the BER results of the real-time oscilloscopes may be better if the received signal is using the input power range is used to a larger extent, than what the oscilloscope choses during the built-in vertical auto-scale operation.

Changing this parameter is not directly changing the input power range but has effect on the next vertical auto-scale operation.

This query returns the present setting.

The SCPI is applicable for UXR real-time oscilloscope.

Example :INP:POW:WIND:ALL 'INF1.DataIn1',75

:INPut:POWer[:NRZ]:THReshold[?]

Syntax :INPut:POWer[:NRZ]:THReshold 'Identifier',<NRf>

:INPut:POWer[:NRZ]:THReshold? 'Identifier'

Input Parameters <identifier>: 'INF*.DataIn1'

Description The command specifies the threshold power of a RX input connector (decision threshold).

This query returns the present setting.

The SCPI is applicable for UXR real-time oscilloscope.

Example :INP:POW:THR INF*.DataIn1,0.0

:INPut:POWer:PAM4[:SYMBol]:THReshold(1|2|3)[?]

Syntax :INPut:POWer:PAM4[:SYMBol]:THReshold(1|2|3) 'identifier', <NRf>

:INPut:POWer:PAM4[:SYMBol]:THReshold(1|2|3)? 'identifier'

Input Parameters <identifier>: 'INF*.DataIn1'

Description The command controls the decision threshold of the PAM-4 decoder within the data input’s input window. The setting of the PAM-4 decision thresholds always refers to the actual input power range.

The 3 decision thresholds are applied as described in the following table:

Threshold	Symbol
1	Decides between symbol 0 and symbol 1.
2	Decides between symbol 1 and symbol 2.
3	Decides between symbol 2 and symbol 3.

This query returns the present state.

The SCPI is applicable for UXR real-time oscilloscope.

It is best practice to configure the 3 decision thresholds in a combined command transaction (see the example) to minimize hardware re-configuration duration.

Example :INP:POW:PAM4:SYMB:THR1 "INF1.DataIn1",1e-3;THR2 "INF1.DataIn1",2e-3;THR3 "INF1.DataIN1",3e-3

:INPut:POWer:PAM6[:SYMBol]:THReshold(1|2|3|4|5)[?]

Syntax :INPut:POWer:PAM6[:SYMBol]:THReshold(1|2|3|4|5) 'identifier', <NRf>

:INPut:POWer:PAM6[:SYMBol]:THReshold(1|2|3|4|5)[?]? 'identifier'

Input Parameters Any valid data input identifier that supports PAM6 (e.g. :'INF*.DataIn1').

Description The command controls the decision threshold of the PAM-6 decoder within the data input's input window. The setting of the PAM-6 decision thresholds always refers to the actual input power range.

The 5 decision thresholds are applied as described in the following table:

Threshold	Symbol
1	Decides between symbol 0 and symbol 1.
2	Decides between symbol 1 and symbol 2.
3	Decides between symbol 2 and symbol 3.
4	Decides between symbol 3 and symbol 4.
5	Decides between symbol 4 and symbol 5.

This query returns the present state.

The SCPI is applicable for UXR real-time oscilloscope that supports PAM6.

It is best practice to configure the 5 decision thresholds in a combined command transaction (see example below) to minimize hardware reconfiguration duration.

Example :INP:POW:PAM6:SYMB:THR1 'INF1.DataIn1',-100e-3;THR2 'INF1.DataIn1'',-5e-3;THR3 'INF1.DataIn1',100e-3;THR4 'INF1.DataIn1',-5e-3;THR5 'INF1.DataIn1',100e-3

:INPut:POWer:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7)[?]

Syntax :INPut:POWer:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7) 'identifier', <NRf>

:INPut:POWer:PAM8[:SYMBol]:THReshold(1|2|3|4|5|6|7)? 'identifier'

Input Parameters Any valid data input identifier that supports PAM6 (e.g. :'INF*.DataIn1').

Description The command controls the decision threshold of the PAM8 decoder within the data input's input window. The setting of the PAM8 decision thresholds always refers to the actual input power range.

The 7 decision thresholds are applied as described in the following table:

Threshold	Symbol
1	Decides between symbol 0 and symbol 1.
2	Decides between symbol 1 and symbol 2.
3	Decides between symbol 2 and symbol 3.
4	Decides between symbol 3 and symbol 4.
5	Decides between symbol 4 and symbol 5.
6	Decides between symbol 5 and symbol 6.
7	Decides between symbol 6 and symbol 7.

This query returns the present state.

The SCPI is applicable for UXR real-time oscilloscope that supports PAM8.

It is best practice to configure the 7 decision thresholds in a combined command transaction (see example below) to minimize hardware reconfiguration duration.

Example :INP:POW:PAM8:SYMB:THR1 'INF1.DataIn1',-100e-3;THR2 'INF1.DataIn1',-5e-3;THR3 'INF1.DataIn1',100e-3;THR4 'INF1.DataIn1',-5e-3;THR5 'INF1.DataIn1',100e-3;THR6 'INF1.DataIn1',-5e-3;THR7 'INF1.DataIn1',100e-3

:INPut:DyRx:State[?]

Syntax	:INPut:DyRx:State 'identifier', <ON|OFF|1|0> :INPut:DyRx:State? 'identifier'
Input Parameters	'identifier' -> 'M*.DataIn' ON|OFF|1|0
Description	The command enables/disables the Rx LTSSM settings on the analyzer module. This query returns the present state. The SCPI is applicable for M8046A.
Example	:INPut:DyRx:State 'M2.DataIn',ON :INPut:DyRx:State? 'M2.DataIn' -> 1

:INPut:LTSSm:FILE[?]

Syntax	:INPut:LTSSm:FILE 'identifier', <RxSettingFile> :INPut:DyRx:State? 'identifier'
Input Parameters	'identifier' -> 'M*.DataIn' <RxSettingFile> -> Defines the Rx settings file for the LTSSM.
Description	The command defines the Rx settings file for the LTSSM. This Rx settings file is in the JSON format. This query returns the Rx settings. The SCPI is applicable for M8046A.
Example	:INPut:LTSSm:FILE 'M2.DataIn','Factory/PCIeLTSSMRxSettings.json' :INPut:LTSSm:FILE? 'M2.DataIn' -> 'PCIeLTSSMRxSettings.json'

:INPut:CDR Subnode

This subnode has the following SCPI structure:

```
:INPut
└:CDR
  ├[:JTF]:LORDer[?]
  ├[:JTF]:CTRL[?]
  ├[:JTF][:STATe][?]
  ├[:JTF]:AUTO[?]
  ├[:JTF]:RELOck
  ├[:JTF]:HIGHTD[?]
  ├[:JTF]:OPTimize
  ├[:JTF]:FIRSt
  │ ├:TDENsity[?]
  │ └:LBANdwidth[?]
  ├[:JTF]:SECond
  │ ├:TDENsity[?]
  │ ├:LBANdwidth[?]
  │ └:PEAKing[?]
  ├[:JTF]:SECond:LSELect:AUTO[?]
  ├[:JTF]:SECond:LSELect[:VALue][?]
  ├:OUTPut[?]
  ├:RCLock[?]
  └:OJTF
    ├:STATe[?]
    └:SECond[?]
      ├:LBANdwidth[?]
      │ └:DIVisor[?]
      └:DFACtor[?]
```

This subnode has the following commands:

Table 67

Name	Description under
:INPut:CDR[:JTF]:LORDer[?]	:INPut:CDR[:JTF]:LORDer[?] on page 347
:INPut:CDR[:JTF]:CTRL[?]	:INPut:CDR[:JTF]:CTRL[?] on page 347
:INPut:CDR[:JTF][:STATe][?]	:INPut:CDR[:JTF][:STATe][?] on page 348
:INPut:CDR[:JTF]:AUTO[?]	:INPut:CDR[:JTF]:AUTO[?] on page 348
:INPut:CDR[:JTF]:RELOck	:INPut:CDR[:JTF]:RELOck on page 349
:INPut:CDR[:JTF]:HIGHTD[?]	:INPut:CDR[:JTF]:HIGHTD[?] on page 349
:INPut:CDR[:JTF]:OPTimize	:INPut:CDR[:JTF]:OPTimize on page 349
:INPut:CDR[:JTF]:FIRSt:TDENsity[?]	:INPut:CDR[:JTF]:FIRSt:TDENsity[?] on page 350
:INPut:CDR[:JTF]:FIRSt:LBANdwidth[?]	:INPut:CDR[:JTF]:FIRSt:LBANdwidth[?] on page 350
:INPut:CDR[:JTF]:SECond:TDENsity[?]	:INPut:CDR[:JTF]:SECond:TDENsity[?] on page 350
:INPut:CDR[:JTF]:SECond:LBANdwidth[?]	:INPut:CDR[:JTF]:SECond:LBANdwidth[?] on page 351
:INPut:CDR[:JTF]:SECond:PEAKing[?]	:INPut:CDR[:JTF]:SECond:PEAKing[?] on page 351
:INPut:CDR[:JTF]:SECond:LSELect[:VALue]	:INPut:CDR[:JTF]:SECond:LSELect[:VALue] on page 352
:INPut:CDR:OUTPut[?]	:INPut:CDR:OUTPut[?] on page 352
:INPut:CDR:RCLock[?]	:INPut:CDR:RCLock[?] on page 353
:INPut:CDR:OJTF[:STATe][?]	:INPut:CDR:OJTF[:STATe][?] on page 354
:INPut:CDR:OJTF:SECond:LBANdwidth[?]	:INPut:CDR:OJTF:SECond:LBANdwidth[?] on page 354
:INPut:CDR:OJTF:SECond:LBANdwidth:DIVisor[?]	:INPut:CDR:OJTF:SECond:LBANdwidth:DIVisor[?] on page 355
:INPut:CDR:OJTF:SECond:DFACtor[?]	:INPut:CDR:OJTF:SECond:DFACtor[?] on page 355

:INPut:CDR[:JTF]:LORDer[?]

Syntax	:INPut:CDR[:JTF]:LORDer 'identifier', <FIRSt|SECond>
	:INPut:CDR[:JTF]:LORDer? '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 applicable for M8041A, M8046A, and M8051A.
Example	:INP:CDR:LORD 'M1.DataIn1', FIRS

:INPut:CDR[:JTF]:CTRL[?]

Syntax	:INPut:CDR[:JTF]:CTRL 'identifier', <MANual|SEQuence>
	:INPut:CDR[:JTF]: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 applicable for M8041A, M8051A, M8046A and M8043A.
Example	:INP:CDR:CTRL 'M1.DataIn1', SEQ

:INPut:CDR[:JTF][:STATe][?]

Syntax	:INPut:CDR[:JTF][:STATe] 'identifier', <ON|OFF|1|0> :INPut:CDR[:JTF][:STATe]? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn' <ON|OFF|1|0>: Enable/disable the CDR.
Return Range	1|0
Description	This command enables or disables CDR. This SCPI is applicable for M8041A, M8051A and M8046A. This command is only available as a query for M8046A.
Example	:INP:CDR 'M1.DataIn1', ON

:INPut:CDR[:JTF]:AUTO[?]

Syntax	:INPut:CDR[:JTF]:AUTO 'identifier', <ON|OFF|1|0> :INPut:CDR[:JTF]: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 applicable for M8062A, M8043A and M8046A.
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[:JTF]:RELOck

Syntax	:INPut:CDR[:JTF]:RELOck 'identifier'
Input Parameters	'identifier': 'M*.DataIn'
Description	This command execute a full CDR relock. This SCPI is applicable for M8062A and M8046A.
Example	:INPut:CDR:RELOck 'M2.DataIn'

:INPut:CDR[:JTF]:HIGHTD[?]

Syntax	:INPut:CDR[:JTF]:HIGHTD 'identifier', <ON|OFF|1|0> :INPut:CDR[:JTF]: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 applicable for M8062A.
Example	:INPut:CDR:HIGHTD 'M2.DataIn', 1

:INPut:CDR[:JTF]:OPTimize

Syntax	:INPut:CDR[:JTF]: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 applicable for M8062A.
Example	:INPut:CDR:OPTimize 'M2.DataIn'

:INPut:CDR[:JTF]:FIRSt:TDENsity[?]

Syntax	:INPut:CDR[:JTF]:FIRSt:TDENsity 'identifier', <NRf>
	:INPut:CDR[:JTF]: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 applicable for M8041A, M8051A, M8043A and M8046A.
Example	:INP:CDR:FIRS:TDEN 'M1.DataIn1', 25

:INPut:CDR[:JTF]:FIRSt:LBANdwidth[?]

Syntax	:INPut:CDR[:JTF]:FIRSt:LBANdwidth 'identifier', <NRf>
	:INPut:CDR[:JTF]: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 applicable for M8041A, M8051A, M8043A and M8046A.
Example	:INP:CDR:FIRS:LBAN 'M1.DataIn1', 4MHz

:INPut:CDR[:JTF]:SECond:TDENsity[?]

Syntax	:INPut:CDR[:JTF]:SECond:TDENsity 'identifier', <NRf>
	:INPut:CDR[:JTF]: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 applicable for M8041A, M8051A and M8046A.

Example :INP:CDR:SEC:TDEN 'M1.DataIn1', 25

:INPut:CDR[:JTF]:SECond:LBANdwidth[?]

Syntax	:INPut:CDR[:JTF]:SECond:LBANdwidth 'identifier', <NRf>
	:INPut:CDR[:JTF]: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 applicable for M8041A, M8051A and M8046A.
Example	:INP:CDR:SEC:LBAN 'M1.DataIn1', 4MHz

:INPut:CDR[:JTF]:SECond:PEAKing[?]

Syntax	:INPut:CDR[:JTF]:SECond:PEAKing 'identifier', <NRf>
	:INPut:CDR[:JTF]: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 applicable for M8041A, M8051A and M8046A.
Example	:INP:CDR:SEC:PEAK 'M1.DataIn1', 0.5

:INPut:CDR[:JTF]:SECond:LSELect[:VALue]

Syntax :INPut:CDR[:JTF]:SECond:LSELect[:VALue] 'identifier', <LOOP1 | LOOP2 | LOOP3 | LOOP4>

:INPut:CDR[:JTF]:SECond:LSELect[:VALue]? 'identifier'

Identifier 'M*.DataIn'

Description This command specifies 2 loop transition frequency and thereby the JTF peaking.

This query returns the present setting.

This SCPI is applicable for M8046A.

Example :INP:CDR:SEC:LSEL 'M2.DataIn', LOOP2

:INPut:CDR:OUTPut[?]

Syntax :INPut:CDR:OUTPut 'identifier', <ON|OFF|1|0>

:INPut:CDR:OUTPut? 'identifier'

Identifier 'M*.DataIn'

Description This command enables or disables the recovered clock signals at the Recovered Clock Output.

This query returns the present setting.

This SCPI is applicable for M8046A.

Example :INP:CDR:OUTP 'M2.DataIn', ON

:INPut:CDR:RCLock[?]

Syntax :INPut:CDR:RCLock 'identifier', <INTernal | E4GHz | E8GHz | E16GHz>

:INPut:CDR:RCLock? 'identifier'

Identifier 'identifier': 'M*.DataIn'

INTernal | E4GHz | E8GHz | E16GHz

Description This command selects the CDR reference clock which can be used for SSC tracking.

This query returns the present setting.

This SCPI is applicable for M8043A.

Example :INP:CDR:RCL 'M3.DataIn', E8GHz

:INP:CDR:RCL? 'M3.DataIn' -> E8GHz

:INPut:CDR:OJTF Subnode

The :INPut:CDR:OJTF subnode allows controlling the clock recovery on an error detector channel that is based on a real-time oscilloscope.

:INPut:CDR:OJTF[:STATe][?]

Syntax :INPut:CDR:OJTF[:STATe] <identifier>,{OFF | ON | 0 | 1}

:INPut:CDR:OJTF[:STATe]? <identifier>

Identifier 'INF*.DataIn1'

Description This command enables or disables the use of the clock recovery. For the current versions of M8070B, the clock recovery will always be enabled and cannot be disabled.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:CDR:OJTF:STAT 'INF1.DataIn1',ON

:INPut:CDR:OJTF:SECond:LBANdwidth[?]

Syntax :INPut:CDR:OJTF:SECond:LBANdwidth 'identifier', <NRf>

:INPut:CDR:OJTF:SECond:LBANdwidth? 'identifier'

Identifier 'INF*.DataIn1'

Description This command configures the loop bandwidth of the clock recovery. For more details, refer the corresponding command in the *Keysight Infiniium Oscilloscopes Programmer’s Guide*.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:CDR:OJTF:SEC:LBAN 'INF1.DataIn1',12.5e6

:INPut:CDR:OJTF:SECond:LBANdwidth:DIVisor[?]

Syntax :INPut:CDR:OJTF:SECond:LBANdwidth:DIVisor 'identifier', <NRf>

:INPut:CDR:OJTF:SECond:LBANdwidth:DIVisor? 'identifier'

Identifier 'INF*.DataIn1'

Description This command configures the loop bandwidth of the CDR by a symbol rate divisor. The resulting loop bandwidth will be the channel’s symbol rate divided by the loop bandwidth divisor.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:CDR:OJTF:SEC:LBAN:DIV 'INF1.DataIn1',1667

:INPut:CDR:OJTF:SECond:DFACtor[?]

Syntax :INPut:CDR:OJTF:SECond:DFACtor 'identifier', <NRf>

:INPut:CDR:OJTF:SECond:DFACtor? 'identifier'

Identifier 'INF*.DataIn1'

Description This command configures the damping factor of the clock recovery. For more details, refer the corresponding command in the *Keysight Infiniium Oscilloscopes Programmer’s Guide*.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:CDR:OJTF:SEC:DFAC 'INF1.DataIn1',0.707

:INPut:ALIGnment Subnode

This subnode has the following SCPI structure:

```
:INPut
└:ALIGnment
  ├:EYE
  │ ├[:AUTO]
  │ ├:ACENter
  │ ├:TCENter
  │ ├:ABORt
  │ ├:THReshold[?]
  │ └:RESult
  │   ├ [:NRZ][:HEIGth]?
  │   ├ :WIDTh?
  │   ├ [:NRZ]:THReshold?
  │   ├ :PAM3:HEIGht(1|2)?
  │   ├ :PAM3:THReshold(1|2)?
  │   ├ :PAM4:HEIGht(1|2|3)?
  │   ├ :PAM4:THReshold(1|2|3)?
  │   ├ :PAM6:HEIGht(1|2|3|4|5)?
  │   ├ :PAM6:THReshold(1|2|3|4|5)?
  │   ├ :PAM8:HEIGht(1|2|3|4|5|6|7)?
  │   ├ :PAM8:THReshold(1|2|3|4|5|6|7)?
  │   ├ :DELay?
  │   ├ :POLarity?
  │   └ :HISTogram
  │     ├ :BIN[:DATA]?
  │     ├ :BIN:BLOCk[:DATA]?
  │     └ :REFResh
  └:STATus[:VALue]?
```

This subnode has the following commands:

Table 68

Name	Description under
:EYE[:AUTO]	:INPut:ALIGnment:EYE[:AUTO] on page 357
:EYE:ACENter	:INPut:ALIGnment:EYE:ACENter on page 358
:EYE:TCENter	:INPut:ALIGnment:EYE:TCENter on page 358
:EYE:ABORt	:INPut:ALIGnment:EYE:ABORt on page 358
:EYE:THReshold[?]	:INPut:ALIGnment:EYE:THReshold[?] on page 359
:EYE:RESult[:NRZ][HEIGht]?	:INPut:ALIGnment:EYE:RESult[:NRZ][:HEIGht]? on page 359

Name	Description under
:EYE:RESult:WIDTh?	:INPut:ALIGnment:EYE:RESult:WIDTh? on page 359
:EYE:RESult[:NRZ]:THReshold?	:INPut:ALIGnment:EYE:RESult[:NRZ]:THReshold? on page 360
:EYE:RESult:PAM3:HEIGht(1|2)?	:INPut:ALIGnment:EYE:RESult:PAM3:HEIGht(1|2)? on page 360
:EYE:RESult:PAM3:THReshold(1|2)?	:INPut:ALIGnment:EYE:RESult:PAM3:THReshold(1|2)? on page 360
:EYE:RESult:PAM4:HEIGht(1|2|3)?	:INPut:ALIGnment:EYE:RESult:PAM4:HEIGht(1|2|3)? on page 361
:EYE:RESult:PAM4:THReshold(1|2|3)?	:INPut:ALIGnment:EYE:RESult:PAM4:THReshold(1|2|3)? on page 361
:EYE:RESult:PAM6:HEIGht(1|2|3|4|5)?	:INPut:ALIGnment:EYE:RESult:PAM6:HEIGht(1|2|3|4|5)? on page 362
:EYE:RESult:PAM6:THReshold(1|2|3|4|5)?	:INPut:ALIGnment:EYE:RESult:PAM6:THReshold(1|2|3|4|5)? on page 362
:EYE:RESult:PAM8:HEIGht(1|2|3|4|5|6|7)?	:INPut:ALIGnment:EYE:RESult:PAM8:HEIGht(1|2|3|4|5|6|7)? on page 362
:EYE:RESult:PAM8:THReshold(1|2|3|4|5|6|7)?	:INPut:ALIGnment:EYE:RESult:PAM8:THReshold(1|2|3|4|5|6|7)? on page 363
:EYE:RESult:DELay?	:INPut:ALIGnment:EYE:RESult:DELay? on page 363
:EYE:RESult:POLarity?	:INPut:ALIGnment:EYE:RESult:POLarity? on page 364
:EYE:RESult:HISTogram:BIN[:DATA]?	:INPut:ALIGnment:EYE:RESult:HISTogram:BIN[:DATA]? on page 364
:EYE:RESult:HISTogram:BIN:BLOCk[:DATA]?	:INPut:ALIGnment:EYE:RESult:HISTogram:BIN:BLOCk[:DATA]? on page 364
:EYE:RESult:HISTogram:REFResh	INPut:ALIGnment:EYE:RESult:HISTogram:REFResh on page 365
:INPut:ALIGnment:STATus[:VALue]?	:INPut:ALIGnment:STATus[:VALue]? on page 366

:INPut:ALIGnment:EYE[:AUTO]

Syntax :INPut:ALIGnment:EYE[:AUTO] 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description This command starts an auto alignment.

This SCPI is applicable for M8041A, M8051A, M8043A, M8046A and real-time oscilloscope based error detector channels.

Example :INP:ALIG:EYE 'M1.DataIn1'

:INPut:ALIGnment:EYE:ACENter

Syntax	:INPut:ALIGnment:EYE:ACENter 'identifier'
Input Parameters	'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'
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 applicable for M8041A, M8051A, M8043A, M8046A, M8062A and real-time oscilloscope based error detector channels.
Example	:INP:ALIG:EYE:ACEN 'M1.DataIn1'

:INPut:ALIGnment:EYE:TCENter

Syntax	:INPut:ALIGnment:EYE:TCENter 'identifier'
Input Parameters	'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'
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 applicable for M8041A, M8051A, M8043A, M8046A and real-time oscilloscope based error detector channels.
Example	:INP:ALIG:EYE:TCEN 'M1.DataIn1'

:INPut:ALIGnment:EYE:ABORt

Syntax	:INPut:ALIGnment:EYE:ABORt 'identifier'
Input Parameters	'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'
Description	This command interrupts and aborts an alignment. This SCPI is applicable for M8041A, M8051A, M8043A, M8046A and real-time oscilloscope based error detector channels.
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', 'M*.DataIn2' or 'M*DataIn' <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 applicable for M8041A, M8051A, M8062A, M8043A and M8046A.
Example	:INP:ALIG:EYE:THR 'M1.DataIn1', 1E-3

:INPut:ALIGnment:EYE:RESult[:NRZ][:HEIGht]?

Syntax	:INPut:ALIGnment:EYE:RESult[:NRZ][:HEIGht]? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'
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 applicable for M8041A, M8051A, M8062A, M8043A and M8046A.
Example	:INP:ALIG:EYE:RES? 'M1.DataIn1'

:INPut:ALIGnment:EYE:RESult:WIDTh?

Syntax	:INPut:ALIGnment:EYE:RESult:WIDTh? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'
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 applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :INP:ALIG:EYE:RES:WIDT? 'M1.DataIn1'

:INPut:ALIGnment:EYE:RESult[:NRZ]:THReshold?

Syntax :INPut:ALIGnment:EYE:RESult[:NRZ]:THReshold? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description Returns the threshold detected at the last alignment run.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :INP:ALIG:EYE:RES:THR? 'M1.DataIn1'

:INPut:ALIGnment:EYE:RESult:PAM3:HEIGht(1|2)?

Syntax :INPut:ALIGnment:EYE:RESult:PAM3:HEIGht(1|2)? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description This query returns 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.

The suffix addresses the height of the lower eye (1) or upper eye (2). If the suffix is omitted, then the returned value is for the lower eye of the PAM3 signal.

This SCPI is applicable for M8046A.

Example :INP:ALIG:EYE:RES:PAM3:HEIG1? 'M1.DataIn1'

:INPut:ALIGnment:EYE:RESult:PAM3:THReshold(1|2)?

Syntax :INPut:ALIGnment:EYE:RESult:PAM3:THReshold(1|2)? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description This query returns the threshold detected at the last alignment run. The suffix addresses the threshold of the lower eye (1) or upper eye (2). If the suffix is omitted, then the returned value is for the lower eye of the PAM3 signal.

This SCPI is applicable for M8046A.

Example :INP:ALIG:EYE:RES:PAM3:THR2? 'M1.DataIn1'

:INPut:ALIGnment:EYE:RESult:PAM4:HEIGht(1|2|3)?

Syntax :INPut:ALIGnment:EYE:RESult:PAM4:HEIGht(1|2|3)? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description This query returns 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.

The suffix addresses the height of the lower eye (1), middle eye (2) or upper eye (3). If the suffix is omitted, then the returned value is for the lower eye of the PAM4 signal.

This SCPI is applicable for M8043A and M8046A.

Example :INP:ALIG:EYE:RES:PAM4:HEIG1? 'M1.DataIn1'

:INPut:ALIGnment:EYE:RESult:PAM4:THReshold(1|2|3)?

Syntax :INPut:ALIGnment:EYE:RESult:PAM4:THReshold(1|2|3)? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description This query returns the threshold detected at the last alignment run. The suffix addresses the threshold of the lower eye (1), middle eye (2) or upper eye (3). If the suffix is omitted, then the returned value is for the lower eye of the PAM4 signal.

This SCPI is applicable for M8043A and M8046A.

Example :INP:ALIG:EYE:RES:PAM4:THR2? 'M1.DataIn1'

:INPut:ALIGnment:EYE:RESult:PAM6:HEIGht(1|2|3|4|5)?

Syntax
:INPut:ALIGnment:EYE:RESult:PAM6:HEIGht(1|2|3|4|5)? 'identifier'

Input Parameters
'identifier': Any valid data input identifier that supports PAM6 (e.g. :'INF*.DataIn1')

Description
This query returns the value of data amplitude that puts the 2/3 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.

The suffix addresses the height of the lower eye (1), middle eyes (2, 3, 4) or upper eye (5). If the suffix is omitted, then the returned value is for the lower eye of the PAM6 signal.

The SCPI is applicable for UXR real-time oscilloscope that supports PAM6.

Example
:INP:ALIG:EYE:RES:PAM6:HEIG1? 'INF1.DataIn1'

:INPut:ALIGnment:EYE:RESult:PAM6:THReshold(1|2|3|4|5)?

Syntax
:INPut:ALIGnment:EYE:RESult:PAM6:THReshold(1|2|3|4|5)? 'identifier'

Input Parameters
'identifier': Any valid data input identifier that supports PAM6 (e.g. :'INF*.DataIn1')

Description
This query returns the threshold detected at the last alignment run. The suffix addresses the threshold of the lower eye (1), middle eyes (2, 3, 4) or upper eye (5). If the suffix is omitted, then the returned value is for the lower eye (1) of the PAM6 signal.

The SCPI is applicable for UXR real-time oscilloscope that supports PAM6.

Example
:INP:ALIG:EYE:RES:PAM6:THR2? 'INF1.DataIn1'

:INPut:ALIGnment:EYE:RESult:PAM8:HEIGht(1|2|3|4|5|6|7)?

Syntax
:INPut:ALIGnment:EYE:RESult:PAM8:HEIGht(1|2|3|4|5|6|7)? 'identifier'

Input Parameters
'identifier': Any valid data input identifier that supports PAM8 (e.g. :'INF*.DataIn1')

Description
This query returns the value of data amplitude that puts the 3/4 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.

The suffix addresses the height of the lower eye (1), middle eyes (2, 3, 4, 5, 6) or upper eye (7). If the suffix is omitted, then the returned value is for the lower eye of the PAM8 signal.

The SCPI is applicable for UXR real-time oscilloscope that supports PAM8.

Example :INP:ALIG:EYE:RES:PAM8:HEIG1? 'INF1.DataIn1'

:INPut:ALIGnment:EYE:RESult:PAM8:THReshold(1|2|3|4|5|6|7)?

Syntax :INPut:ALIGnment:EYE:RESult:PAM8:THReshold(1|2|3|4|5|6|7)? 'identifier'

Input Parameters 'identifier': Any valid data input identifier that supports PAM6 (e.g. :'INF*.DataIn1')

Description This query returns the threshold detected at the last alignment run. The suffix addresses the threshold of the lower eye (1), middle eyes (2, 3, 4, 5, 6) or upper eye (7). If the suffix is omitted, then the returned value is for the lower eye (0) of the PAM-8 signal.

The SCPI is applicable for UXR real-time oscilloscope that supports PAM8.

Example :INP:ALIG:EYE:RES:PAM8:THR2? 'INF1.DataIn1'

:INPut:ALIGnment:EYE:RESult:DELay?

Syntax :INPut:ALIGnment:EYE:RESult:DELay? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description Returns the delay detected at the last alignment run.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

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', 'M*.DataIn2' or 'M*DataIn'
Return Range	NOR|INV
Description	Returns the polarity of the signal detected at the last alignment run. This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.
Example	:INP:ALIG:EYE:RES:POL? 'M1.DataIn1'

:INPut:ALIGnment:EYE:RESult:HISTogram:BIN[:DATA]?

Syntax	:INPut:ALIGnment:EYE:RESult:HISTogram:BIN[:DATA]? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1' or 'M*.DataIn2'
Description	Returns the sampling point alignment histogram bin as a comma separated list. This SCPI is only applicable for M8046A.
Example	:INP:ALIG:EYE:HIST:BIN? 'M1.DataIn1'

:INPut:ALIGnment:EYE:RESult:HISTogram:BIN:BLOCk[:DATA]?

Syntax	:INPut:ALIGnment:EYE:RESult:HISTogram:BIN:BLOCk[:DATA]? 'identifier'
Input Parameters	'identifier': 'M*.DataIn1' or 'M*.DataIn2'
Description	Returns the sampling point alignment histogram bin as definite length block. This SCPI is only applicable for M8046A.
Example	:INP:ALIG:EYE:RES:HIST:BIN:BLOC? 'M1.DataIn1'

INPut:ALIGnment:EYE:RESult:HISTogram:REFResh

Syntax	:INPut:ALIGnment:EYE:RESult:HISTogram:REFResh 'identifier'
Input Parameters	'identifier': 'M*.DataIn1' or 'M*.DataIn2'
Description	This command refreshes the histogram data immediately. It also refreshes the histogram displayed in the Sampling Point dialog. This SCPI is only applicable for M8046A.
Example	INP:ALIG:EYE:RES:HIST:REFR 'M2.DataIn1'

:INPut:ALIGnment:STATus Subnode

This subnode reports alignment specific status information and offers an alternative way of retrieving the same status information that is available via the :STATus:INSTrument:ALIGnment Subnode on page 164.

:INPut:ALIGnment:STATus[:VALue]?

Syntax :INPut:ALIGnment:STATus[:VALue]? 'Identifier'

Input Parameters 'identifier': Any error detector location and group name. E.g. 'M1.DataIn1', 'M2.DataIn', 'INF1.DataIn1'

Description This command retrieves alignment status information for an error detector or error detector group.

This command can be used to retrieve information about the alignment procedure that is working on the addressed error detector channel, or a group of error detectors. This command is reporting the status independently of the executing alignment procedure. There is no differentiation in respect to different types of alignments.

This command returns the following values:

- INPROGRESS - At least one addressed error detector is executing an alignment procedure.
- ABORTED - The alignment procedure has been aborted by the user before all aligning error detector channels finished the procedure.
- FAILED - All addressed error detector channels have finished the alignment procedure, but at least one was unable to finish successfully.
- SUCCESS - The addressed error detector channels have successfully finished the alignment procedure. Error detector channels that are addressed by the command but return UNKNOWN state, do not contribute to the result.
- UNKNOWN - None of the addressed error detectors has executed an alignment procedure yet.

NOTE This command cannot be used in the same compound SCPI statement that is starting the alignment procedure. If this command is sent to the instrument right after the command that is initiating the alignment procedure, then the returned value is undefined and can reflect the state that was valid before the alignment. This can be overcome by reading any value from the instrument (e.g. *ESE?), except for *OPC? as this will block until the alignment procedure is finished.

Example :INP:ALIGN:STAT? 'M1.DataIn1'

:INPut:EQUalization Subnode

This subnode controls the equalizer settings on a data input. Depending on the product number different methods of equalizations are available.

:INPut:EQUalization:CURSor subsystem

The commands of the INPut:EQUalization:CURSor subsystem configure the Feed Forward Equalizer of an M8046A error detector module.

This equalizer can operate in the following three modes:

- Equalizer Presets with cable compensation

  The equalizer gain is controlled as Equalizer Level, and the losses of the standard cabling is automatically included in the equalizer setting.
- Manual coefficient entry with cable compensation

  The coefficients of the Feed Forward Equalizer can be entered manually. Additionally, the losses of the standard cabling is automatically included in the equalizer setting.
- Manual coefficient entry

  The coefficients of the Feed Forward Equalizer can be entered manually. No cable losses are automatically compensated. Use this mode when using non-standard cabling.

Additional to the manual control of the equalizer mode and coefficient settings, the coefficients can be automatically determined from the received signal.

This subnode has the following SCPI structure:

```
:INPut
└:EQUalization
  ├:CURSor
  │ ├:MODE[?]
  │ ├:MAGNitude[?]
  │ └:DETect
  ├[:PREset][?]
  ├[:PREset]:TYPE[?]
  ├[:PREset]:GAIN[?]
  ├[:PREset]:LEVel[?]
  ├:CHANnel:LOSS[?]
  ├:COMPensation
  │ ├:ADJustment[?]
  │ └:BIAS[?]
  ├:FFEQualizer
  │ ├:STATe[?]
  │ ├:NTAPs[?]
  │ ├:NPRecursor[?]
  │ └:TAP
  │   └:AUTO
  │     ├:INCLude[?]
  │     └:WIDTh[?]
  ├:CTLequalizer
  │ ├:ACGain[?]
  │ ├:DCGain[?]
  │ ├:DCGain2[?]
  │ ├:STATe[?]
  │ ├:NUMPoles[?]
  │ ├:P1[?]
  │ ├:P2[?]
  │ ├:P3[?]
  │ ├:P4[?]
  │ ├:P5[?]
  │ ├:P6[?]
  │ ├:Z1[?]
  │ ├:Z2[?]
  │ ├:LF[?]
  │ ├[:PRESet][?]
  │ ├[:PRESet]:TYPE[?]
  │ └[:PRESet]:GAIN[?]
  ├: DFEQualizer
  │ ├:STATe
  │ ├:NTAPs
  │ └:TAP
  │   ├:ALGorithm[?]
  │   └:AUTO
  │     └:INCLude[?]
  └:PATequalizer:STATe[?]
```

This subnode has the following commands:

Table 69

Name	Description under
:INPut:EQUalization:CURSor:MODE[?]	:INPut:EQUalization:CURSor:MODE[?] on page 371
:INPut:EQUalization:CURSor:MAGNitude[0|1|2|3|...|15][?]	:INPut:EQUalization:CURSor:MAGNitude[0|1|2|3|...|15][?] on page 371
:INPut:EQUalization:CURSor:DETect	:INPut:EQUalization:CURSor:DETect on page 372
:INPut:EQUalization[:PRESet][?]	:INPut:EQUalization[:PRESet][?] on page 373
INPut:EQUalization[:PRESet]:TYPE[?]	:INPut:EQUalization[:PRESet]:TYPE[?] on page 374
:INPut:EQUalization[:PRESet]:GAIN[?]	:INPut:EQUalization[:PRESet]:GAIN[?] on page 374
:INPut:EQUalization[:PRESet]:LEVel[?]	:INPut:EQUalization[:PRESet]:LEVel[?] on page 375
:INPut:EQUalization:CHANnel:LOSS[?]	:INPut:EQUalization:CHANnel:LOSS[?] on page 376
:INPut:EQUalization:COMPensation:ADJustment[?]	:INPut:EQUalization:COMPensation:ADJustment[?] on page 376
:INPut:EQUalization:COMPensation:BIAS[?]	:INPut:EQUalization:COMPensation:BIAS[?] on page 377
:INPut:EQUalization:FFEQualizer:STATe[?]	:INPut:EQUalization:STATe[?] on page 378
:INPut:EQUalization:FFEQualizer:NTAPs[?]	:INPut:EQUalization:FFEQualizer:NTAPs[?] on page 378
:INPut:EQUalization:FFEQualizer:NPRecursor[?]	:INPut:EQUalization:FFEQualizer:NPRecursor[?] on page 379
:INPut:EQUalization:FFEQualizer:TAP:AUTO[?]	:INPut:EQUalization:FFEQualizer:TAP:AUTO[?] on page 379
:INPut:EQUalization:FFEQualizer:TAP:AUTO:INCLude[?]	:INPut:EQUalization:FFEQualizer:TAP:AUTO:INCLude[?] on page 379
:INPut:EQUalization:FFEQualizer:TAP:WIDTh[?]	:INPut:EQUalization:FFEQualizer:TAP:WIDTh[?] on page 380
:INPut:EQUalization:CTLequalizer:ACGain[?]	:INPut:EQUalization:CTLequalizer:ACGain[?] on page 381
:INPut:EQUalization:CTLequalizer:DCGain[?]	:INPut:EQUalization:CTLequalizer:DCGain[?] on page 381
:INPut:EQUalization:CTLequalizer:DCGain2[?]	:INPut:EQUalization:CTLequalizer:DCGain2[?] on page 382
:INPut:EQUalization:CTLequalizer:STATe[?]	:INPut:EQUalization:CTLequalizer:STATe[?] on page 382
:INPut:EQUalization:CTLequalizer:NUMPoles[?]	:INPut:EQUalization:CTLequalizer:NUMPoles[?] on page 383
:INPut:EQUalization:CTLequalizer:P1[?]	:INPut:EQUalization:CTLequalizer:P1[?] on page 383
:INPut:EQUalization:CTLequalizer:P2[?]	:INPut:EQUalization:CTLequalizer:P2[?] on page 384

Name	Description under
:INPut:EQUalization:CTLequalizer:P3[?]	:INPut:EQUalization:CTLequalizer:P3[?] on page 384
:INPut:EQUalization:CTLequalizer:P4[?]	:INPut:EQUalization:CTLequalizer:P4[?] on page 385
:INPut:EQUalization:CTLequalizer:P5[?]	:INPut:EQUalization:CTLequalizer:P5[?] on page 385
:INPut:EQUalization:CTLequalizer:P6[?]	:INPut:EQUalization:CTLequalizer:P6[?] on page 386
:INPut:EQUalization:CTLequalizer:Z1[?]	:INPut:EQUalization:CTLequalizer:Z1[?] on page 386
:INPut:EQUalization:CTLequalizer:Z2[?]	:INPut:EQUalization:CTLequalizer:Z2[?] on page 387
:INPut:EQUalization:CTLequalizer:LF[?]	:INPut:EQUalization:CTLequalizer:LF[?] on page 387
:INPut:EQUalization:CTLequalizer:PRESet[?]	:INPut:EQUalization:CTLequalizer:PRESet[?] on page 388
:INPut:EQUalization:DFEQualizer:STATe[?]	:INPut:EQUalization:DFEQualizer:STATe[?] on page 390
:INPut:EQUalization:DFEQualizer:NTAPs[?]	:INPut:EQUalization:DFEQualizer:NTAPs[?] on page 391
:INPut:EQUalization:DFEQualizer:TAP:ALGorithm[?]	:INPut:EQUalization:DFEQualizer:TAP:ALGorithm[?] on page 391
:INPut:EQUalization:DFEQualizer:TAP:AUTO	:INPut:EQUalization:DFEQualizer:TAP:AUTO on page 392
:INPut:EQUalization:DFEQualizer:TAP:AUTO:INCLude[?]	:INPut:EQUalization:DFEQualizer:TAP:AUTO:INCLude[?] on page 392
:INPut:EQUalization:PATequalizer:STATe[?]	:INPut:EQUalization:PATequalizer:STATe[?] on page 393

:INPut:EQUalization:CURSor:MODE[?]

Syntax :INPut:EQUalization:CURSor:MODE 'Identifier', {PRESet | MCABle | MANual}

:INPut:EQUalization:CURSor:MODE? 'Identifier'

Input Parameters 'identifier': 'M*.DataIn'
PRESet | MCABle | MANual

Description This command the equalization control and behavior. The following three equalization modes are available for operation:

- PRESet

  Select the EQ gain level via presets with automatic cable compensation.
- MCABle

  Manual entry of FFE coefficients with automatic cable compensation.
- MANual

  Manual entry of FFE coefficients without cable compensation.

This SCPI is applicable for M8043A and M8046A.

Example :INP:EQU:CURS:MODE 'M2.DataIn',MAN

:INP:EQU:CURS:MODE? 'M2.DataIn' -> MAN

:INPut:EQUalization:CURSor:MAGNitude[0|1|2|3|...|15][?]

Syntax :INPut:EQUalization:CURSor:MAGNitude[0|1|2|3|...|15] 'Identifier', <NRf>

:INPut:EQUalization:CURSor:MAGNitude[0|1|2|3|...|15]? 'Identifier'

Input Parameters 'identifier': 'M*.DataIn'

Description This command allows to set the equalizer coefficients manually. There are 16 filter coefficients, numbered from 0 to 15. Coefficient 2 is the main-cursor and cannot be changed.

The available value range depends on the coefficient and is defined as follows:

Coefficient 0: -0.25 .. + 0.25

Coefficient 1: -0.5 .. + 0.5

Coefficient 2: 1.0

Coefficient 3: -0.5 .. +0.5

Coefficient 4: -0.25 .. + 0.25

Coefficient 5: -0.125 .. +0.125

Coefficient 6 to 15: -0.0625 .. +0.0625

Additionally, the sum of all 16 coefficients may not be 0.

This SCPI is applicable for M8043A and M8046A.

Example :INP:EQU:CURS:MAGN3 'M2.DataIn',-0.15

:INP:EQU:CURS:MAGN3? 'M2.DataIn' -> -0.15

:INPut:EQUalization:CURSor:DETect

Syntax :INPut:EQUalization:CURSor:DETect 'Identifier'

Input Parameters 'identifier': 'M*.DataIn'

Description This command automatically determines the correct FFE coefficients for the current input signal..

- The execution progress can be queried using the command, see :INPut:ALIGnment:STATus[:VALue]? on page 366.
- The execution can be aborted using the command, see :INPut:ALIGnment:EYE:ABORt on page 358.

This SCPI is applicable for M8043A and M8046A.

Example :INP:EQU:CURS:DET 'M2.DataIn'

: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 -6dB
- P2 – PCIe 8Gb/s -9dB
- P3 – PCIe 8Gb/s -12dB
- P4 – USB 5Gb/s -3.5dB
- P5 – PCIe 16Gb/s -6dB
- P6 – PCIe 16Gb/s -9dB
- P7 – PCIe 16Gb/s -12dB
- P8 – USB 10Gb/s 0dB
- P9 – USB 10Gb/s -3dB
- P10 – USB 10Gb/s -6dB

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 applicable for M8041A, M8051A 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

:INP:EQU? 'M1.DataIn1' -> P1

:INPut:EQUalization[:PRESet]:TYPE[?]

Syntax
:INPut:EQUalization[:PRESet]:TYPE 'Identifier', <OFF | P1 | P2 | P3 | P4 | P5 | P6 | P7>

:INPut:EQUalization[:PRESet]:TYPE? 'Identifier'

Input Parameters
'identifier': 'M*.DataIn'
<OFF | P1 | P2 | P3 | P4 | P5 | P6 | P7>

Return Range
OFF | P1 | P2 | P3 | P4 | P5 | P6 | P7

Description
This command is used to select from the following equalization presets:

- OFF - Disables equalization
- P1 - PCIe 8Gb/s
- P2 - PCIe 16Gb/s
- P3 - PCIe 32Gb/s
- P4 - PCIe 64Gb/s
- P5 - PCIe 128Gb/s
- P6 - CEI-112G
- P7 - IEEE 802.3 200Gaui-4

To change the gain use the command :INPut:EQUalization[:PRESet]:GAIN[?] on page 374 (M8043A only).

This SCPI is applicable for M8043A.

Example
:INP:EQU:TYPE 'M1.DataIn',P1

:INP:EQU:TYPE? 'M1.DataIn' -> P1

:INPut:EQUalization[:PRESet]:GAIN[?]

Syntax
:INPut:EQUalization[:PRESet]:GAIN 'Identifier', <NRf>

:INPut:EQUalization[:PRESet]:GAIN? 'Identifier'

Input Parameters
'identifier': 'M*.DataIn'

Description
This command defines the gain for the selected preset. You can set a gain

with an accuracy of 0.1 dB. The minimum and maximum gain is defined in the standard for each preset type.

This command is only accessible if the equalization preset type is not set to

OFF, that is, an equalizer preset type was set before. (For details, see :INPut:EQUalization[:PRESet]:TYPE[?] on page 374).

This SCPI is applicable for M8043A.

Example :INP:EQU:GAIN 'M1.DataIn', -6.0

:INP:EQU:GAIN? 'M1.DataIn' -> -6.00000000000000E+00

:INPut:EQUalization[:PRESet]:LEVel[?]

Syntax :INPut:EQUalization[:PRESet]:LEVel 'Identifier', <NRf>

:INPut:EQUalization[:PRESet]:LEVel ? 'Identifier'

Input Parameters 'identifier': 'M*.DataIn'
The specified number has a range from 0 – 120.

Return Range 0 – 120

Description This command controls the integrated and adjustable equalization of M8046A, to compensate for loss characteristic of the back channel. This feature is available for NRZ and PAM4 signals.

Equalization can be adjusted from no equalization (0) to maximum equalization (120) in equidistant steps. The available range is restricted to 0 - 55 for PAM4 signals.

This SCPI is applicable for M8043A and M8046A.

Example :INP:EQU:LEV 'M1.DataIn',12

:INP:EQU:LEV? 'M1.DataIn'

:INPut:EQUalization:CHANnel:LOSS[?]

Syntax

:INPut:EQUalization:CHANnel:LOSS 'identifier', <Value>

:INPut:EQUalization:CHANnel:LOSS? 'identifier'

Input Parameters

Identifier: 'M*.DataIn'

<Value>: The estimated loss in dB

Description

This command sets a nominal estimated loss on the back channel in dB that the user expects to compensate.

This query returns the current value.

This SCPI is applicable for M8047C.

Example

Command

:INP:EQU:CHAN:LOSS 'M1.DataIn',30

Query

:INP:EQU:CHAN:LOSS 'M1.DataIn'

:INPut:EQUalization:COMPensation:ADJustment[?]

Syntax

:INPut:EQUalization:COMPensation:ADJustment 'identifier', <Value>

:INPut:EQUalization:COMPensation:ADJustment? 'identifier'

Input Parameters

Identifier: 'M*.DataIn'

<Value>: The compensation adjustment factor

Description

This command allows the user to fine-tune the compensation with respect to the given estimated channel loss.

This query returns the current value.

This SCPI is applicable for M8047C.

Example

Command

:INP:EQU:COMP:ADJ 'M1.DataIn',30

Query

:INP:EQU:COMP:ADJ 'M1.DataIn'

:INPut:EQUalization:COMPensation:BIAS[?]

Syntax

:INPut:EQUalization:COMPensation:BIAS 'identifier', <CompensationBiasFactor>

:INPut:EQUalization:COMPensation:BIAS? 'identifier'

Input Parameters

Identifier: 'M*.DataIn'

<CompensationBiasFactor>: NOBS | X1 | X2 | X3

The enum elements correspond to the following bias currents:

- NOBS - 0 mA
- X1 - 0.68 mA
- X2 - 1.36 mA
- X3 - 2.03 mA

Description

This command defines the factor of the bias current.

This query returns the current value.

This SCPI is applicable for M8047C.

Example

Command

:INP:EQU:COMP:BIAS 'M1.DataIn',X3

Query

:INP:EQU:COMP:BIAS 'M1.DataIn'

:INPut:EQUalization:FFEQualizer Subnode

This subnode configures a Feed-Forward Equalizer on a data input. This subnode is available for real-time oscilloscope based error detector channels. It requires the existence for the Equalization license (option DEQ) on the real-time oscilloscope. If the license is not available, then equalization will not be offered in M8070B for the data input.

:INPut:EQUalization:STATe[?]

Syntax: :INPut:EQUalization:STATe <identifier>,{OFF|ON|0|1}

:INPut:EQUalization:STATe? <identifier>

Identifier: 'INF*.DataIn1'

Description: This command enables or disables the use of the FFE algorithm on the given data input.

This query returns the present setting.

This SCPI is applicable for M8043A and real-time oscilloscope based error detector channels.

Example: For real-time oscilloscope - :INP:EQU:FFEQ:STAT 'INF1.DataIn1',ON

For M8043A - :INP:EQU:STAT 'M3.DataIn',ON

:INPut:EQUalization:FFEQualizer:NTAPs[?]

Syntax: :INPut:EQUalization:FFEQualizer:NTAPs <identifier>,<number>

:INPut:EQUalization:FFEQualizer:NTAPs? <identifier>

Identifier: 'INF*.DataIn1'

Description: This command sets the number of taps to be used in the FFE algorithm.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example: :INP:EQU:FFEQ: NTAPs 'INF1.DataIn1',6

:INPut:EQUalization:FFEQualizer:NPRecursor[?]

Syntax :INPut:EQUalization:FFEQualizer:NPRecursor <identifier>,<number>

:INPut:EQUalization:FFEQualizer:NPRecursor? <identifier>

Identifier 'INF*.DataIn1'

Description This command sets the number of precursor taps to be used in the FFE algorithm.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:FFEQ:NPR 'INF1.DataIn1',2

:INPut:EQUalization:FFEQualizer:TAP:AUTO[?]

Syntax :INPut:EQUalization:FFEQualizer:TAP:AUTO <identifier>

:INPut:EQUalization:FFEQualizer:TAP:AUTO? <identifier>

Identifier 'INF*.DataIn1'

Description This command starts the FFE tap optimization. Be sure to first specify the number of taps.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:FFEQ:TAP:AUTO 'INF1.DataIn1'

:INPut:EQUalization:FFEQualizer:TAP:AUTO:INCLude[?]

Syntax :INPut:EQUalization:FFEQualizer:TAP:AUTO:INCLude <identifier>, <0|1|OFF|ON>

:INPut:EQUalization:FFEQualizer:TAP:AUTO:INCLude? <identifier>

Identifier 'INF*.DataIn1'

Description This command controls if the FFE tap optimization is done during a full sample point alignment. If this is set to OFF, then the tap coefficients must be set manually, or by executing the :INPut:EQUalization:FFEQualizer:TAP:AUTO command.

FFE tap optimization is part of the full sample point optimization by default. Use this command to disable tap optimization in scenarios where the input signal is too distorted to produce reliable tap optimization results.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:FFEQ:TAP:AUTO:INCL 'INF1.DataIn1', ON

:INP:EQU:FFEQ:TAP:AUTO:INCL? 'INF1.DataIn1'

ON

:INPut:EQUalization:FFEQualizer:TAP:WIDTh[?]

Syntax :INPut:EQUalization:FFEQualizer:TAP:WIDTh <identifier>,<width>

:INPut:EQUalization:FFEQualizer:TAP:WIDTh? <identifier>

Identifier 'INF*.DataIn1'

Description The tap width sets the Eye Width field for the FFE tap optimization. Setting the width to 0.0 means the optimization is only preformed at the location of the clock. Setting the width to 1.0 means the entire acquisition is used in the optimization. The default value for FFE is 0.33. For more information on this parameter, refer to the *N5461A Infiniium Serial Data Equalization User's Guide*

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:FFEQ:TAP:WIDT 'INF1.DataIn1',0.1

:INP:EQU:FFEQ:TAP:WIDT? 'INF1.DataIn1' -> 0.1

:INPut:EQUalization:CTLequalizer Subnode

The CTLequalizer subnode is configuring the use of the Constant Time Linear Equalizer (CTLE) of a real-time oscilloscope that is used as error detector in M8070B.

:INPut:EQUalization:CTLequalizer:ACGain[?]

Syntax	:INPut:EQUalization:CTLequalizer:ACGain <identifier>,<NRf>
	:INPut:EQUalization:CTLequalizer:ACGain? <identifier>
Identifier	'INF*.DataIn1'
Input Parameters	<identifier>
	<NRf> - AC Gain
Description	This command sets the AC Gain parameter for the Continuous Time Linear Equalization when '2-poles with AC gain' is selected for the "# of Poles" parameter.
	This query returns the present setting.
	This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:CTL:ACG 'INF1.DataIn1',1

:INPut:EQUalization:CTLequalizer:DCGain[?]

Syntax	:INPut:EQUalization:CTLequalizer:DCGain <identifier>,<NRf>
	:INPut:EQUalization:CTLequalizer:DCGain? <identifier>
Identifier	'INF*.DataIn1'
Input Parameters	<identifier>
	<NRf> - DC Gain
Description	This command sets the DC Gain parameter for the Continuous Time Linear Equalization.
	This query returns the present setting.
	This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:CTL:DCG 'INF1.DataIn1',1

:INPut:EQUalization:CTLequalizer:DCGain2[?]

Syntax :INPut:EQUalization:CTLequalizer:DCGain2 <identifier>,<NRf>

:INPut:EQUalization:CTLequalizer:DCGain2? <identifier>

Identifier 'INF*.DataIn1'

Input Parameters <identifier>

<NRf> - DC Gain2

Description This command sets the DC Gain2 parameter for the Continuous Time Linear Equalization.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:CTL:DCG2 'INF1.DataIn1',1

:INPut:EQUalization:CTLequalizer:STATe[?]

Syntax :INPut:EQUalization:CTLequalizer:STATe <identifier>, {OFF | ON | 0 | 1}

:INPut:EQUalization:CTLequalizer:STATe? <identifier>

Identifier 'INF*.DataIn1'

Input Parameters <identifier>

OFF | ON | 0 | 1

Description This command enables or disables the use of the Continuous Time Linear Equalization.

The CTLE cannot be used at the same time as the Feed Forward Equalization (FFE).

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:CTL:STAT 'INF1.DataIn1',ON

:INPut:EQUalization:CTLequalizer:NUMPoles[?]

Syntax :INPut:EQUalization:CTLequalizer:NUMPoles <identifier>, { P2Z1 | P3Z1 | P3Z2 | P2ACG | P4Z1 | P6Z2 | LFG2 }

:INPut:EQUalization:CTLequalizer:NUMPoles? <identifier>

Identifier 'INF*.DataIn1'

Input Parameters <identifier>

P2Z1 | P3Z1 | P3Z2 | P2ACG | P4Z1 | P6Z2 | LFG2

Description This command selects either a 2 Pole 1 Zero, 3 Pole 1 Zero, 3 Pole 2 Zeros, 2 Poles and AC gain (for6 USB 3.1), 4 Pole 1 Zero, 6 Pole 2 Zero, 2 Poles and 2 gains Continuous Time Linear Equalization (CTLE).

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:CTL:NUMP 'INF1.DataIn1',P3Z1

:INPut:EQUalization:CTLequalizer:P1[?]

Syntax :INPut:EQUalization:CTLequalizer:P1 <identifier>, <NRf>

:INPut:EQUalization:CTLequalizer:P1? <identifier>

Identifier 'INF*.DataIn1'

Input Parameters <identifier>

<NRf> - Pole 1 frequency

Description This command sets the Pole 1 frequency for the Continuous Time Linear Equalization.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:CTL:P1 'INF1.DataIn1',1e9

:INPut:EQUalization:CTLequalizer:P2[?]

Syntax	:INPut:EQUalization:CTLequalizer:P2 <identifier>, <NRf>
	:INPut:EQUalization:CTLequalizer:P2? <identifier>
Identifier	'INF*.DataIn1'
Input Parameters	<identifier>
	<NRf> - Pole 2 frequency
Description	This command sets the Pole 2 frequency for the Continuous Time Linear Equalization.
	This query returns the present setting.
	This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:CTL:P2 'INF1.DataIn1',4e9

:INPut:EQUalization:CTLequalizer:P3[?]

Syntax	:INPut:EQUalization:CTLequalizer:P3 <identifier>, <NRf>
	:INPut:EQUalization:CTLequalizer:P3? <identifier>
Identifier	'INF*.DataIn1'
Input Parameters	<identifier>
	<NRf> - Pole 3 frequency
Description	This command sets the Pole 3 frequency for the Continuous Time Linear Equalization.
	This query returns the present setting.
	This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:CTL:P3 'INF1.DataIn1',4e9

:INPut:EQUalization:CTLequalizer:P4[?]

Syntax	:INPut:EQUalization:CTLequalizer:P4 <identifier>, <NRf>
	:INPut:EQUalization:CTLequalizer:P4? <identifier>
Identifier	'INF*.DataIn1'
Input Parameters	<identifier>
	<NRf> - Pole 4 frequency
Description	This command sets the Pole 4 frequency for the Continuous Time Linear Equalization.
	This query returns the present setting.
	This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:CTL:P4 'INF1.DataIn1',4e9

:INPut:EQUalization:CTLequalizer:P5[?]

Syntax	:INPut:EQUalization:CTLequalizer:P5 <identifier>, <NRf>
	:INPut:EQUalization:CTLequalizer:P5? <identifier>
Identifier	'INF*.DataIn1'
Input Parameters	<identifier>
	<NRf> - Pole 5 frequency
Description	This command sets the Pole 5 frequency for the Continuous Time Linear Equalization.
	This query returns the present setting.
	This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:CTL:P5 'INF1.DataIn1',4e9

:INPut:EQUalization:CTLequalizer:P6[?]

Syntax	:INPut:EQUalization:CTLequalizer:P6 <identifier>, <NRf>
	:INPut:EQUalization:CTLequalizer:P6? <identifier>
Identifier	'INF*.DataIn1'
Input Parameters	<identifier>
	<NRf> - Pole 6 frequency
Description	This command sets the Pole 6 frequency for the Continuous Time Linear Equalization.
	This query returns the present setting.
	This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:CTL:P6 'INF1.DataIn1',4e9

:INPut:EQUalization:CTLequalizer:Z1[?]

Syntax	:INPut:EQUalization:CTLequalizer:Z1 <identifier>, <NRf>
	:INPut:EQUalization:CTLequalizer:Z1? <identifier>
Identifier	'INF*.DataIn1'
Input Parameters	<identifier>
	<NRf> - Zero Frequency 1
Description	This command sets first zero frequency for the 2-pole and 3-pole Continuous Time Linear Equalization.
	This query returns the present setting.
	This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:CTL:Z1 'INF1.DataIn1',650e6

:INPut:EQUalization:CTLequalizer:Z2[?]

Syntax	:INPut:EQUalization:CTLequalizer:Z2 <identifier>, <NRf> :INPut:EQUalization:CTLequalizer:Z2? <identifier>
Identifier	'INF*.DataIn1'
Input Parameters	<identifier> <NRf> - Zero Frequency 2
Description	This command sets second zero frequency for the 2-pole and 3-pole Continuous Time Linear Equalization. This query returns the present setting. This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:CTL:Z2 'INF1.DataIn1',4e9

:INPut:EQUalization:CTLequalizer:LF[?]

Syntax	:INPut:EQUalization:CTLequalizer:LF <identifier>, <LF_frequency> :INPut:EQUalization:CTLequalizer:LF? <identifier>
Identifier	'INF*.DataIn1'
Input Parameters	<identifier> <LF_frequency> A real number.
Description	This command sets the LF Frequency parameter when "2 Pole, 2 Gain" is selected for the "# of Poles" option. This query returns the present setting. This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:CTL:LF 'INF1.DataIn1',200

:INPut:EQUalization:CTLequalizer:PRESet[?]

Syntax :INPut:EQUalization:CTLequalizer:PRESet <identifier>, “preset-name”

:INPut:EQUalization:CTLequalizer:PRESet? <identifier>

Identifier 'INF*.DataIn1'

Input Parameters <identifier>

“preset-name” - preset name, see the supporting preset names.

Supported preset names are:

- "CEI-56G-VSR-NRZ (1 dB)"
- "CEI-56G-VSR-NRZ (2 dB)"
- "CEI-56G-VSR-NRZ (3 dB)"
- "CEI-56G-VSR-NRZ (4 dB)"
- "CEI-56G-VSR-NRZ (5 dB)"
- "CEI-56G-VSR-NRZ (6 dB)"
- "CEI-56G-VSR-NRZ (7 dB)"
- "CEI-56G-VSR-NRZ (8 dB)"
- "CEI-56G-VSR-NRZ (9 dB)"
- "CEI-56G-VSR-NRZ (10 dB)"
- "CEI-56G-VSR-NRZ (11 dB)"
- "CEI-56G-VSR-NRZ (12 dB)"
- "IEEE 802.3bs CDAUI-8 (1 dB)"
- "IEEE 802.3bs CDAUI-8 (1.5 dB)"
- "IEEE 802.3bs CDAUI-8 (2 dB)"
- "IEEE 802.3bs CDAUI-8 (2.5 dB)"
- "IEEE 802.3bs CDAUI-8 (3 dB)"
- "IEEE 802.3bs CDAUI-8 (3.5 dB)"
- "IEEE 802.3bs CDAUI-8 (4 dB)"
- "IEEE 802.3bs CDAUI-8 (4.5 dB)"
- "IEEE 802.3bs CDAUI-8 (5 dB)"
- "IEEE 802.3bs CDAUI-8 (5.5 dB)"
- "IEEE 802.3bs CDAUI-8 (6 dB)"
- "IEEE 802.3bs CDAUI-8 (6.5 dB)"
- "IEEE 802.3bs CDAUI-8 (7 dB)"
- "IEEE 802.3bs CDAUI-8 (7.5 dB)"
- "IEEE 802.3bs CDAUI-8 (8 dB)"

- "IEEE 802.3bs CDAUI-8 (8.5 dB)"
- "IEEE 802.3bs CDAUI-8 (9 dB)"
- "OIF CEI-28G-VSR (0 dB)"
- "OIF CEI-28G-VSR (1 dB)"
- "OIF CEI-28G-VSR (2 dB)"
- "OIF CEI-28G-VSR (3 dB)"
- "OIF CEI-28G-VSR (4 dB)"
- "OIF CEI-28G-VSR (5 dB)"
- "OIF CEI-28G-VSR (6 dB)"
- "OIF CEI-28G-VSR (7 dB)"
- "OIF CEI-28G-VSR (8 dB)"
- "OIF CEI-28G-VSR (9 dB)"
- "PCIe Gen3 (6 dB)"
- "PCIe Gen3 (7 dB)"
- "PCIe Gen3 (8 dB)"
- "PCIe Gen3 (9 dB)"
- "PCIe Gen3 (10 dB)"
- "PCIe Gen3 (11 dB)"
- "PCIe Gen3 (12 dB)"
- "PCIe Gen4 (6 dB)"
- "PCIe Gen4 (7 dB)"
- "PCIe Gen4 (8 dB)"
- "PCIe Gen4 (9 dB)"
- "PCIe Gen4 (10 dB)"
- "PCIe Gen4 (11 dB)"
- "PCIe Gen4 (12 dB)"
- "PCIe Gen5 (6 dB)"
- "PCIe Gen5 (7 dB)"
- "PCIe Gen5 (8d B)"
- "PCIe Gen5 (9 dB)"
- "PCIe Gen5 (10 dB)"
- "PCIe Gen5 (11 dB)"
- "PCIe Gen5 (12 dB)"
- "USB3 Gen1 Short Channel"
- "USB3 Gen1 Long Channel"

- "USB3 Gen2 (0 dB)"
- "USB3 Gen2 (1 dB)"
- "USB3 Gen2 (2 dB)"
- "USB3 Gen2 (3 dB)"
- "USB3 Gen2 (4 dB)"
- "USB3 Gen2 (5 dB)"
- "USB3 Gen2 (6 dB)"
- "IEEE 802.3 ck"
- "IEEE 802.3 dj"

Description: This command selects a predefined setting of the Continuous Time Linear Equalizer (CTLE).

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example: :INP:EQU:CTL:PRES 'INF1.DataIn1', "PCIe Gen3 (6dB)"

:INPut:EQUalization:DFEQualizer Subnode

The DFEQualizer subnode configures a Decision Feedback Equalization on a data input. This sub-system is available for real-time oscilloscope based error detector channels. It requires the existence for the Equalization license (option DEQ) on the real-time oscilloscope. If the license is not available, then equalization will not be offered in M8070B for the data input.

:INPut:EQUalization:DFEQualizer:STATe[?]

Syntax: :INPut:EQUalization:DFEQualizer:STATe <identifier>, <OFF|ON|0|1>

:INPut:EQUalization:DFEQualizer:STATe? <identifier>

Input Parameters: <identifier> - 'INF*.DataIn1'

<OFF|ON|0|1>

Description: This command enable or disable use of the DFE algorithm on the given data input.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:DFEQ:STAT 'INF1.DataIn1', ON

:INP:EQU:DFEQ:STAT? 'INF1.DataIn1' -> 1

:INPut:EQUalization:DFEQualizer:NTAPs[?]

Syntax :INPut:EQUalization:DFEQualizer:NTAPs <identifier>,<number>

:INPut:EQUalization:DFEQualizer:NTAPs? <identifier>

Input Parameters <identifier> - 'INF*.DataIn1'

<number> - Number of taps to be used in the DFE algorithm

Description This command sets the number of taps to be used in the DFE algorithm.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:DFEQ: NTAPs 'INF1.DataIn1',6

:INP:EQU:DFEQ: NTAPs? 'INF1.DataIn1' -> 6

:INPut:EQUalization:DFEQualizer:TAP:ALGorithm[?]

Syntax :INPut:EQUalization:DFEQualizer:TAP:ALGorithm <identifier>, LSEF | PMAX

:INPut:EQUalization:DFEQualizer:TAP:ALGorithm? <identifier>

Input Parameters <identifier> - 'INF*.DataIn1'

LSEF | PMAX

Description This command allows the DFE filter to set the following algorithms:

- Least Square Error Fit
- PMAX Normalized based on pulse response.

This query returns the present setting.

This SCPI is applicable for real-time oscilloscope based error detector channels.

Example :INP:EQU:DFEQ:TAP:ALG 'INF1.Dataln1', LSEF

:INP:EQU:DFEQ:TAP:ALG? 'INF1.Dataln1' -> LSEF

:INPut:EQUalization:DFEQualizer:TAP:AUTO

Syntax	:INPut:EQUalization:DFEQualizer:TAP:AUTO <identifier>
Input Parameters	<identifier> - 'INF*.DataIn1'
Description	This command starts the DFE tap optimization. Ensure to first specify the number of taps. This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:DFEQ:TAP:AUTO 'INF1.DataIn1'

:INPut:EQUalization:DFEQualizer:TAP:AUTO:INCLude[?]

Syntax	:INPut:EQUalization:DFEQualizer:TAP:AUTO:INCLude <identifier>, <0|1|OFF|ON> :INPut:EQUalization:DFEQualizer:TAP:AUTO:INCLude? <identifier>
Input Parameters	<identifier> - 'INF*.DataIn1' <0|1|OFF|ON>
Description	This command controls if the DFE tap optimization is done during a full sample point alignment. If this is set to OFF, then the tap coefficients must be set manually, or by executing the :INPut:EQUalization:DFEQualizer:TAP:AUTO on page 392 command. DFE tap optimization is part of the full sample point optimization by default. Use this command to disable tap optimization in scenarios where the input signal is too distorted to produce reliable tap optimization results. This query returns the present setting. This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:DFEQ:TAP:AUTO:INCL 'INF1.DataIn1',OFF :INP:EQU:DFEQ:TAP:AUTO:INCL 'INF1.DataIn1' -> 0

:INPut:EQUalization:PATequalizer:STATe[?]

Syntax	:INPut:EQUalization:PATequalizer:STATe 'identifier', <1|0|ON|OFF>
	:INPut:EQUalization:PATequalizer:STATe? 'identifier'
Input Parameters	<identifier> - 'INF*.DataIn1'
	<0|1|OFF|ON>
Description	This command enables or disables the pattern-based equalizer.
	This query returns the present state.
	This SCPI is applicable for real-time oscilloscope based error detector channels.
Example	:INP:EQU:PAT:STAT 'INF1.DataIn1', 1
	:INP:EQU:PAT:STAT? 'INF1.DataIn1'

:INPut:SQUElch Subnode

The Squelch Threshold feature allows you to adjust the squelch threshold values in link training applications (e.g. USB link training).

This subnode has the following SCPI structure:

```
:SQUElch
├─:MODe[?]
├─:VALue[?]
└─:PREset
```

This subnode has the following commands:

Table 70

Name	Description under
:INPut:SQUElch:MODe[?]	:INPut:SQUElch:MODe[?] on page 394
:INPut:SQUElch:VALue[?]	:INPut:SQUElch:VALue[?] on page 395
:INPut:SQUElch:VALue:PREset	:INPut:SQUElch:VALue:PREset on page 395

:INPut:SQUElch:MODe[?]

Syntax :INPut:SQUElch:MODe <identifier>,{MANual | AUTo}

:INPut:SQUElch:MODe? <identifier>

Identifier 'M*.DataIn1' or 'M*.DataIn2'

Description This command selects the squelch mode to either manual or auto on a data input.

This query returns the present setting.

This SCPI is applicable for M8041A and M8051A.

Example :INP:SQUE:MOD 'M2.DataIn2',AUT

:INP:SQUE:MOD? 'M2.DataIn2' -> AUT

:INPut:SQUElch:VALue[?]

Syntax :INPut:SQUElch:VALue <identifier>,<number>

:INPut:SQUElch:VALue? <identifier>

Identifier 'M*.DataIn1' or 'M*.DataIn2'

Description This command sets the squelch threshold value in range from 0 ... 255.

This query returns the present setting.

This SCPI is applicable for M8041A and M8051A.

Example :INP:SQUE:VAL 'M2.DataIn2',40

:INP:SQUE:VAL? 'M2.DataIn2' -> 40

:INPut:SQUElch:VALue:PREset

Syntax :INPut:SQUElch:VALue:PREset <identifier>

Identifier 'M*.DataIn1' or 'M*.DataIn2'

Description This command sets the default squelch threshold value.

This SCPI is applicable for M8041A and M8051A.

Example :INP:SQUE:VAL:PRE 'M2.DataIn2'

:INPut:PROBe Subnode

The PROBe subsystem allows you to detect the probe connection on the real time scope channel

This subnode has the following SCPI structure:

```
:PROBe
├─:ID?
└─:DETect
  └─:AUTO
```

This subnode has the following commands:

Table 71

Name	Description under
:INPut:PROBe:ID?	:INPut:PROBe:ID? on page 396
:INPut:PROBe:DETect:AUTO	:INPut:PROBe:DETect:AUTO on page 396

:INPut:PROBe:ID?

Syntax :INPut:PROBe:ID? <identifier>

Identifier 'INF*.DataIn1'

Return Value <probe_id> A string of alphanumeric characters. Some of the possible returned values are: N7005A, N7004A, No Probe.

Description This query returns the type of probe attached to the specified channel of a real-time oscilloscope.

This SCPI is applicable to only UXR real-time oscilloscope.

Example :INP:PROB:ID? 'INF1.DataIn1' -> N7005A

:INPut:PROBe:DETect:AUTO

Syntax :INPut:PROBe:DETect:AUTO <identifier>

Identifier 'INF*.DataIn1'

Description This command detects the probe connection on real-time oscilloscope channel.

This SCPI is applicable to only UXR real-time oscilloscope.

Example :INP:PROB:DET:AUTO 'INF1.DataIn1'

:INPut:DEMBedding Subnode

This subnode has the following SCPI structure:

```
:DEMBedding
├:ISYMbol
│ ├:EXTernal[?]
│ ├:FREQuency[?]
│ ├:STATe[?]
│ ├:FILE[?]
│ ├:OPERator[?]
│ ├:PORT[?]
│ ├:STATe[?]
│ └:WEIGht[?]
├:MODE[?]
└:MEASurement
  ├:STATe[?]
  ├:SBLOck[?]
  └:AUTO
```

This subnode has the following commands:

Table 72

Name	Description under
:INPut:DEMBedding:ISYMbol:CLOSs[?]	:INPut:DEMBedding:ISYMbol:CLOSs[?] on page 398
:INPut: DEMBedding:ISYMbol:EXTernal[?]	:INPut:DEMBedding:ISYMbol:EXTernal[?] on page 398
:INPut: DEMBedding:ISYMbol:FREQuency[?]	:INPut:DEMBedding:ISYMbol:FREQuency[?] on page 399
:INPut: DEMBedding:ISYMbol:STATe[?]	:INPut:DEMBedding:ISYMbol:STATe[?] on page 399
:INPut:DEMBedding[1|2]:FILE[?}	:INPut:DEMBedding[1|2]:FILE[?] on page 400
:INPut:DEMBedding[1|2]:OPERator[?]	:INPut:DEMBedding[1|2]:OPERator[?] on page 400
:INPut:DEMBedding[1|2]:PORT[?]	:INPut:DEMBedding[1|2]:PORT[?] on page 401
:INPut:DEMBedding[1|2]1:STATe[?]	:INPut:DEMBedding[1|2]1:STATe[?] on page 401
:INPut:DEMBedding[1|2]1:WEIGht[?]	:INPut:DEMBedding[1|2]1:WEIGht[?] on page 402
:INPut:DEMBedding:MODE[?]	:INPut:DEMBedding:MODE[?] on page 402
:INPut:DEMBedding:MEASurement:STATe[?]	:INPut:DEMBedding:MEASurement:STATe[?] on page 403
:INPut:DEMBedding:MEASurement:SBLOck[?]	:INPut:DEMBedding:MEASurement:SBLOck[?] on page 403
:INPut:DEMBedding:MEASurement:AUTO	:INPut:DEMBedding:MEASurement:AUTO on page 404

:INPut:DEMBedding:ISYMbol:CLOSs[?]

Command	:INPut:DEMBedding:ISYMbol: CLOSs 'Identifier', <NRf>
Query	:INPut:DEMBedding:ISYMbol: CLOSs? 'Identifier'
Parameters	Identifier: 'M*.DataIn' <NRf> - From 0 dB to 120 dB
Description	This command defines the attenuation of the ISI board at the specified frequency in dB. Make sure the de-embedding mode is configured to 'FILe'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402. This SCPI is applicable for M8043A.
Example	INP:DEMB:ISYM:CLOS 'M1.DataIn', 10 INP:DEMB:ISYM:CLOS? 'M1.DataIn' -> 10

:INPut:DEMBedding:ISYMbol:EXTernal[?]

Command	:INPut:DEMBedding:ISYMbol:EXTernal 'Identifier', <NONE | EB00114 | EB00124 | EB00134 | EB00144 | EB00154 | EB00208 | EB002135 | EB00219 | EB002245 | EB002355 | EB002465 | EB002575 | EB002685 | EB002795 | EB003905 | EB0031125 | EB0031345 | EB0031565 | EB0031785 | EB0032005 | EB0032225 | EB00162 | EB001116 | EB001171 | EB00238 | EB00289 | EB002144 | EB00314 | EB00343 | EB00429 | EB00458 | EB00573 | EB00588>
Query	:INPut:DEMBedding:ISYMbol:EXTernal? 'Identifier'
Parameters	'Identifier': 'M*.DataIn' <NONE | EB00114 | EB00124 | EB00134 | EB00144 | EB00154 | EB00208 | EB002135 | EB00219 | EB002245 | EB002355 | EB002465 | EB002575 | EB002685 | EB002795 | EB003905 | EB0031125 | EB0031345 | EB0031565 | EB0031785 | EB0032005 | EB0032225 | EB00162 | EB001116 | EB001171 | EB00238 | EB00289 | EB002144 | EB00314 | EB00343 | EB00429 | EB00458 | EB00573 | EB00588>
Description	This command selects the external ISI board. Make sure the de-embedding mode is configured to 'FILe'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.

This SCPI is applicable for M8043A.

Example INP:DEMB:ISYM:EXT 'M1.DataIn', EB00114

INP:DEMB:ISYM:EXT? 'M1.DataIn' -> EB00114

:INPut:DEMBedding:ISYMbol:FREQuency[?]

Command :INPut:DEMBedding::ISYMbol:FREQuency 'Identifier', <NRf>

Query :INPut:DEMBedding::ISYMbol:FREQuency? 'Identifier'

Parameters Identifier: 'M*.DataIn'

<NRf> - From 0 MHz to 64 GHz

Description This command defines the frequency, where the ISI board has the specified loss.

Make sure the de-embedding mode is configured to 'FILe'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.

This SCPI is applicable for M8043A.

Example :INP:DEMB:ISYM:FREQ 'M1.DataIn', 32000000000

:INP:DEMB:ISYM:FREQ 'M1.DataIn'? -> 32000000000

:INPut:DEMBedding:ISYMbol:STATe[?]

Command :INPut:DEMBedding:ISYMbol:STATe 'Identifier', <0|1|ON|OFF>

Query :INPut:DEMBedding:ISYMbol:STATe? 'Identifier'

Parameters Identifier: 'M*.DataIn'

0|1|ON|OFF

Return Range 0|1

Description This command enables/disables the compensation of an ISI board.

Make sure the de-embedding mode is configured to 'FILe'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.

This SCPI is applicable for M8043A.

Example :INP:DEMB:ISYM 'M1.DataIn', ON

:INP:DEMB:ISYM? 'M1.DataIn' -> 1

:INPut:DEMBedding[1|2]:FILE[?]

Command :INPut:DEMBedding[1|2]:FILE 'Identifier', <File_Path>

Query :INPut:DEMBedding[1|2]:FILE? 'Identifier'

Parameters Identifier: 'M*.DataIn'

<File_Path> - Path of the S-Parameter file.

Description This command selects the S-Parameter file for deembedding.

Make sure the de-embedding mode is configured to 'FILe'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.

This SCPI is applicable for M8043A.

Example :INP:DEMB1:FILE 'M1.DataIn', 'Factory/ISIBoard/M8049_001_TRACE1.s2p'

:INP:DEMB2:FILE 'M1.DataIn', 'Factory/ISIBoard/M8049_001_TRACE1.s2p'

:INPut:DEMBedding[1|2]:OPERator[?]

Command :INPut:DEMBedding[1|2]:OPERator 'Identifier', <DIFFerential | SEComplement| SENormal>

Query :INPut:DEMBedding[1|2]:OPERator? 'Identifier'

Parameters Identifier: 'M*.DataIn'

<DIFFerential | SEComplement| SENormal>

Description This command selects operator output for S-Parameter profile.

Make sure the de-embedding mode is configured to 'FILe'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.

This SCPI is applicable for M8043A.

Example :INP:DEMB1:OPER 'M1.DataIn', DIFF

:INP:DEMB1:OPER 'M1.DataIn'?, DIFF

:INPut:DEMBedding[1|2]:PORT[?]

Command	:INPut:DEMBedding[1|2]:PORT 'Identifier', <PORT12 | PORT13>
Query	:INPut:DEMBedding[1|2]:PORT? 'Identifier'
Parameters	Identifier: 'M*.DataIn'
	PORT12 | PORT13
Description	This command selects port for S-Parameter profile for external ISI board.
	Make sure the de-embedding mode is configured to 'FILe'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.
	This SCPI is applicable for M8043A.
Example	:INP:DEMB1:PORT 'M1.DataIn', PORT12
	:INP:DEMB1:PORT? 'M1.DataIn' -> PORT12

:INPut:DEMBedding[1|2]1:STATe[?]

Command	:INPut:DEMBedding[1|2]:STATe 'Identifier', <0|1|ON|OFF>
Query	:INPut:DEMBedding[1|2]:STATe? 'Identifier'
Parameters	Identifier: 'M*.DataIn'
	0|1|ON|OFF
Return Range	0|1
Description	This command enables/disables the de-embedding 1 or 2 configuration.
	Make sure the de-embedding mode is configured to 'FILe'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.
	This SCPI is applicable for M8043A.
Example	:INP:DEMB1:STAT 'M1.DataIn', ON
	:INP:DEMB2:STAT 'M1.DataIn', ON

:INPut:DEMBedding[1|2]1:WEIGht[?]

Command	:INPut:DEMBedding[1|2]:WEIGht 'Identifier', <Num>
Query	:INPut:DEMBedding[1|2]:WEIGht? 'Identifier'
Parameters	Identifier: 'M*.DataIn'
	<Num> - From 0 to 10
Description	This command sets the weight for de-embedding 1 or 2.
	Make sure the de-embedding mode is configured to 'FILe'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.
	This SCPI is applicable for M8043A.
Example	:INP:DEMB1:WEIG 'M1.DataIn', 1
	:INP:DEMB1:WEIG? 'M1.DataIn' -> 1

:INPut:DEMBedding:MODE[?]

Command	:INPut:DEMBedding:MODE 'Identifier', <FILe | MEASure>
Query	:INPut:DEMBedding:MODE? 'Identifier'
Parameters	Identifier: 'M*.DataIn'
	FILe | MEASure
Description	This command allows to set the de-embedding mode between 'File' and 'Measure' options.
	Make sure the de-embedding mode is configured to 'MEASure'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.
	This SCPI is applicable for M8043A.
Example	:INP:DEMB:MODE 'M3.DataIn', MEAS
	:INP:DEMB:MODE? 'M3.DataIn' -> MEAS

:INPut:DEMBedding:MEASurement:STATe[?]

Command	:INPut:DEMBedding:MEASurement:STATe 'Identifier', <ON | OFF | 1 | 0>
Query	:INPut:DEMBedding:MEASurement:STATe? 'Identifier'
Parameters	Identifier: 'M*.DataIn'
	ON | OFF | 1 | 0
Description	This command enables or disables the current auto-correction results.
	Make sure the de-embedding mode is configured to 'MEASure'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.
	This SCPI is applicable for M8043A.
Example	:INP:DEMB:MEAS:STAT 'M3.DataIn', ON
	:INP:DEMB:MEAS:STAT? 'M3.DataIn' -> 1

:INPut:DEMBedding:MEASurement:SBLOck[?]

Command	:INPut:DEMBedding:MEASurement:SBLOck 'Identifier', <Seq_Block>
Query	:INPut:DEMBedding:MEASurement:SBLOck? 'Identifier'
Parameters	Identifier: 'M*.DataIn'
	Seq_Block: 1 to 100
Description	This command defines the number of sequence blocks that include the training pattern.
	Make sure the de-embedding mode is configured to 'MEASure'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402.
	This SCPI is applicable for M8043A.
Example	:INP:DEMB:MEAS:SBLO 'M3.DataIn', 5
	:INP:DEMB:MEAS:SBLO? 'M3.DataIn' -> 5

:INPut:DEMBedding:MEASurement:AUTO

Command	:INPut:DEMBedding:MEASurement:AUTO 'Identifier'
Parameters	Identifier: 'M*.DataIn'
Description	This command automatically determines the best de-embedding setting by measuring the current channel response. Make sure the de-embedding mode is configured to 'MEASure'. You can set or query the de-embedding mode by using the command :INPut:DEMBedding:MODE[?] on page 402. To abort the measurement, use :INPut:ALIGnment:EYE:ABORt on page 358. This SCPI is applicable for M8043A.
Example	:INP:DEMB:MEAS:AUTO 'M3.DataIn'

SENSe Subsystem

The SENSe subsystem represents the analyzer's Data In port.

This subsystem has the following commands:

Table 73

Name	Description under
:SENSe:SRATe[?]	:SENSe:SRATe[?] on page 405

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 74

License	Description	Modules
M8043A-A32	Error analysis up to 32.4 GBd for NRZ and PAM4	M8043A
M8046A-A32	Analyzer, one channel, data rate up to 32 GBd, NRZ	M8046A

:SENSe:SRATe[?]

Syntax :SENSe:SRATe <identifier>, <NRf>

:SENSe:SRATe? <identifier>

Input Parameters 'identifier': 'M*.DataIn', 'INF*.DataIn1'

Description This command controls the symbol rate for the data input. The symbol rate can only be controlled directly when the clock source is set to CDR. When Clk In or Sys Clk is selected as the clock source, then the symbol rate can only be read.

This query returns the present setting.

This SCPI is applicable for M8046A and real-time oscilloscope based error detector channels.

This SCPI is applicable as a query for M8043A.

Example :SENS:SRAT 'M2.DataIn', 32e9

:SENS:SRAT? 'M2.DataIn' -> 32e9

CLOCk Subsystem

The CLOCk subsystem is used to detect clock source or to set the clock frequency applied to input or output channel.

This subsystem has the following SCPI structure:

```
:CLOCk
├ [:SOURce][?]
│ └ :CRECovery:SELect[?]
├ :FREQuency [?]
│ └ :MULTiplier[?]
└ :TRACk
  └ [:STATe][?]
```

This subsystem has the following commands:

Table 75

Name	Description under
[:SOURce][?]	:CLOCk[:SOURce][?] on page 408
:CLOCk[:SOURce]:CRECovery:SELect[?]	:CLOCk[:SOURce]:CRECovery:SELect[?] on page 408
:FREQuency [?]	:CLOCk:FREQuency [?] on page 409
:FREQuency:MULTiplier[?]	:CLOCk:FREQuency:MULTiplier[?] on page 409
:TRACk[:STATe][?]	:CLOCk:TRACk[:STATe][?] on page 410

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 76

License	Description	Modules
M8009A-062	Clock Generator two channel 60 GHz	M8009A
M8009A-0G6	Reference Clock Multiplier	M8009A
M8041A-0A2	Second Channel for Analyzer	M8041A
M8041A-0G2	Second Channel for Generator	M8041A
M8041A-C08	BERT one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8042A-G32	32 GBd PG for NRZ and PAM-4	M8042A
M8043A-A32	Error analysis up to 32.4 GBd for NRZ and PAM4	M8043A
M8045A-G32	Pattern generator one channel NRZ, data rate up to 32 GBd (requires remote head, M8057A/B)	M8045A
M8046A-0A4	Clock Recovery for 32 GBd	M8046A
M8046A-A32	Analyzer, one channel, data rate up to 32 GBd, NRZ	M8046A
M8062A-C32	32 Gb/s BERT front-end	M8062A
M8062A-G32	32 Gb/s Pattern generator front-end	M8062A

:CLOCk[:SOURce][?]

Syntax	:CLOCk[:SOURce] 'identifier', <NRf> :CLOCk[:SOURce]? 'identifier'
Input Parameters	'identifier': 'M*.DataOut' and 'M*.DataIn' <NRf>: CDR | CLK | SYS | AUXCLK | ECRevovery
Return Range	CDR | CLK | SYS | AUXCLK | ECR
Description	This command sets the source of the clock signal. It is supported by: • M8062A • M8046A The availability of the different clock sources depends on the product being controlled and possible even on installed product options (e.g. CDR). The following are the different clock sources that can be selected: **CDR** - Clock Recovery **CLK** - Front panel 'CLK IN' connector **SYS** - Internal clock generation **AUXCLK** - Front panel 'AUX CLK IN' connector **ECRecovery** - External clock recovery. Selection is done with the command :CLOCk[:SOURce]:CRECovery:SELect. For details, see :CLOCk[:SOURce]:CRECovery:SELect[?] on page 408. This query returns the present state. This SCPI is applicable for M8062A, M8043A and M8046A.
Example	:CLOCk:SOURce 'M3.DataIn', CLK

:CLOCk[:SOURce]:CRECovery:SELect[?]

Syntax	:CLOCk:CRECovery 'Identifier', 'Clock Recovery Identifier' :CLOCk:CRECovery? 'Identifier'
Input Parameters	'identifier': 'M*.DataIn' 'Clock Recovery Identifier' - ECRecovery
Return Range	ECR
Description	This command select the external clock recovery to be used when using the clock source 'ECRecovery'

This query returns the present state.

This SCPI is applicable for M8046A.

Example :CLOCK:CREC:SEL 'M2.DataIn', 'DCA1.Slot1'

:CLOCk:FREQuency [?]

Syntax CLOCk:FREQuency 'Identifier', <NRf>

CLOCk:FREQuency? 'Identifier'

Input Parameters 'identifier': 'M*.DataOut' or 'M*.DataIn'

Return Range NRf

Description This command sets the clock frequency applied to the output channel.

This query returns the present state.

This SCPI is applicable for M8062A and M8046A.

Example :CLOCK:FREQ 'M3.DataOut', 8.0e9

:CLOCk:FREQuency:MULTiplier[?]

Syntax :CLOCk:FREQuency:MULTiplier 'Identifier', <NRf>

:CLOCk:FREQuency:MULTiplier? 'Identifier'

Input Parameters 'identifier': 'M*.DataIn'

Return Range NRf

Description This command sets the frequency multiplier that is applied to the externally provided clock in order to achieve the desired symbol rate.

Symbol Rate = Multiplier x Clock Frequency

This query returns the present setting.

This SCPI is applicable for M8046A.

Example :CLOCK:FREQ:MULT 'M2.DataIn',2

:CLOCk:TRACk[:STATe][?]

Syntax :CLOCk:TRACk[:STATe] <identifier>, <ON|OFF|1|0>

CLOCk:TRACk[:STATe]? 'Identifier'

Input Parameters 'identifier': 'M*.DataIn', 'INF*.DataIn1'

Description This command enables or disables the automatic system symbol rate tracking. When this is enabled, then the system symbol rate is used as the symbol rate of the addressed functional unit (e.g. DataIn). In this case, the required clock frequency will be determined automatically based on the selected frequency multiplier.

See:

- :CLOCk:FREQuency:MULTiplier[?]
- :[:SENSe]:SRATe[?]

This query returns the present state.

This SCPI is applicable for M8043A, M8046A and real-time oscilloscope based error detector channels.

Example :CLOC:TRAC 'M2.DataIn',ON

N1000 Subsystem

The N1000 subsystem is used to control external clock recovery instruments that are themselves controlled from within the FlexDCA N1000-Series System Software.

NOTE **The minimum required version of the FlexDCA N1000-Series System Software is A.05.61.23.**

It is possible to share one FlexDCA instance among multiple M8070B instances. However, to ensure correct access control, all M8070B instances have to be executed by the same PC.

All the commands used in this subsystem are basically identical with the SCPI commands of FlexDCA N1000-Series System Software. The only difference is that the M8070B does not use a suffix to address specific parts of the instrument, but instead always has the identifier parameter as first argument of the SCPI commands.

Detailed information about the features and SCPI commands of the supported clock recoveries can be found in the corresponding documentation:

N1076A User Guide
http://literature.cdn.keysight.com/litweb/pdf/N1076-90001.pdf?id=2784631

N1010A FlexDCA Online Help
http://www.keysight.com/main/redirector.jspx?action=ref&cname=EDITORIAL&ckey=2038334&lc=ger&cc=DE&nfr=-11143.0.00

N1076B Documentation

https://www.keysight.com/en/pdx-2892129-pn-N1076B/electrical-clock-recovery?nid=-33198.1234885.00&cc=IN&lc=eng

N1078A Documentation

https://www.keysight.com/en/pdx-2906605-pn-N1078A/optical-electrical-clock-recovery?nid=-33198.1238480.00&cc=US&lc=eng

CDR Data Sheet
http://literature.cdn.keysight.com/litweb/pdf/5992-1620EN.pdf?id=2772197

This subsystem has the following SCPI structure:

```
:N1000
├─:EMODules
│  ├─:CONNect[?]
│  └─:SLOT:MODel?
└─:CRECovery
   ├─:CLBandwidth[?]
   ├─:CRATe [?]
   ├─:ELEVel[?]
   ├─:LBWMode[?]
   ├─:LOCKed?
   ├─:LSELect [?]
   │  └─:AUTomatic[?]
   ├─:ODRatio[?]
   │  └─:AUTO[?]
   ├─:RDIVider[?]
   ├─:RELock
   ├─:SOURce[?]
   └─:SSCLock[?]
```

This subsystem has the following commands:

Table 77

Name	Description under
:N1000:EMODules:SLOT:CONNect[?]	:N1000:EMODules:SLOT:CONNect[?] on page 413
:N1000:EMODules:SLOT:MODel?	:N1000:EMODules:SLOT:MODel? on page 414
:N1000:CRECovery:CLBandwidth[?]	:N1000:CRECovery:CLBandwidth[?] on page 414
:N1000:CRECovery:CRATe [?]	:N1000:CRECovery:CRATe? on page 414
:N1000:CRECovery:SOURce[?]	:N1000:CRECovery:ELEVel[?] on page 415
:N1000:CRECovery:LBWMode[?]	:N1000:CRECovery:LBWMode[?] on page 415
:N1000:CRECovery:LOCKed?	:N1000:CRECovery:LOCKed? on page 415
:N1000:CRECovery:LSELect[?]	:N1000:CRECovery:LSELect[?] on page 416

Name	Description under
:N1000:CRECovery:LSELect:AUTomatic[?]	:N1000:CRECovery:LSELect:AUTomatic[?] on page 416
:N1000:CRECovery:ODRatio[?]	:N1000:CRECovery:ODRatio? on page 417
:N1000:CRECovery:ODRatio:AUTO[?]	:N1000:CRECovery:ODRatio:AUTO? on page 417
:N1000:CRECovery:RDIVider[?]	:N1000:CRECovery:RDIVider? on page 417
:N1000:CRECovery:RELock	:N1000:CRECovery:RELock on page 418
:N1000:CRECovery:SOURce[?]	:N1000:CRECovery:SOURce[?] on page 418
:N1000:CRECovery:SSCLock[?]	:N1000:CRECovery:SSCLock [?] on page 419

:N1000:EMODules:SLOT:CONNect[?]

Syntax :N1000:EMODules:SLOT:CONNect <identifier>,{ON | OFF | 1 | 0}

:N1000:EMODules:SLOT:CONNect? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

ON | OFF | 1 | 0

Return Range ON | OFF | 1 | 0

Description This command enables usage of the addressed slot by M8070B. In case of using multiple M8070B instances, make sure to enable a slot only in exactly one M8070B instance.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :N1000:EMOD:SLOT:CONN 'DCA1.SLOT1',ON

NOTE

This command is not the same as the similar FlexDCA command :EMODules:SLOT{n}:CONNect[?] {ON | OFF | 1 | 0}

The above command must be sent to FlexDCA prior starting up the M8070B software instances that shall control FlexDCA!

Managing the slot usage and connection state must be done using the FlexDCA commands or user interface before starting up M8070B.

:N1000:EMODules:SLOT:MODel?

Syntax :N1000:EMODules:SLOT:MODel? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This query returns the model number of all DCA related modules which are connected to M8070B system software.

This SCPI is applicable for M8046A.

Example :N1000:EMOD:SLOT:MOD? 'DCA1.SLOT1'

:N1000:CRECovery:CLBandwidth[?]

Syntax :N1000:CRECovery:CLBandwidth <identifier>,<bandwidth>

:N1000:CRECovery:CLBandwidth? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This command sets the fix loop bandwidth.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :N1000:CREC:CLB 'DCA1.SLOT1',10e6

For detailed documentation, see :CRECovery{n}:CLBandwidth in the FlexDCA online help.

:N1000:CRECovery:CRATe?

Syntax :N1000:CRECovery:CRATe? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This query returns the clock recovery’s symbol rate.

This SCPI is applicable for M8046A.

Example :N1000:CREC:CRAT? 'DCA1.SLOT1'

For detailed documentation, see :CRECovery{n}:CRATe in the FlexDCA online help.

:N1000:CRECovery:ELEVel[?]

Syntax :N1000:CRECovery:ELEVel <identifier>,<NRf>

:N1000:CRECovery:ELEVel? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This command the equalizer at the electrical inputs of the clock recovery. The equalizer level range is 0 to 100.

This query returns the current settings.

This SCPI is applicable for N1076B and N1078A.

Example :N1000:CREC:ELEVel 'DCA1.SLOT1',12

For detailed documentation, see :CRECovery{n}:ELEVel in the FlexDCA online help.

:N1000:CRECovery:LBWMode[?]

Syntax :N1000:CRECovery:LBWMode <identifier>,{FIXed | RDEPendent}

:N1000:CRECovery:LBWMode? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This command sets the loop bandwidth mode to either fixed or symbol rate dependent.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :N1000:CREC:LBWM 'DCA1.SLOT1',RDEP

For detailed documentation, see :CRECovery{n}:LBWMode in the FlexDCA online help.

:N1000:CRECovery:LOCKed?

Syntax :N1000:CRECovery:LOCKed? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This query displays the clock recovery's lock status.

Response: 1 (for locked) or 0 (for unlocked)

This SCPI is applicable for M8046A.

Example :N1000:CREC:LOCK? 'DCA1.SLOT1'

For detailed documentation, see :CRECovery{n}:LOCKed the FlexDCA online help.

:N1000:CRECovery:LSELect[?]

Syntax :N1000:CRECovery:LSELect <identifier>, <LOOP1 | LOOP2 | LOOP3 | LOOP4>

:N1000:CRECovery:LSELect? <identifier>,

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This command selects the type 2 loop transition frequency.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :N1000:CREC:LSEL 'DCA1.SLOT1', LOOP1

For detailed documentation, see :CRECovery{n}:LSELect in the FlexDCA online help.

:N1000:CRECovery:LSELect:AUTomatic[?]

Syntax :N1000:CRECovery:LSELect:AUTomatic <identifier>, <ON | OFF | 1 | 0>

:N1000:CRECovery:LSELect:AUTomatic? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

ON | OFF | 1 | 0

Description This command enables or disables automatic selection of the type 2 loop transition frequency.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :N1000:CREC:LSEL:AUTO 'DCA1.SLOT1',ON

For detailed documentation, see :CRECovery{n}:LSELect:AUTO in the

FlexDCA online help.

:N1000:CRECovery:ODRatio?

Syntax	:N1000:CRECovery:ODRatio? <identifier>
Input Parameters	'identifier': 'DCA*.SLOT*'
Return Range	SUB32 | SUB16 | SUB8 | SUB4 | SUB2 | UNITy | SUPer2 | SUPer4 | SUPer8
Description	This query returns the recovered clock's output divide ratio. This SCPI is applicable for M8046A.
Example	:N1000:CREC:ODR? 'DCA1.SLOT1' For detailed documentation, see :CRECovery{n}:ODRatio in the FlexDCA online help.

:N1000:CRECovery:ODRatio:AUTO?

Syntax	:N1000:CRECovery:ODRatio:AUTO? <identifier>
Input Parameters	Input Parameters 'identifier': 'DCA*.SLOT*'
Return Range	ON | OFF | 1 | 0
Description	This query returns the present state. This SCPI is applicable for M8046A.
Example	:N1000:CREC:ODR:AUTO? 'DCA1.SLOT1' For detailed documentation, see :CRECovery{n}:ODR:AUTO in the FlexDCA online help.

:N1000:CRECovery:RDIVider?

Syntax	:N1000:CRECovery:RDIVider? <identifier>,
Input Parameters	'identifier': 'DCA*.SLOT*'
Description	This query returns the loop bandwidth symbol rate divider. This SCPI is applicable for M8046A.
Example	:N1000:CREC:RDIV? 'DCA1.SLOT1'

For detailed documentation, see :CRECovery{n}:RDIVider in the FlexDCA online help.

:N1000:CRECovery:RELock

Syntax :N1000:CRECovery:RELock <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This command initiates a re-lock of the clock recovery.

NOTE **The re-lock operation can take quite some time. Especially when the clock recovery cannot lock to the received signal. Therefore, the VISA timeout shall be increased to well above 10 seconds to ensure that the controlling program is waiting long enough for the re-lock operation to finish.**

This SCPI is applicable for M8046A.

Example :N1000:CREC:REL 'DCA1.SLOT1';*OPC?

For detailed documentation, see :CRECovery{n}:RELock in the FlexDCA online help.

:N1000:CRECovery:SOURce[?]

Syntax :N1000:CRECovery:SOURce<identifier>,{ELECtrical | EINVerted | DIFFerential}

:N1000:CRECovery:SOURce?<identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

ELECtrical | EINVerted | DIFFerential

Description This command selects the input of the clock recovery.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :N1000:CREC:SOUR 'DCA1.SLOT1',DIFF

For detailed documentation, see :CRECovery{n}:SOURce in the FlexDCA online help.

:N1000:CRECovery:SSCLock [?]

Syntax :N1000:CRECovery:SSCLock [?] <identifier>,{OFF | ON | 0 | 1}

:N1000:CRECovery:SSCLock? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

ELECtrical | EINVerted | DIFFerential | AUXilliary

Description This command turns on the ability to use the nominal symbol rate when recovering the clock on Spread Spectrum Clock (SSC) signals. Normally, FlexDCA uses the average symbol rate to recovery the clock. However, when performing clock recovery on a SSC signal, using the nominal symbol rate may be required. The default setting is off.

This command requires FlexDCA revision A.05.80 and above, and is only available for clock recovery modules with Option JSA (see the Electrical and Optical Clock Data Recovery Solutions data sheet, 5992-1620EN)

This query returns the present state.

This SCPI is applicable for M8046A.

Example :N1000:CREC:SSCL 'DCA1.SLOT1',ON

For detailed documentation, see :CRECovery{n}:SSCLock in the FlexDCA online help.

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?
│  └─:DETailed?
└─:IBERate?
```

This subsystem has the following commands:

Table 78

Name	Description under
:BCOunter?	:FETCh:BCOunter? on page 421
:BCOunter:DETailed?	:FETCh:BCOunter:DETailed? on page 424
:IBERate?	:FETCh:IBERate? on page 425

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 79

License	Description	Modules
M8041A-0A2	Second Channel for Analyzer	M8041A
M8041A-0G2	Second Channel for Generator	M8041A
M8041A-C08	BERT one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8043A-A32	Error analysis up to 32.4 GBd for NRZ and PAM4	M8043A
M8046A-A32	Analyzer, one channel, data rate up to 32 GBd, NRZ	M8046A
M8062A-C32	32 Gb/s BERT front-end	M8062A
M8062A-G32	32 Gb/s Pattern generator front-end	M8062A

:FETCh:BCOunter?

Syntax :FETCh:BCOunter? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2', or 'M*.DataIn', 'INF*.DataIn1'

Description This query returns the accumulated bit counts and coding specific counts since start of accumulation period. 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 be a comma and enclosed by parenthesis.

The returned expression contains a timestamp, the location and a set of bit counter values.

(TimeStamp, “Location”, Counted1s, Counted0s, Erroneous1s, Erroneous0s[, coding specific counters])

Timestamp: As timestamp the UNIX epoch time is used; it is defined as elapsed seconds, milliseconds etc. since 01.01.1970 UTC.

Location: It’s a string referred to the place the bit counter are located. This parameter is important if a group name is used as 'Identifier'.

Counted1s: The number of received ones since start of accumulation period

Counted0s: The number of received zeros since start of accumulation period.

Erroneous1s: The number of received erroneous ones since start of accumulation period.

Erroneous0s: The number of received erroneous zeros since start of accumulation period.

Additional Counters for 8b10b coding:

- ComparedSymbols
- ErroredSymbols
- IllegalSymbols
- WrongDisparity
- FillerSymbols
- ReceivedSymbols
- Frames
- ErroredFrames

Additional Counters for 128/130 and 128/132 coding:

- Blocks
- ErroredBlocks
- IllegalSyncHeaders
- FillerSymbols
- ModifiedFillerSymbols
- Frames
- ErroredFrames
- Additional Counter for 128/132 coding
- CorrectedSyncHeaders

Additional Counters for PAM4 line coding

- Compared Symbols
- Errored Symbols
- Compared Symbol0
- Errored Symbol0
- Compared Symbol1
- Errored Symbol1
- Compared Symbol2

- Errored Symbol2
- Compared Symbol3
- Errored Symbol3

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.

This query returns the present state.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A, M8046A and real-time oscilloscope based error detector channels.

Example :FETCh:BCO? 'M1.DataIn1'

Response: (1.366014336e9,'M1.DataIn1',1.2222e10,1.112345e10,5.1e1,7.2e2)

:FETCh:BCO? 'Lanes'

Response: (1.366014336e9, 'M1.DataIn1',1.2222e10,1.112345e10,5.1e1,7.2e2, 'M1.DataIn2', ...,...)

NOTE **Unix time or POSIX time, is a system for describing instances in time, defined as the number of seconds that have elapsed since midnight Coordinated Universal Time (UTC), 1. January 1970, not including leap seconds.**

:FETCh:BCOunter:DETailed?

Syntax :FETCh:BCOunter:DETailed? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1' or 'M*.DataIn2', 'INF*.DataIn1'

Description This query is reporting all available error counts in full detail.

Currently this command is giving the same response as :FETC:BCO? for all counter types except when the line coding of the error detector is configured to PAM4.

In PAM4 mode, the PAM4 counters are more detailed, and report the number of times a certain symbol level has been received for an expected symbol level. This allows detailed analysis of the analog effect that cause symbol errors.

The response format is as documented for :FETCh:BCOunter? above, with the following change for the PAM4 counter set.

Additional Counters for PAM-4 line-coding

- Compared Symbols
- Errored Symbols
- Number of Symbol Levels (4 for PAM4)
- (Symbol0 counters)
- (Symbol1 counters)
- (Symbol2 counters)
- (Symbol3 counters)

The symbol counters are reporting the following counts

- Symbol Level (0 .. 3 for PAM4)
- Symbol Error Ratio for this symbol level
- Number of compared symbols for this symbol level
- Number of symbol errors for this symbol level
- (Received Symbol Counters)

The Received Symbol Counters report how often a specific symbol level has been received for the expected symbol level that it is reported for. Every received symbol that is on another symbol level than the expected symbol level is an erroneous received symbol.

- Number of received Symbol0

- Number of received Symbol1
- Number of received Symbol2
- Number of received Symbol3

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A, M8046A and real-time oscilloscope based error detector channels.

:FETCh:IBERate?

Syntax :FETCh:IBERate? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1' or 'M*.DataIn2', 'INF*.DataIn1'

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.

(TimeStamp, “Location”, BER[, coding specific ratios])

Timestamp: As timestamp the Unix epoch time is used; it is defined as elapsed seconds, milliseconds etc., since 01.01.1970 UTC.

Location: It is a string referred to the place the ratios are 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 Ratios for 128/130 and 128/132 coding:

- BlockErrorRatio
- FillerSymbolRatio
- FrameErrorRatio

Additional Counters for PAM4 line coding

- Symbol Error Ratio
- Symbol0 Error Ratio
- Symbol1 Error Ratio

- Symbol2 Error Ratio
- Symbol3 Error Ratio

This query returns the present state.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A, M8046A and real-time oscilloscope based error detector channels.

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.

Example :FETC:IBER? 'M6.DataIn1'
Response: (1.266014336e9, 'M6.DataIn1', 1.2345e-1)

:FETC:IBER? 'Lanes'
Response: (1.266514336e9, 'M1.DataIn1', 1.2345e-1, 'M1.DataIn2', 0.0e0)

:FETCh:HISTogram:BIN[:DATA]?

Syntax :FETCh:HISTogram:BIN[:DATA]? 'Identifier'

Input Parameters 'Identifier': 'M*.DataIn'

Description This query returns the histogram measured in the data input signal path in ASCII format as comma separated list. In the M8043A the histogram is measured after Clock Data Recovery and FFE. Use the :INPut[:VOLTage]:WINDow:HISTogram? query to retrieve the voltage window covered in the histogram.

This SCPI is applicable for M8043A.

Example :FETC:HIST:BIN? 'M1.DataIn'

:FETCh:EDIagram Subnode

This subnode provides commands and queries to retrieve the data of the histogram-based eye diagram of the Sampling Point Setup.

NOTE The M8070ADVB (Advanced Measurement Package) license must be installed to run the commands and queries for this subnode.

This subnode has the following commands:

Table 80

Name	Description under
:FETCh:EDIagram:SINGle	:FETCh:EDIagram:SINGle on page 427
:FETCh:EDIagram:COLumns?	:FETCh:EDIagram:COLumns? on page 428
:FETCh:EDIagram[:DATA]?	:FETCh:EDIagram[:DATA]? on page 428
:FETCh:EDIagram:ROWS?	:FETCh:EDIagram:ROWS? on page 429
:FETCh:EDIagram:XINCrement?	:FETCh:EDIagram:XINCrement? on page 430
:FETCh:EDIagram:XORigin?	:FETCh:EDIagram:XORigin? on page 430
:FETCh:EDIagram:YINCrement?	:FETCh:EDIagram:YINCrement? on page 430
:FETCh:EDIagram:YORigin?	:FETCh:EDIagram:YORigin? on page 431

:FETCh:EDIagram:SINGle

Syntax :FETCh:EDIagram:SINGle 'Identifier'

Input Parameters 'Identifier': 'M*.DataIn'

Description This command creates a new eye diagram for which the data can be retrieved using the :FETCh:EDIagram[:DATA]? query.

This SCPI is applicable for M8043A.

Example :FETC:EDI:SING 'M3.DataIn'

:FETCh:EDIagram:COLumns?

Syntax :FETCh:EDIagram:COLumns? 'Identifier'

Input Parameters 'Identifier': 'M*.DataIn'

Description This query returns the number of columns in the color grade/gray scale data array. Use the :FETCh:EDIagram[:DATA]? query to actually transfer the array.

This SCPI is applicable for M8043A.

Example :FETC:EDI:COL? 'M3.DataIn'

:FETCh:EDIagram[:DATA]?

Syntax :FETCh:EDIagram[:DATA]? 'Identifier'

Input Parameters 'Identifier': 'M*.DataIn'

Description This query returns color grade/gray scale database values in ASCII format as comma-separated values. Color grade/gray scale data is transferred as a two-dimensional array. The number of rows and columns can be retrieved by using the :FETCh:EDIagram:ROWS? and :FETCh:EDIagram:COLumns? queries. This array corresponds to the graticule display, where each point contains a sample hit count. The array is transferred column by column, starting with the lower left corner of the graticule.

The time for any column of data points can be calculated as:

$$X_{time} = (X_{increment})(X_{index}) + X_{origin}$$

where,

$X_{increment}$ is the time spacing between database columns and is returned with the :FETCh:EDIagram:XINCrement? query.

X_{index} is the column count index, starting at zero and ending at the value returned by the :FETCh:EDIagram:COLumns? query.

X_{origin} is the time position of the first column and is returned using the :FETCh:EDIagram:XORigin? query.

The voltage for any data point at a given row (that contains a hit count) is determined by using the following equation:

$$Y_{amplitude} = (Y_{increment})(Y_{index}) + Y_{origin}$$

where,

$Y_{increment}$ is the amplitude spacing between database rows and is returned with the :FETCh:EDIagram:YINCrement?query.

Y_{index} is the row count index, starting at zero and ending at the value returned by :FETCh:EDIagram:ROWS? query.

Y_{origin} is a voltage value of the top row and is returned using the :FETCh:EDIagram:YORigin? query.

This SCPI is applicable for M8043A.

Example :FETC:EDI? 'M3.DataIn'

:FETCh:EDIagram:ROWS?

Syntax :FETCh:EDIagram:ROWS? 'Identifier'

Input Parameters 'Identifier': 'M*.DataIn'

Description This query returns the number of rows in the color grade/gray scale data array. Use the :FETCh:EDIagram[:DATA]? query to actually transfer the array.

This SCPI is applicable for M8043A.

Example :FETC:EDI:ROWS? 'M3.DataIn'

:FETCh:EDIagram:XINCrement?

Syntax :FETCh:EDIagram:XINCrement? 'Identifier'

Input Parameters 'Identifier': 'M*.DataIn'

Description This query returns the difference in time between two adjacent horizontal points (adjacent data columns) in the eye diagram. To query the difference in amplitude between adjacent vertical points (adjacent data rows) in the eye diagram, use the :FETCh:EDIagram:YINCrement? query.

This SCPI is applicable for M8043A.

Example :FETC:EDI:XINC? 'M3.DataIn'

:FETCh:EDIagram:XORigin?

Syntax :FETCh:EDIagram:XORigin? 'Identifier'

Input Parameters 'Identifier': 'M*.DataIn'

Description This query returns the time value at the x-axis origin. The origin for both the y and x axis is located at the lower-left corner of the display graticule. To query the amplitude value at the y-axis origin, use the :FETCh:EDIagram:YORigin? query.

This SCPI is applicable for M8043A.

Example :FETC:EDI:XOR? 'M3.DataIn'

:FETCh:EDIagram:YINCrement?

Syntax :FETCh:EDIagram:YINCrement? 'Identifier'

Input Parameters 'Identifier': 'M*.DataIn'

Description This query returns the difference in amplitude between two adjacent vertical points (adjacent data rows) in the eye diagram. To query the difference in time between adjacent points (adjacent data columns) in the eye diagram, use the :FETCh:EDIagram:XINCrement? query.

This SCPI is applicable for M8043A.

Example :FETC:EDI:YINC? 'M3.DataIn'

:FETCh:EDIagram:YORigin?

Syntax	:FETCh:EDIagram:YORigin? 'Identifier'
Input Parameters	'Identifier': 'M*.DataIn'
Description	This query returns the amplitude value at the y-axis origin. The origin for both the y and x axis is located at the lower-left corner of the display graticule. To query the time value at the x-axis origin, use the :FETCh:EDIagram:XORigin? query. This SCPI is applicable for M8043A.
Example	:FETC:EDI:YOR? 'M3.DataIn'

MMEMory Subsystem

All instrument states of the M8020A 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]
  │   ├ [:FILE]
  │   ├ :DATA
  │   ├ :CATalog?
  │   └ :DELete
  └ :EXPort
    └ [:USER]
      ├ [:FILE]
      ├ :DATA
      ├ :CATalog?
      └ :FACTory:CATalog?
```

This subsystem has the following commands:

Table 81

Name	Description under
:WORKspace:SETTings:USER:CATalog?	:MMEMory:WORKspace:SETTings[:USER]:CATalog? on page 434
:WORKspace:SETTings:USER:STORe	:MMEMory:WORKspace:SETTings[:USER]:STORe on page 434
:WORKspace:SETTings:USER:RECall	:MMEMory:WORKspace:SETTings[:USER]:RECall on page 434
:WORKspace:SETTings:USER:DELete	:MMEMory:WORKspace:SETTings[:USER]:DELete on page 435
:WORKspace:SETTings:FACTory:CATalog?	:MMEMory:WORKspace:SETTings:FACTory:CATalog? on page 435
:WORKspace:SETTings:FACTory:RECall	:MMEMory:WORKspace:SETTings:FACTory:RECall on page 435
:WORKspace:IMPort[:USER]	:MMEMory:WORKspace:IMPort[:USER] on page 436
WORKspace:IMPort[:USER][:FILE]	:MMEMory:WORKspace:IMPort[:USER][:FILE] on page 437
:WORKspace:IMPort[:USER]:DATA	:MMEMory:WORKspace:IMPort[:USER]:DATA on page 438
:WORKspace:IMPort[:USER]:CATalog?	:MMEMory:WORKspace:IMPort[:USER]:CATalog? on page 439
:WORKspace:IMPort[:USER]:DELete	:MMEMory:WORKspace:IMPort[:USER]:DELete on page 439
:WORKspace:EXPort[:USER][:FILE]	:MMEMory:WORKspace:EXPort[:USER][:FILE] on page 440
:WORKspace:EXPort[:USER]:DATA?	:MMEMory:WORKspace:EXPort[:USER]:DATA? on page 441
:WORKspace:EXPort[:USER]:CATalog?	:MMEMory:WORKspace:EXPort[:USER]:CATalog? on page 442
:WORKspace:EXPort:FACTory:CATalog?	:MMEMory:WORKspace:EXPort:FACTory:CATalog? on page 442

: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][:FILE]

Syntax :MMEMory:WORKspace:IMPort[:USER][:FILE] <ProfileType>,<"SourcePath">,<"RelativeTargetPath">[,<"ArbitaryProfileType">]

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 JTTemplate | PCIDeemphasis | PCI6Deemphasis | DEPResets | MODClockDeviations | PSMapping | LTRSettings

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:IMP pcid, "c:/temp/my_presets.xml","shared:pci_calibrated_presets.xml"

:MMEMory:WORKspace:IMPort[:USER]:DATA

Syntax :MMEMory:WORKspace:IMPort[:USER]:DATA <ProfileType>,<"RelativeTargetPath">,<definite length block>[,<"ArbitaryProfileType">]

Input Parameters <ProfileType>: JSPRofile | JTTemplate | PCIDeemphasis | SSCProfile | SPProfile | AFSChema.

<"RelativeTargetPath">: "current:..." addresses the current setting and "shared: ..." addresses a location in the storage that is common for all user settings.

<definite length block>: Definite length binary data, Contains the data of the file. The format is #lLLLddddd

a l

Number of digits of the length field LLL

b LLL

Number of bytes of data

c ddddd

The data content for the file

Name of the storage schema. This has to be provided for the profile type AFSChema and can be omitted otherwise.

File extensions to be used are:

Jitter sweep profiles: .jcs

SSC Profiles:.txt

Jitter Tolerance Template: .jtt

PCIDeemphasis preset definition: .xml.

S-parameter profiles containing1- to 9-port components are supported. The file extension "...s1p...s9p" can be used.

AFSChema is an arbitrary profile type, which is then specified in the ArbitaryProfileType argument. This is specific to plugins that have own profile types and is documented there.

Description This command imports a file into the workspace as binary data block.

Example :MMEM:WORK:IMP:DATA jspr, "shared:Jittersweep.jcs",#46730.......

Example :MMEM:WORK:IMP:DATA jspr, "current:Jittersweep.jcs",#46730.......

Example :MMEM:WORK:IMP:DATA jspr, "current:test/PCIe3.0/Jittersweep.jcs",#46730.......

Example :MMEM:WORK:IMP:DATA ssp, “shared:INCH7_7_88.s4p,#46730.......”

Example :MMEM:WORK:IMP:DATA pcid, "shared:pci_calibrated_presets.xml,#46730......."

: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"

:MMEMory:WORKspace:EXPort[:USER][:FILE]

Syntax

:MMEMory:WORKspace:EXPort[:USER][:FILE] <ProfileType>,<"SourcePath">,<"RelativeTargetPath">[,<"ArbitaryProfileType">]

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 exported for generating jitter sweep or spread spectrum clocking. Three arguments have to be specified:

The first one is the profile type JSPRofile | JTTemplate | PCIDeemphasis | SSCProfile | SPProfile | AFSChema | DEemphasis PRofile,

second one is the absolute source path of the file to be exported,

third one is a relative target path.

For jitter sweep files the file extension "...jcs" has to be used and for spread spectrum clocking the extension "...txt". For Jitter Tolerance Template files the extension ...jtt.

PCIDeemphasis & DEemphasis PRofile preset definition files must use the file extension xml.

S-parameter profiles containing1- to 9-port components are supported. The file extension "...s1p...s9p" can be used.

AFSChema is an arbitrary profile type, which is then specified in the ArbitaryProfileType argument. This is specific to plugins that have own profile types and is documented there.

Example

```
:MMEM:WORK:EXP
depr, "shared:DeemphasisPresetProfile.xml","c:/temp/
DeemphasisPresetProfile.xml"
```

:MMEMory:WORKspace:EXPort[:USER]:DATA?

Syntax :MMEMory:WORKspace:EXPort[:USER]:DATA? <ProfileType>,<"RelativeTargetPath">

Input Parameters <ProfileType>: JSPRofile | JTTemplate | PCIDeemphasis | SSCProfile | SPProfile | AFSChema | DEemphasis PRofile,

<RelativeTargetPath>: "current:...“addresses the current setting and "shared: ..." addresses a location in the storage that is common for all user settings & "factory:..." addressed location of factory files.

Name of the storage schema. This has to be provided for the profile type AFSChema and can be omitted otherwise.

File extensions to be used are:

Jitter sweep profiles: .jcs

SSC Profiles:.txt

Jitter Tolerance Template: .jtt

PCIDeemphasis & DEemphasis PRofile preset definition: .xml.

S-parameter profiles containing1- to 9-port components are supported. The file extension "...s1p...s9p" can be used.

AFSChema is an arbitrary profile type, which is then specified in the ArbitaryProfileType argument. This is specific to plugins that have own profile types and is documented there.

Description This command returns the data of the specified workspace file as a binary block.

Example :MMEM:WORK:EXP:DATA?
depr, "shared:DeemphasisPresetProfile.xml"

:MMEMory:WORKspace:EXPort[:USER]:CATalog?

Syntax	:MMEMory:WORKspace:EXPort[:USER]:CATalog? <profileType>
Input Parameters	<ProfileType>: The profile type specifier can be any of the ones listed in :MMEMory:WORKspace:IMPort[:USER]:CATalog? on page 439.
Description	This command returns a list of files that can be exported.
Example	:MMEM:WORK:EXP:CAT? jspr
Returns	"current: Jittersweep.jcs","shared: Jittersweep.jcs","shared:Sweep.jcs","shared:Test42.jcs"

:MMEMory:WORKspace:EXPort:FACTory:CATalog?

Syntax	:MMEMory:WORKspace:EXPort:FACTory:CATalog? <profileType>
Input Parameters	<ProfileType>: The profile type specifier can be any of the ones listed in :MMEMory:WORKspace:IMPort[:USER]:CATalog? on page 439.
Description	This command returns a list of factory files that can be exported.
Example	:MMEM:WORK:EXP:FACT:CAT? jspr
Returns	"current: Jittersweep.jcs","shared: Jittersweep.jcs","shared:Sweep.jcs","shared:Test42.jcs"

ACQuisition subsystem

The acquisition subsystem controls how a real-time oscilloscope is being configured in order to capture the signal for BER measurements. This is comparable to setting up the acquisition memory of the oscilloscope, but in terms of bits or symbols, and not in analog samples.

The channel bandwidth limit must be equal or less to the global bandwidth limit setting. The available bandwidth limit depends on the oscilloscope model being used.

Additionally, the acquisition can be stopped in M8070B, so that the oscilloscope becomes free for manual adjustments, while still being under parametric control of M8070B.

This subsystem has the following SCPI structure:

```
:ACQuisition
├─:BANDwidth
│ ├─:MODE[?]
│ ├─:STATe[?]
│ ├─:TYPE[?]
│ └─:VALue[?]
├─:BITS[?]
├─:DISPlay[?]
├─:POINts[:ANALog]?
│ └─:REConfigure
├─:STATe[?]
└─:SYMBols[?]
```

This subsystem has the following commands:

Table 82

Name	Description under
:ACQuisition:BANDwidth:MODE[?]	:ACQuisition:BANDwidth:MODE[?] on page 444
:ACQuisition:BANDwidth:STATe[?]	:ACQuisition:BANDwidth:STATe[?] on page 445
:ACQuisition:BANDwidth:TYPE[?]	:ACQuisition:BANDwidth:TYPE[?] on page 445
:ACQuisition:BANDwidth:VALue[?]	:ACQuisition:BANDwidth:VALue[?] on page 446
:ACQuisition:BITS[?]	:ACQuisition:BITS[?] on page 446
:ACQuisition:DISPlay[?]	:ACQuisition:DISPlay[?] on page 447
:ACQuisition:POINts[:ANALog]?	:ACQuisition:POINts[:ANALog]? on page 447
:ACQuisition:POINts:REConfigure	:ACQuisition:POINts:REConfigure on page 447
:ACQuisition:STATe[?]	:ACQuisition:STATe[?] on page 448
:ACQuisition:SYMBols[?]	:ACQuisition:SYMBols[?] on page 448

:ACQuisition:BANDwidth:MODE[?]

Syntax	:ACQuisition:BANDwidth:MODE <identifier>,{AUTO | MANual | MAX}
	:ACQuisition:BANDwidth:MODE? <identifier>
Input Parameters	<identifier>: 'INF*.Common'
	AUTO | MANual | MAX
Description	This command controls the global bandwidth limit setting of the real-time oscilloscope. The available bandwidth limit depends on the oscilloscope model being used.
	This query returns the present setting.
Example	:ACQ:BAND:MODE 'INF1.Common',MAN

:ACQuisition:BANDwidth:STATe[?]

Syntax :ACQuisition:BANDwidth:STATe <identifier>,{OFF | ON | 0 | 1}

:ACQuisition:BANDwidth:STATe? <identifier>

Input Parameters <identifier>: 'INF*.DataIn1'

OFF | ON | 0 | 1

Description This command enables or disabled the channel bandwidth limit of the real-time oscilloscope. The channel bandwidth limit must be equal or less to the global bandwidth limit setting.

The available bandwidth limit depends on the oscilloscope model being used.

This query returns the present setting.

Example :ACQ:BAND:STAT 'INF1.DataIn1',ON

:ACQuisition:BANDwidth:TYPE[?]

Syntax :ACQuisition:BANDwidth:TYPE <identifier>,{BESSEL4 | WALL | BUTTerworth | BANDpass}

:ACQuisition:BANDwidth:TYPE? <identifier>

Input Parameters <identifier>: 'INF*.DataIn1'

BESSEL4 | WALL | BUTTerworth | BANDpass

Description This command controls the channel bandwidth limit filter of the real-time oscilloscope. The available bandwidth limit depends on the oscilloscope model being used.

The following options are available:

- WALL — Specifies a Brick-Wall response for the bandwidth limit filter. This response has a sharp roll-off.
- BESSEL4 — Specifies a 4th order Bessel response for the bandwidth limit filter. This response has a more gradual roll-off.
- BUTTerworth — Specifies a 4th order Butterworth response for the bandwidth limit filter. This response has a roll-off that is between the Brick Wall and Bessel responses.
- BANDpass — This option is used with the Phase Noise analysis application. As you do not have control over the slope of the transition from the pass band to the rejected bands, this option has limited usefulness as a general-purpose bandpass filter.

This query returns the present setting.

Example :ACQ:BAND:TYPE 'INF1.DataIn1',BESSEL4

:ACQuisition:BANDwidth:VALue[?]

Syntax :ACQuisition:BANDwidth:VALue <identifier>,<bandwidth>

:ACQuisition:BANDwidth:VALue? <identifier>

Input Parameters <identifier>: 'INF*.Common' or 'INF*.DataIn1'

Description This command controls the bandwidth limit of the real-time oscilloscope.

The available bandwidth depends on the oscilloscope model being used.

This query returns the present setting.

Example :ACQ:BAND:VAL 'INF*.Common',40e9

:ACQuisition:BITS[?]

Syntax :ACQuisition:BITS<identifier>,<number_of_bits>

:ACQuisition:BITS? <identifier>

Input Parameters <identifier>: 'INF*.DataIn1'

Description This command configures the number of bits to capture and process per acquisition. This does directly control the smallest detectable BER value per acquisition, which corresponds to the instantaneous BER result (see.:FETCh:BCOunter:DETailed? on page 424).

Increasing the number of bits per acquisition will lower the detectable BER, but at the same time increase the processing duration per acquisition.

This query returns the present setting.

Example :ACQ:BITS 'INF1.DataIn1',1e6

:ACQuisition:DISPlay[?]

Syntax :ACQuisition:DISPlay <identifier>,<1|0|ON|OFF>

:ACQuisition:DISPlay? <identifier>

Input Parameters 1|0|ON|OFF

<identifier>: 'INF*.Common'

Description This command enables/disables the RED mode in the scope. When the RED mode is enabled, the hardware acceleration is fast (up to five times) in the scope. To enable the RED mode, the Acquisition and Display parameters should be enabled (ON).

This query returns the present setting.

Example :ACQ:DISP 'INF1.Common',ON

:ACQuisition:POINts[:ANALog]?

Syntax :ACQuisition:POINts[:ANALog]? <identifier>

Input Parameters <identifier>: 'INF*.DataIn1', 'INF*.Common'

Description This query returns the number of analog sample points that are configured into the oscilloscope. This value is for information only, and derives from the configured number of bits per acquisitions and the channel's symbol rate.

Example :ACQ:POIN? 'INF1.DataIn1'

Returns 2.96548e6

:ACQuisition:POINts:REConfigure

Syntax :ACQuisition:POINts:REConfigure <identifier>

Input Parameters <identifier>: 'INF*.DataIn1', 'INF*.Common'

Description This command performs a reset and full reprogramming of the oscilloscope. It effectively re-applies the M8070B settings and resets the manual changes.

Example :ACQ:POIN:REC 'INF1.DataIn1'

:ACQuisition:STATe[?]

Syntax :ACQuisition:STATe <identifier>, {OFF | ON | 0 | 1}

:ACQuisition:STATe? <identifier>

Input Parameters <identifier>: 'INF*.Common'

Description This command enables or disables the acquisition. If <identifier> addresses a location that is not a channel, then the acquisition on all channels of the oscilloscope will be controlled by this command.

This query returns the present setting.

Example :ACQ:STAT 'INF1.Common',OFF

:ACQuisition:SYMBols[?]

Syntax :ACQuisition:SYMBols <identifier>,<number_of_UI>

:ACQuisition:SYMBols? <identifier>

Input Parameters <identifier>: 'INF*.DataIn1'

Description This command configures the number of symbols to capture and process per acquisition. This does directly control the smallest detectable BER value per acquisition, which corresponds to the instantaneous BER result (see :FETCh:BCOunter:DETailed? on page 424).

Increasing the number of symbols per acquisition will lower the detectable BER, but at the same time increase the processing duration per acquisition. In case of NRZ, this command will produce the same results as :ACQ:BITS (see :ACQuisition:BITS[?] on page 446), in case of PAM4 is will configure the number of bits to twice the number of symbols.

This query returns the present setting.

Example :ACQ:SYMB 'INF1.DataIn1',1e6

PLUGin Subsystem

The M8020A/M8040A/M8050A platform supports a plugin interface used to implement certain interfaces recognized by the software and integrated into the M8070B GUI and instrument software. Error Ratio, Output Timing, Jitter Tolerance, Output Level, Eye Diagram, Parameter Sweep, Pattern Capture and Error Distribution Analysis measurements are examples of the plugin concept.

For more information about the plugin interface, refer to PLUGin Subsystem on page 33.

Out of these measurements, Error Ratio measurement and Pattern Capture is available with M8070B software system. However, all other measurements are the part of plugins and will only be available on installing them in M8070B software system. The following plugins are available:

- Advanced Measurement Package - The Advanced Measurement Package provides the following measurements:
  - Output Timing
  - Output Level
  - Jitter Tolerance
  - Eye Diagram
  - Parameter Sweep
  - BER Based Deemphasis

  For details on the measurements provided by Advanced Measurement Package, please refer to *M8000 Series Advance Measurement Package User Guide*.
- Error Distribution Analysis Package - The Error Distribution Analysis Package provides the following measurement:
  - Error Distribution Analysis

  For details on the measurement provided by Error Distribution Analysis Package, please refer to *M8000 Series Error Distribution Analysis Package User Guide*.

NOTE **The Advanced Measurement Package and Error Distribution Analysis Package requires a valid license for activation.**

This subsystem has the following SCPI structure:

```
:PLUGin
 ├:CATalog?
 ├:RNODes
 │ └[:LIST]?
 │
 │ Default measurements provided by
 │ M8070B System Software
 │
 ├:ERATio
 │ └...
 ├:CCAPture
 │ └ ...
 ├:LFPSgen
 │ └ ...
 │
 │ Additional measurements provided by
 │ M8070B System Software on installing
 │ the Advance Measurement Package
 │
 ├:OTIMing
 │ └...
 ├:OLEVel
 │ └...
 ├:JTOLerance
 │ └...
 ├:EDIagram
 │ └...
 ├:PSWEep
 │ └...
 │
 │ Additional measurement provided by
 │ M8070B System Software on installing
 │ the Error Distribution Analysis Package
 │
 └:EDA
   └...
```

This subsystem has the following subnodes:

Table 83

Name	Description under
:CATalog?	:PLUGin:CATalog? on page 452
:RNODes[:LIST]?	:PLUGin:RNODes[:LIST]? on page 452
:ERATio Subnode	:PLUGin:ERATio Subnode on page 453
:CCAPture Subnode	:PLUGin:CCAPture Subnode on page 482
:LFPSgen Subnode	:PLUGin:LFPSgen Subnode on page 501
Advanced Measurement Package	Advanced Measurement Package on page 512
Error Distribution Analysis Package	Error Distribution Analysis Package on page 512

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 84

License	Description	Modules
M8009A-062	Clock Generator two channel 60 GHz	M8009A
M8009A-0G6	Reference Clock Multiplier	M8009A
M8041A-0A2	Second Channel for Analyzer	M8041A
M8041A-0G2	Second Channel for Generator	M8041A
M8041A-C08	BERT one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8042A-G32	32 GBd PG for NRZ and PAM-4	M8042A
M8043A-A32	Error analysis up to 32.4 GBd for NRZ and PAM4	M8043A
M8045A-G32	32 GBd Pattern Generator	M8045A
M8046A-A32	32 GBd Error Detector	M8046A
M8062A-C32	32 Gb/s BERT front-end	M8062A
M8062A-G32	32 Gb/s Pattern generator front-end	M8062A

: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:RNODes[:LIST]?

Syntax :PLUGin:RNODes[:LIST]?

Description This query returns a list of SCPI plugin root nodes that can be used to address different type of measurements (plugins).

Example :PLUG:RNOD:LIST?
"CCAPture","ERATio","JTOLerance","OLEVel","OTIMing","LTXGKR","EDIag ram","PSWEep","OCEI","CPHYplugin"

: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 M8070B 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?, :PLUGin:ERATio:FETCh:DATA:ALL? or :FETCh subsystem commands.

The M8040A and M8050A system also supports error ratio measurement using NRZ or PAM4.

This subnode has the following SCPI structure:

```
:PLUGin
└:ERATio
  ├:ACQuisition
  │ ├:ALOCation[?]
  │ ├:AEND[?]
  │ ├:COMPared[?]
  │ ├:DURation[?]
  │ ├:HISTory[?]
  │ ├:INTerval[?]
  │ ├:TCLevel[?]
  │ ├:TERatio[?]
  │ └:TIME[?]
  ├:EVALuation
  │ ├:CERatio[?]
  │ └:RVMode?
  ├:FETCh
  │ ├:DATA?
  │ │ └:TABLe?
  │ ├:DATA:ALL?
  │ │ └:TABLe?
  │ └[:DURation]?
  │   ├:TABLe?
  │   ├:DETailed?
  │   └:PFResult?
  │     └:DETailed?
  ├:CATalog?
  ├:DELete
  ├:NEW
  ├:RESet
  ├:STARt
  ├:STOP
  ├:SHOW:SGLegends[?]
  ├:SHOW:COMment[?]
  ├:GRAPh?
  └:RUN
    ├:HISTory
    │ ├[:STATe][?]
    │ └:CLEar
    ├:LOG?
    ├:MESSage?
    ├:PROGress?
    └[:STATus]?
```

This subnode has the following commands:

Table 85

Name	Description under
:ACQuisition:ALOCation[?]	:PLUGin:ERATio:ACQuisition:ALOCation[?] on page 456
:ACQuisition:AEND[?]	:PLUGin:ERATio:ACQuisition:AEND[?] on page 457
:ACQuisition:SCRiteria[?]	:PLUGin:ERATio:ACQuisition:SCRiteria[?] on page 457
:ACQuisition:COMPared[?]	:PLUGin:ERATio:ACQuisition:COMPared[?] on page 458
:ACQuisition:DURation[?]	:PLUGin:ERATio:ACQuisition:DURation[?] on page 458
:ACQuisition:HISTory[?]	:PLUGin:ERATio:ACQuisition:HISTory[?] on page 459
:ACQuisition:INTerval[?]	:PLUGin:ERATio:ACQuisition:INTerval[?] on page 459
:ACQuisition:TCLevel[?]	:PLUGin:ERATio:ACQuisition:TCLevel[?] on page 460
:ACQuisition:TERatio[?]	:PLUGin:ERATio:ACQuisition:TERatio[?] on page 460
:ACQuisition:TIME[?]	:PLUGin:ERATio:ACQuisition:TIME[?] on page 461
:EVALuation:CERatio[?]	:PLUGin:ERATio:SHOW:CERatio[?] on page 461
:EVALuation:RVMode?	:PLUGin:ERATio:SHOW:RVMode[?] on page 462
:FETCh:DATA?	:PLUGin:ERATio:FETCh:DATA? on page 462
:FETCh:DATA:TABLe?	:PLUGin:ERATio:FETCh:DATA:TABLe? on page 464
:FETCh:DATA:ALL?	:PLUGin:ERATio:FETCh:DATA:ALL? on page 465
:FETCh:DATA:ALL:TABLe?	:PLUGin:ERATio:FETCh:DATA:ALL:TABLe? on page 469
:FETCh[:DURation]?	:PLUGin:ERATio:FETCh[:DURation]? on page 469
:FETCh[:DURation]:TABLe?	:PLUGin:ERATio:FETCh[:DURation]:TABLe? on page 470
:FETCh[:DURation]:DETailed?	:PLUGin:ERATio:FETCh[:DURation]:DETailed? on page 471
:FETCh[:DURation]:PFResult?	:PLUGin:ERATio:FETCh[:DURation]:PFResult? on page 473
:FETCh[:DURation]:PFResult:DETailed?	:PLUGin:ERATio:FETCh[:DURation]:PFResult:DETailed? on page 473
:CATalog?	:PLUGin:ERATio:CATalog? on page 474
:TABS?	:PLUGin:ERATio:TABS? on page 475
:EXPort	:PLUGin:ERATio:EXPort on page 475
:DELete	:PLUGin:ERATio:DELete on page 476

Name	Description under
:NEW	:PLUGin:ERATio:NEW on page 475
:RESet	:PLUGin:ERATio:RESet on page 476
:STARt	:PLUGin:ERATio:STARt on page 477
:STOP	:PLUGin:ERATio:STOP on page 477
:SHOW:SGLegends	:PLUGin:ERATio:SHOW:SGLegends[?] on page 477
:SHOW:COMment[?]	:PLUGin:ERATio:SHOW:COMment[?] on page 478
:GRAPh?	:PLUGin:ERATio:GRAPh? on page 478
:RUN:HISTory[:STATe]	:PLUGin:ERATio:RUN:HISTory[:STATe][?] on page 479
:RUN:HISTory:CLEar	:PLUGin:ERATio:RUN:HISTory:CLEar on page 479
:RUN:LOG?	:PLUGin:ERATio:RUN:LOG? on page 480
:RUN:MESSage?	:PLUGin:ERATio:RUN:MESSage? on page 480
:RUN:PROGress?	:PLUGin:ERATio:RUN:PROGress? on page 481
:RUN[:STATus]?	:PLUGin:ERATio:RUN[:STATus]? on page 481

: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 applicable for M8041A, M8051A, M8062A and M8046A.

Example :PLUG:ERAT:ACQ:ALOC 'MyMeasurement','M1.DataIn1'

:PLUGin:ERATio:ACQuisition:AEND[?]

Syntax	:PLUGin:ERATio:ACQuisition:AEND 'identifier', <PFA|FDUR:CBIT>
	:PLUGin:ERATio:ACQuisition:AEND? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
	<PFA|FDUR|CBIT>: Set accumulation end mode.
Return Range	PFA|FDUR|CBIT
Description	This command sets the accumulation to end when a pass/fail (PFA) condition has been met or end after the full duration (FDUR) or no.of bits (CBIT) of the measurement for the addressed measurement name identifier. The query returns the current accumulation end value.
	This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.
Example	:PLUG:ERAT:ACQ:AEND 'MyMeasurement', PFA

:PLUGin:ERATio:ACQuisition:SCRiteria[?]

Syntax	:PLUGin:ERATio:ACQuisition:SCRiteria 'identifier', <PAN|CAN>
	:PLUGin:ERATio:ACQuisition:SCR? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
	<PAN|CAN>: Set accumulation stop criteria.
Return Range	PAN|CAN
Description	This command sets the accumulation stop criteria as either "Per Analyzer" or "Combined Analyzer". The combined result will appear in form of Graph, Data Tab and Calculated Results. • Per Analyzer (PAN) - Graph, Data Tab and combined Calculated Results will not be visible. • Combined Analyzer (CAN) - Graph, Data Tab and combined Calculated Results will be visible.

NOTE

This command is only valid when a group is selected as an analyzer (If the group is having only one location, this command is not applicable).

This command depends on "Accumulation End". It will not be not applicable if the "Accumulation End" is set as "Full Duration".

The query returns the current accumulation end value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example :PLUG:ERAT:ACQ:SCR 'MyMeasurement', CAN

:PLUG:ERAT:ACQ:SCR? 'MyMeasurement',

:PLUGin:ERATio:ACQuisition:COMPared[?]

Syntax :PLUGin:ERATio:ACQuisition:COMPared 'identifier', <NRf>

:PLUGin:ERATio:ACQuisition:COMPared? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<NRf>: IE+6 to IE+18

Return Range IE+6 to IE+18

Description This command sets the Number of Compared Bits for the addressed measurement name identifier. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example :PLUG:ERAT:ACQ:COMP 'MyMeasurement', IE+9

NOTE **The BER measurement processes at least as many bits as defined by the acquisition parameters, it stops when the end criteria is reached or exceeded. Due to update period and hardware granularities, an individual BER measurement will typically process more bits than configured.**

: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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

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:SHOW:CERatio[?]

Syntax	:PLUGin:ERATio:SHOW:CERatio 'identifier', <ERAT|EZR|EOR>
	:PLUGin:ERATio:SHOW: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 applicable for M8041A, M8051A, M8062A and M8046A.
Example	:PLUG:ERAT:SHOW:CER 'MyMeasurement',EOR

:PLUGin:ERATio:SHOW:RVMode[?]

Syntax	:PLUGin:ERATio:SHOW:RVMode 'identifier', <DET|SUMM>
	:PLUGin:ERATio:SHOW: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 applicable for M8041A, M8051A, M8062A and M8046A.
Example	:PLUG:ERAT:SHOW:RVM 'MyMeasurement', DET

:PLUGin:ERATio:FETCh:DATA?

Syntax	:PLUGin:ERATio:FETCh:DATA? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name.
	[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Return Range	'M*.DataIn1' or 'M*.DataIn2' (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

Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, ...

Counter for 8b/10b Bit Coding

Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, Compared Symbols, Errored Symbols, Illegal Symbols, Wrong Disparity, Filler Symbols, Received Symbols, Frames, Errored Frames, ...

Counter for 128/130 Bit Coding

Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, Blocks, Errored Blocks, Illegal Syn Headers, Filler Symbols, Modified Filler Symbols, Frames, Errored Frames, ...

Counter for 128/132 Bit Coding

Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, Blocks, Errored Blocks, Illegal Syn Headers, Corrected Sync Headers, Filler Symbols, Modified Filler Symbols, Frames, Errored Frames, ...

Counter for PCIe6 1b1b Coding

Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, Compared Symbols, Errored Symbols, Filler Bits, Modified Filler Count, Filler Count, Flit Count, FEC Corrected Bytes, FLIT CRC Error Count, PAM Result)

This SCPI is applicable for M8041A, M8051A, M8043A, M8045A, M8046A 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, ...)

The following example shows the results returned when line coding is selected as PAM4:

("M2.DataIn",1511414904.34319,6064099210500,6064099210500,0,0,6064099210496,0,1516024802624,0,1516024802624,0,1516024802624,0,1516024802624,0,...)

:PLUGin:ERATio:FETCh:DATA:TABLe?

Syntax :PLUGin:ERATio:FETCh:DATA:TABLe? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional)

Return Range 'M*.DataIn1' or 'M*.DataIn2' (data results in tabular format)

Description This command returns the raw data of the error ratio measurement for the addressed measurement name identifier and location identifier in tabular format.

It works in the same way as the command :PLUGin:ERATio:FETCh:DATA?, with the only difference being that it returns the measurement data in a tabular format.

For more details, refer to the description of :PLUGin:ERATio:FETCh:DATA?.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example The examples of this command are is similar to :PLUGin:ERATio:FETCh:DATA?, with the only difference being that the measurement data are returned in tabular format. For details, refer to the examples of :PLUGin:ERATio:FETCh:DATA?.

:PLUGin:ERATio:FETCh:DATA:ALL?

Syntax :PLUGin:ERATio:FETCh:DATA:ALL? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Return Range 'M*.DataIn1' or 'M*.DataIn2' (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

Location, Time Stamp, Compared One Bits, Compared Zero Bits, Errored One Bits, Errored Zero Bits, Error Zero Ratio, Error One Ratio, Compared Bits, Errored Bits, Bit Error Ratio....

Counter for 8b/10b Bit Coding

Location, Time Stamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, ErroneousZeroRatio, ErroneousOneRatio, ComparedBits, ErroredBits, ErrorRatio, ComparedSymbols, ErroredSymbols, SymbolErrorRatio, IllegalSymbols, IllegalSymbolRatio, FillerSymbols, FillerSymbolRatio, WrongDisparity, DisparityErrorRatio, Frames,ErroredFrames, FrameErrorRatio....

Counter for 128/130 Bit Coding

Location, Time Stamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, ErroneousZeroRatio, ErroneousOneRatio, ComparedBits, ErroredBits, ErrorRatio, Blocks, ErroredBlocks, BlockErrorRatio, IllegalSyncHeaders, FillerSymbols, FillerSymbolRatio, ModifiedFillerSymbols, Frames,ErroredFrames, FrameErrorRatio...

Counter for 128/132 Bit Coding

Location, Time Stamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, ErroneousZeroRatio, ErroneousOneRatio, ComparedBits, ErroredBits, ErrorRatio, Blocks, ErroredBlocks, BlockErrorRatio, IllegalSyncHeaders, CorrectedSyncHeaders, FillerSymbols, FillerSymbolRatio, ModifiedFillerSymbols, Frames, ErroredFrames, FrameErrorRatio...

Counter for PCIe6 1b1b Coding

Location, Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, Error Zero Ratio, Error One Ratio, Compared Bits, Errored Bits, Error Ratio, Compared Symbols, Errored Symbols, Symbol Error Ratio, Filler Symbols, Modified Filler Symbols, Filler Count, Flit Count, FEC Corrected Bytes, CRC Error Count, PAM Results (SymbolLevel, Compared, Errored, ErrorRatio)...

The following examples shows the definition of the results returned for the different line codings:

NRZ

Location, Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, Error Zero Ratio, Error One Ratio, Compared Bits, Errored Bits, Error Ratio...

PAM3

Location, Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, Errored One Ratio, Errored Zero Ratio, Compared Bits, Errored Bits, Error Ratio, Total compared Symbols, Total Errored Symbols, Symbol Error Ratio, Level0, Level0 Compared Symbols, Level0 Erroneous Symbols, Level0 Symbol Error Ratio, Level1, Level1 Compared Symbols, Level1 Erroneous Symbols, Level1 Symbol Error Ratio, Level2, Level2 Compared Symbols, Level2 Erroneous Symbols, Level2 Symbol Error Ratio....

PAM4

Location, Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, Errored One Ratio, Errored Zero Ratio, Compared Bits, Errored Bits, Error Ratio, Total compared Symbols, Total Errored Symbols, Symbol Error Ratio, Level0, Level0 Compared Symbols, Level0 Erroneous Symbols, Level0 Symbol Error Ratio, Level1, Level1 Compared Symbols, Level1 Erroneous Symbols, Level1 Symbol Error Ratio, Level2, Level2 Compared Symbols, Level2 Erroneous Symbols, Level2 Symbol Error Ratio, Level3, Level3 Compared Symbols, Level3 Erroneous Symbols, Level3 Symbol Error Ratio...

PAM6

Location, Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, Errored One Ratio, Errored Zero Ratio, Compared Bits, Errored Bits, Error Ratio, Total compared Symbols, Total Errored Symbols, Symbol Error Ratio, Level0, Level0 Compared Symbols, Level0 Erroneous Symbols, Level0 Symbol Error Ratio, Level1, Level1 Compared Symbols, Level1 Erroneous Symbols, Level1 Symbol Error Ratio, Level2, Level2 Compared Symbols, Level2 Erroneous Symbols, Level2 Symbol Error Ratio, Level3, Level3 Compared Symbols, Level3 Erroneous Symbols, Level3 Symbol Error Ratio,Level4, Level4 Compared Symbols, Level4 Erroneous Symbols, Level4 Symbol Error Ratio, Level5, Level5 Compared Symbols, Level5 Erroneous Symbols, Level5 Symbol Error Ratio...

PAM8

Location, Time Stamp, Compared Ones, Compared Zeroes, Errored Ones, Errored Zeroes, Errored One Ratio, Errored Zero Ratio, Compared Bits, Errored Bits, Error Ratio, Total compared Symbols, Total Errored Symbols, Symbol Error Ratio, Level0, Level0 Compared Symbols, Level0 Erroneous Symbols, Level0 Symbol Error Ratio, Level1, Level1 Compared Symbols, Level1 Erroneous Symbols, Level1 Symbol Error Ratio, Level2, Level2 Compared Symbols, Level2 Erroneous Symbols, Level2 Symbol Error Ratio, Level3, Level3 Compared Symbols, Level3 Erroneous Symbols, Level3 Symbol Error Ratio, Level4, Level4 Compared Symbols, Level4 Erroneous Symbols, Level4 Symbol Error Ratio, Level5, Level5 Compared Symbols, Level5 Erroneous Symbols, Level5 Symbol Error Ratio, Level6, Level6 Compared Symbols, Level6 Erroneous Symbols, Level6 Symbol Error Ratio, Level7, Level7 Compared Symbols, Level7 Erroneous Symbols, Level7 Symbol Error Ratio...

This SCPI is applicable for M8041A, M8051A, M8043A, M8045A, M8046A and M8062A.

Example The following example shows the results returned for 128/130 Bit Coding.

:PLUG:ERAT:FETC:DATA:ALL? 'Error Ratio 1','M1.DataIn1'

Return:
("M1.DataIn1",1711532262.13709,2096726081,2096760919,1048354165,1048371584,0.499995767042432,0.4999957669721,4193487000,2096725749,0.499995767007266,32257589,32257589,1,0,0,0,0,0,0,0, ...)

If the location is omitted from the above example e.g. if the query is :PLUG:ERAT:FETC:DATA:ALL? 'Error Ratio 1' then the query returns the comma separated list of all the analyzer location group participating in the measurement.

Return:
("M1.DataIn1",1711532262.13709,2096726081,2096760919,1048354165,
1048371584,0.499995767042432,0.4999957669721,4193487000,209672
5749,0.499995767007266,32257589,32257589,1,0,0,0,0,0,0,0, ...),
("M1.DataIn1",1711532262.13709,2096726081,2096760919,1048354165,
1048371584,0.499995767042432,0.4999957669721,4193487000,209672
5749,0.499995767007266,32257589,32257589,1,0,0,0,0,0,0,0, ...)

The following examples shows the results returned for the different line codings:

NRZ

("M1.DataIn",1711513924.9324,3087686204,3087497668,3020,3161,1.02
380644130093E-06,9.78078664887541E-07,6175183872,6181,1.000941
85503145E-06, ...)

PAM3

("M1.DataIn",1711514244.20589,4080601982,8161209986,8131,8135,9.9
6788468126055E-07,1.99259815974868E-06,12241811968,16266,1.328
72486871381E-06,6120905984,12235,1.99888709808355E-06,0,204030
4002,4104,2.01146495619137E-06,1,2040297291,4005,1.962949231794
08E-06,2,2040304691,4126,2.0222469801693E-06, ...)

PAM4

("M1.DataIn",1711513043.68138,6125475936,6125101984,6129,6126,1.0
0014661241598E-06,1.00057531268375E-06,12250577920,12255,1.000
36096909296E-06,6125288960,12255,2.00072193818592E-06,0,153118
1984,3079,2.01086483002924E-06,1,1531369055,3044,1.987763818304
4E-06,2,1531368960,3107,2.02890360269546E-06,3,1531368961,3025,1
.97535674095461E-06, ...)

PAM6

("INF1.DataIn1",1711516441.77907,0,0,0,0,0,0,0,0,NaN,503240,0,0,0,879
69,0,0,1,75190,0,0,2,84035,0,0,3,84041,0,0,4,86492,0,0,5,85513,0,0,1711
516443.59262,0,0,0,0,0,0,0,0,NaN,503240,0,0,0,87971,0,0,1,75195,0,0,2,
84042,0,0,3,84026,0,0,4,86496,0,0,5,85510,0,0, ...)

PAM8

("INF1.DataIn1",1711523164.6691,750411,759309,84,156,0.0002054499
55156596,0.000111938657615627,1509720,240,0.000158969875208648
,503240,225,0.000447102774024322,0,67825,106,0.0015628455584224
1,1,65358,20,0.000306006915756296,2,60942,3,4.92271339962587E-05,
3,64371,1,1.55349458607137E-05,4,53082,18,0.000339097999321804,5,
71753,13,0.000181177093640684,6,60946,16,0.000262527483345913,7,
58963,48,0.000814069840408392, ...)

To get the measurement results in detailed or summary mode, refer to the :PLUGin:ERATio:SHOW:RVMode[?] on page 462.

:PLUGin:ERATio:FETCh:DATA:ALL:TABLe?

Syntax	:PLUGin:ERATio:FETCh:DATA:ALL:TABLe? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional)
Return Range	'M*.DataIn1' or 'M*.DataIn2' (data results in tabular format)
Description	This command returns the raw data of the error ratio measurement for the addressed measurement name identifier and location identifier in tabular format. It works in the same way as the command :PLUGin:ERATio:FETCh:DATA:ALL?, with the only difference being that it returns the measurement data in a tabular format. For more details, refer to the description of :PLUGin:ERATio:FETCh:DATA:ALL?. This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.
Example	The examples of this command are is similar to :PLUGin:ERATio:FETCh:DATA:ALL?, with the only difference being that the measurement data are returned in tabular format. For details, refer to the examples of :PLUGin:ERATio:FETCh:DATA:ALL?.

: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.

The following example shows the results returned when line coding is selected as PAM4:

("M2.DataIn",1511414904.34319,6064099210500,6064099210500,0,0,6064099210496,0,1516024802624,0,1516024802624,0,1516024802624,0,1516024802624,0,...)

To get the measurement results in detailed or summary mode, refer to the :PLUGin:ERATio:SHOW:RVMode[?] on page 462.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example :PLUG:ERAT:FETC? 'MyMeasurement','M1.DataIn1'

:PLUGin:ERATio:FETCh[:DURation]:TABLe?

Syntax :PLUGin:ERATio:FETCh[:DURation]:TABLe? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (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 in the tabular format.

It works in the same way as the command :PLUGin:ERATio:FETCh[:DURation]?, with the only difference being that it returns the measurement data in a tabular format.

For more details, refer to the description of :PLUGin:ERATio:FETCh[:DURation]?.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example The examples of this command are is similar to :PLUGin:ERATio:FETCh[:DURation]?, with the only difference being that the measurement data are returned in tabular format. For details, refer to the examples of :PLUGin:ERATio:FETCh[:DURation]?.

:PLUGin:ERATio:FETCh[:DURation]:DETailed?

Syntax :PLUGin:ERATio:FETCh[:DURation]:DETailed? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

'Location': Optionally specify the error detector. This reduces the returned results to the given error detector location. If this is not given, then all the error detectors that are being used by the measurement are reported in the response.

Description This query returns the error counter results in full detail.

The response of this query is identical to :PLUGin:ERATio:FETCh[:DURation]? 'Identifier'[,'Location'] for all PHY-Protocols and Line-Codings, except for PAM4.

When the error detector is receiving PAM4 symbols, the PAM4 counters are more detailed and report the number of times a certain symbol level has been received for every expected symbol level.

The results are combined to an expression. An expression is comparable to a structure in a programming language. It's a set of parameters of different types separated be a comma and enclosed by parenthesis.

The returned expression contains the location, a timestamp and a set of symbol/bit counter values per threshold level. The returned expression contains counters according to the line coding. If the line coding is PAM4 then symbol counters are returned for four levels and in case of NRZ the bit counters are returned for two levels.

Returned expression in case of PAM4:

("Location", TimeStamp, SymbolErrorRatio, ComparedSymbols, ErroneousSymbols, NLevels,(

(Level0, Level0SymbolErrorRatio, Level0ComparedSymbols, Level0ErroneousSymbols,

(RecievedLevel0, RecievedLevel1, RecievedLevel2, RecievedLevel3)),

(Level1, Level1SymbolErrorRatio, Level1ComparedSymbols, Level1ErroneousSymbols,

(RecievedLevel0, RecievedLevel1, RecievedLevel2, RecievedLevel3)),

(Level2, Level2SymbolErrorRatio, Level2ComparedSymbols, Level2ErroneousSymbols,

(RecievedLevel0, RecievedLevel1, RecievedLevel2, RecievedLevel3)), (Level3, Level3SymbolErrorRatio, Level3ComparedSymbols, Level3ErroneousSymbols,

(RecievedLevel0, RecievedLevel1, RecievedLevel2, RecievedLevel3))))

Location: It's a string referred to the analyzer location for which symbol errors are calculated.

Timestamp: As timestamp the UNIX epoch time is used; it is defined as elapsed seconds, milliseconds etc. since 01.01.1970 UTC.

SymbolErrorRatio: Symbol Error Ratio calculated from total errored and compared symbols.

ComparedSymbols: No of total compared symbols.

ErroneousSymbols: No of total erroneous symbols.

NLevels: No of symbol levels compared i.e. 4 in case of PAM4.

Level0-N: Symbol Level no for the following counter values, 0 to 3 in case for PAM4.

Level0-NSymbolErrorRatio: Symbol error ratio for level N

Level0-NComparedSymbols: No of expected/compared symbols for level N

Level0-NErroneousSymbols: No of erroneous symbols for symbol level N

RecievedLevel0: No of received Symbol Level 0 for expected Symbol Level N

RecievedLevel1: No of received Symbol Level 1 for expected Symbol Level N

RecievedLevel2: No of received Symbol Level 2 for expected Symbol Level N

RecievedLevel3: No of received Symbol Level 3 for expected Symbol Level N

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example Identifier: 'Error Ratio 1'

:PLUGin:ERATio:FETCh:DURation:DETailed? 'Error Ratio 1'

Response for Line coding PAM4:

("M1.DataIn1",1558690992.76551,0.000125,40000,5,4,((0,0.00016949,11800,2,(11798,2,0,0)),(1,0.0002439,8200,2,(0,8198,2,0)),(2,0.00009091,11000,1,(0,0,10999,0)),(3,0.0,9000,0,(0,0,0,9000))))

: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 applicable for M8041A, M8051A, M8062A and M8046A.
Example	:PLUG:ERAT:FETC:PFR? 'MyMeasurement','M1.DataIn1'

:PLUGin:ERATio:FETCh[:DURation]:PFResult:DETailed?

Syntax	:PLUGin:ERATio:FETCh[:DURation]:DETailed? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. 'Location': Optionally specify the error detector. This reduces the returned results to the given error detector location. If this is not given, then all the error detectors that are being used by the measurement are reported in the response.
Description	This query is reporting the full counter results as documented for :PLUGin:ERATio:FETCh[:DURation]:DETailed? combined with the pass/fail results.

It returns exactly the same result, and appends the following pass fail results to the end of the response.

- ConfidenceLevelAtTargetErrorRatio
- MinimumErrorRatioAtTargetConfidenceLevel
- MaximumErrorRatioAtTargetConfidenceLevel
- PassFailedResult

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example

Example: for Line-Coding PAM4

:PLUGin:ERATio:FETCh:DURation:PFResult:DETailed? "Error Ratio 1"

Response:

("M2.DataIn",1572199686.69608,712958676914,712958676914,0,0,712958676912,0,4,(0,0,178239669228,0,(178239669228,0,0,0)),(1,0,178239669228,0,(0,178239669228,0,0)),(2,0,178239669228,0,(0,0,178239669228,0)),(3,0,178239669228,0,(0,0,0,178239669228)),1,0,2.10091578274974E-12,"PASS")

: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 applicable for M8041A, M8051A, M8062A and M8046A.

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:TABS?

Syntax :PLUGin:ERATio:TABS?

Description This query returns a list of all results tabs which can be exported as a CSV file.

Example :PLUG:ERAT:TABS?

Output “M*.DataIn”, "Calculated Results"

:PLUGin:ERATio:EXPort

Syntax :PLUGin:ERATio:EXPort 'Identifier', 'TabName', 'FilePath'

Description This command exports the measurement data of the selected results tab to a csv file.

Input Parameters

- 'Identifier': Specify the measurement name; for example ''Error Ratio 1''.
- 'TabName' – Specify the results tab name. To fetch the results tabs for a measurement, refer to the query “:PLUGin:ERATio:TABS? on page 475”. However, if you want all tabs to be saved as a csv file, use " " (single/double quotes).
- 'FilePath' – Specify the location where the CSV file should be saved.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Examples Example 1: The following command will save the 'Calculated Results' tab of the error ratio measurement to *Results.csv*:

:PLUGin:ERAT:EXPort 'Error Ratio 1', 'Calculated Results', 'C:\Results.csv'

Example 2: The following command will save all results tab of the error ratio measurement to *Results.csv*:

:PLUGin:ERAT:EXPort 'Error Ratio 1', ' ', 'C:\Results.csv'

: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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

Example :PLUG:ERAT:SHOW:SGL 'MyMeasurement', 1

:PLUGin:ERATio:SHOW:COMment[?]

Syntax	:PLUGin:ERATio:SHOW:COMment 'Identifier', <“Comment”> :PLUGin:ERATio:SHOW:COMment? 'Identifier'
Input Parameters	Identifier: Specify the measurement name. For example; 'MyMeasurement' <“Comment”>: Enter the desired comment.
Description	This command is used to enter a comment for each measurement. This comment will be displayed on the **Measurement History** tab. This SCPI is applicable for M8041A, M8051A, M8062A, M8045A and M8046A.
Example	:PLUG:ERAT:SHOW:COM 'MyMeasurement', “Eye Measurement Example” :PLUG:ERAT:SHOW:COM? 'MyMeasurement' Eye Measurement Example

:PLUGin:ERATio:GRAPh?

Syntax	:PLUGin:ERATio:GRAPh? 'identifier' [,DesiredWidth][,DesiredHeight][,<CURR | WHIT>][,<PNG>][,1 | 0][,'TabName']
Input Parameters	'identifier': Specify measurement name. Other optional parameters are: [,DesiredWidth]: Specify the desired width of the image. [,DesiredHeight]: Specify the desired height of the image. [,<CURR | WHIT>]: Specify whether the user wants to capture in current theme or wants to capture in white background. [,<PNG>]: Specify the format of the image. The default format is PNG. [,1 | 0]: Specify whether to capture the graph with legends or not. [,’TabName’]: Specify the tab name on which the graph is supposed to be captured.
Description	This query returns data of the image captured from the graph view of the plugin in the specified format.
Example	:PLUG:ERAT:GRAP? 'MyMeasurement',1000,800,CURR,PNG,1,'Graph'

: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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.

Example The following example returns logs for the measurement name identifier called 'MyMeasurement':

:PLUGin:ERATio:RUN:LOG? 'MyMeasurement'

The format of log returned is in the following way:

```
#XY Message Log
```

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

For example:

#10 means, there is 1 digit after 1 and there is 0 character in the log.

#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

: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 applicable for M8041A, M8051A, M8061, M8062A and M8046A.

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 applicable for M8041A, M8051A, M8062A and M8046A.
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 applicable for M8041A, M8051A, M8062A and M8046A.
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:CCAPture Subnode

The M8020A/M8040A/M8050A 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 error 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 hold-off 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 86

Name	Description under
:CATalog?	:PLUGin:CCAPture:CATalog? on page 484
DELete[?]	:PLUGin:CCAPture:DELete on page 484
NEW[?]	:PLUGin:CCAPture:NEW on page 484
RESet[?]	:PLUGin:CCAPture:RESet on page 485
STARt[?]	:PLUGin:CCAPture:STARt on page 485
STOP[?]	:PLUGin:CCAPture:STOP on page 485
SAVE[?]	:PLUGin:CCAPture:SAVE on page 486
:TRIGger:IMMediate[:ONCE]	:PLUGin:CCAPture:TRIGger:IMMediate[:ONCE] on page 486
:SHOW:BIT:PATTern:VIEW	:PLUGin:CCAPture:SHOW:BIT[:PATTern][:VIEW][?] on page 487
:SHOW:SYMBol:PATTern:VIEW	:PLUGin:CCAPture:SHOW:SYMBol[:PATTern][:VIEW][?] on page 487
Subnodes	
FETCh	:PLUGin:CCAPture:FETCh Subnode on page 488
ACQuisition	:PLUGin:CCAPture:ACQuisition Subnode on page 494
RUN	:PLUGin:CCAPture:RUN Subnode on page 498

: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.

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.

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.

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.

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.

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'.

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.

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.

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.
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.
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]?
└─ :ERRor:SYMBol
    └[:COUNt]?
```

This subnode has the following commands:

Table 87

Name	Description under
:ERRor:COUNt?	:PLUGin:CCAPture:FETCh:ERRor:COUNt? on page 489
:PATTern[:DATA]:HOLDoff?	:PLUGin:CCAPture:FETCh:PATTern[:DATA]:HOLDoff? on page 489
:PATTern[:DATA]:DEPTh?	:PLUGin:CCAPture:FETCh:PATTern[:DATA]:DEPTh? on page 490
:PATTern:ERRor[:SYMBol]:FIRSt?	:PLUGin:CCAPture:FETCh:PATTern:ERRor[:SYMBol]:FIRSt? on page 490

Name	Description under
:PATTern:ERRor[:SYMBol]:LAST?	:PLUGin:CCAPture:FETCh:PATTern:ERRor[:SYMBol]:LAST? on page 490
:RUN[:STATus]?	:PLUGin:CCAPture:FETCh:RUN[:STATus]? on page 491
:ERRor:SYMBol[:COUNt]?	:PLUGin:CCAPture:FETCh:ERRor:SYMBol[:COUNt]? on page 492

: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 query gives the error count of the captured data. If simply run without specifying the location, then it returns the results for the current run.

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': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2'

Description This query 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.

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': Specify the measurement name [,'Location']: 'M*.DataIn1' or 'M*.DataIn2'
Description	This query returns the total number of data bits captured. Returned value of this query is only valid when capture measurement is finished.
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' [,M8131A/M8132A: SW Release V4.3.1.0Location]
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2'
Description	This query fetches the first error symbol.
Example	:PLUGin:CCAPture:FETCh:PATTern:ERRor: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': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2'
Description	This query fetches the last error symbol.
Example	:PLUGin:CCAPture:FETCh:PATTern:ERRor:LAST? 'MyMeasurement','M1.DataIn1' Return: ("M1.DataIn1", ...)

:PLUGin:CCAPture:FETCh:RUN[:STATus]?

Syntax	:PLUGin:CCAPture:FETCh:RUN[:STATus]? 'identifier' [,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2'
Description	This query 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.
Example	:PLUGin:CCAPture:FETCh:RUN:STATus? 'MyMeasurement','M1.DataIn1' Return: ("M1.DataIn1", ...)

:PLUGin:CCAPture:FETCh:ERRor:SYMBol[:COUNt]?

Syntax :PLUGin:CCAPture:FETCh:ERRor:SYMBol[:COUNt]? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This query returns the symbol error counts per threshold level. The results are combined to an expression. An expression is comparable to a structure in a programming language. It's a set of parameters of different types separated be a comma and enclosed by parenthesis.

The returned expression contains the location, a timestamp and a set of symbol/bit counter values per threshold level. The returned expression contains counters according to the line coding. If the line coding is PAM4 then symbol counters are returned for four levels and in case of NRZ the bit counters are returned for two levels.

Returned expression in case of PAM4:

("Location", TimeStamp, SymbolErrorRatio, ComparedSymbols, ErroneousSymbols, NLevels,(

(Level0, Level0SymbolErrorRatio, Level0ComparedSymbols, Level0ErroneousSymbols, (RecievedLevel0, RecievedLevel1, RecievedLevel2, RecievedLevel3)),

(Level1, Level1SymbolErrorRatio, Level1ComparedSymbols, Level1ErroneousSymbols, (RecievedLevel0, RecievedLevel1, RecievedLevel2, RecievedLevel3)),

(Level2, Level2SymbolErrorRatio, Level2ComparedSymbols, Level2ErroneousSymbols, (RecievedLevel0, RecievedLevel1, RecievedLevel2, RecievedLevel3)),

(Level3, Level3SymbolErrorRatio, Level3ComparedSymbols, Level3ErroneousSymbols, (RecievedLevel0, RecievedLevel1, RecievedLevel2, RecievedLevel3))

))

Location: It's a string referred to the analyzer location for which symbol errors are calculated.

Timestamp: As timestamp the UNIX epoch time is used; it is defined as elapsed seconds, milliseconds etc. since 01.01.1970 UTC.

SymbolErrorRatio: Symbol Error Ratio calculated from total erroroneous and compared symbols.

ComparedSymbols: No of total compared symbols.

ErroneousSymbols: No of total erroneous symbols.

NLevels: No of levels compared i.e. 4 in case of PAM4 and 2 in case of NRZ.

Level0-N: Level no that specifies counters start for specific level 0 to 3 in case for PAM4 and 0 or 1 in case of NRZ.

Level0-NSymbolErrorRatio: Symbol error ratio for level N

Level0-NComparedSymbols: No of expected/compared symbols for level N

Level0-NErroneousSymbols: No of erroneous symbols for level N

RecievedLevel0: No of received Level 0 for expected Level N

RecievedLevel1: No of received Level 1 for expected Level N

RecievedLevel2: No of received Level 2 for expected Level N

RecievedLevel3: No of received Level 3 for expected Level N

Example :PLUGin:CCAPture:FETCh:ERRor:SYMBol:COUNt? 'Pattern Capture 1'

Return:

Line coding PAM4

("M1.DataIn1",1558690992.76551,0.000125,40000,5,4,((0,0.00016949,11800,2,(11798,2,0,0)),(1,0.0002439,8200,2,(0,8198,2,0)),(2,0.00009091,11000,1,(0,0,10999,0)),(3,0.0,9000,0,(0,0,0,9000))))

Line coding NRZ

("M1.DataIn1",1558690992.76551,0.0002,20000,4,2,((0,0.00016667,12000,2,(11998,2)),(1,0.00025,8000,2,(2,7998))))

:PLUGin:CCAPture:ACQuisition Subnode

This subnode has the following SCPI structure:

```
:ACQUisition
├:ALOCation[?]
├:DEPTh[?]
├:HOLDoff[?]
├:SOURce[?]
├:CIMemory[?]
└:CAPOnly[?]
```

This subnode has the following commands:

Table 88

Name	Description under
:ALOCation[?]	:PLUGin:CCAPture:ACQuisition:ALOCation[?] on page 495
:DEPTh[?]	:PLUGin:CCAPture:ACQuisition:DEPTh[?] on page 495
:HOLDoff[?]	:PLUGin:CCAPture:ACQuisition:HOLDoff[?] on page 495
:SOURce[?]	:PLUGin:CCAPture:ACQuisition:SOURce[?] on page 496
:SLOPe[?]	:PLUGin:CCAPture:ACQuisition:SLOPe[?] on page 496
:CIMemory[?]	:PLUGin:CCAPture:ACQuisition:CIMemory[?] on page 496
:CAPOnly[?]	:PLUGin:CCAPture:ACQuisition:CAPOnly[?] on page 497

:PLUGin:CCAPture:ACQuisition:ALOCation[?]

Syntax	:PLUGin:CCAPture:ACQuisition:LOCation 'identifier','Location'
	:PLUGin:CCAPture:ACQuisition:LOCation? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
	[,'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 accepts error detector locations of M8041A, M8051A, M8062A, and error detector channels of all supported real-time oscilloscopes.
Example	PLUGin:CCAPture:ACQuisition:LOCation 'My Measurement', 'M1.DataIn1'

:PLUGin:CCAPture:ACQuisition:DEPTh[?]

Syntax	:PLUGin:CCAPture:ACQuisition:DEPTh 'identifier',<NRf>
	:PLUGin:CCAPture:ACQuisition:DEPTh? 'identifier'
Input Parameters	'identifier': Specify the measurement name.
Range	0 to 2147483648
Description	This command configures how long the capture logic will write the capture memory after detecting the trigger event.
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': Specify the measurement name.
Range	0 to 2147483648
Description	Specifies the amount of bits the trigger event is ignored after starting capturing of data.
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.

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.

Example :PLUGin:CCAPture:ACQuisition:SLOPe 'My Measurement', POS

:PLUGin:CCAPture:ACQuisition:CIMemory[?]

Syntax :PLUGin:CCAPture:ACQuisition:CIMemory 'identifier', <ON|OFF|1|0>

:PLUGin:CCAPture:ACQuisition:CIMemory? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

< ON | OFF | 1 | 0>

Return Range 1 | 0

Description This commands allows the user to capture the data in memory.

Example :PLUG:CCAP:ACQ:CIM 'My Measurement', ON

:PLUG:CCAP:ACQ:CIM? 'My Measurement'

1

:PLUGin:CCAPture:ACQuisition:CAPOnly[?]

Syntax	:PLUGin:CCAPture:ACQuisition:CAPOnly 'identifier', <ON|OFF|1|0> :PLUGin:CCAPture:ACQuisition:CAPOnly? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <ON | OFF | 1 | 0>
Return Range	1 | 0
Description	This commands allows the user to capture data without any comparison. So use this command to if you don't care about the comparison and also don’t want to save the error bits.
Example	:PLUG:CCAP:ACQ:CAPO 'My Measurement', ON :PLUG:CCAP:ACQ:CAPO? 'My Measurement' 1

:PLUGin:CCAPture:RUN Subnode

This subnode has the following SCPI structure:

```
:RUN
├:MESSage?
├:PROGress?
├:STATus?
└:LOG?
```

This subnode has the following commands:

Table 89

Name	Description under
:MESSage?	:PLUGin:CCAPture:RUN:MESSage? on page 499
:PROGress?	:PLUGin:CCAPture:RUN:PROGress? on page 499
:STATus?	:PLUGin:CCAPture:RUN:STATus? on page 500
LOG?	:PLUGin:CCAPture:RUN:LOG? on page 500

: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.
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.
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.

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'

The format of log returned is in the following way:

`#XY Message Log`

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

For example:

#10 means, there is 1 digit after 1 and there is 0 character in the log.

#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

:PLUGin:LFPSgen Subnode

The LFPS Generator utility is used to generate LFPS patterns and sequences using the M8070B system software. It allows for the configuration of LFPS cycles, period, duty cycle, pre-data duration, and electrical idle duration. This utility is supported by the M8042A and M8045A pattern generator modules. The LFPS Generator supports USB4v2 Gen 2 (10 Gbps), Gen 3 (20 Gbps), and Gen 4 (25.6 Gbps) data rates. For USB4v2 Gen 2 and Gen 3, NRZ line coding is used, while for USB4v2 Gen 4, PAM3 line coding is used.

This subnode has the following SCPI structure:

```
:LFPSgen
├─:CATalog
├─:DELete
├─:NEW
├─:RESet
├─:STARt
├─:STOP
├─:TABS?
├─:RUN
│ ├─:LOG?
│ ├─:MESSage?
│ ├─:PROGress?
│ └─[:STATus]?
├─:INSTrument
│ └─:LOCation[?]
├─:CYCLes[?]
├─:DRATe?
├─:DUTYCycle[?]
├─:EIDuration[?]
├─:GENerate
├─:LINE
│ └─:CODing?
├─:PATTern?
├─:PROTocol[?]
├─:PERiod[?]
├─:PREData[?]
│ └─:ADJusted?
└─:TEST
  └─:MODE[?]
```

This subnode has the following commands:

Table 90

Name	Description under
:PLUGin:LFPSgen:CATalog?	:PLUGin:LFPSgen:CATalog? on page 503
:PLUGin:LFPSgen:DELete	:PLUGin:LFPSgen:DELete on page 503
:PLUGin:LFPSgen:NEW	:PLUGin:LFPSgen:NEW on page 503
:PLUGin:LFPSgen:RESet	:PLUGin:LFPSgen:RESet on page 504
:PLUGin:LFPSgen:STARt	:PLUGin:LFPSgen:STARt on page 504
:PLUGin:LFPSgen:STOP	:PLUGin:LFPSgen:STOP on page 505
:PLUGin:LFPSgen:TABS?	:PLUGin:LFPSgen:TABS? on page 505
:PLUGin:LFPSgen:RUN:LOG?	:PLUGin:LFPSgen:RUN:LOG? on page 505
:PLUGin:LFPSgen:RUN:MESSage?	:PLUGin:LFPSgen:RUN:MESSage? on page 506
:PLUGin:LFPSgen:RUN:PROGress?	:PLUGin:LFPSgen:RUN:PROGress? on page 506
:PLUGin:LFPSgen:RUN[:STATus]?	:PLUGin:LFPSgen:RUN[:STATus]? on page 507
:PLUGin:LFPSgen:INSTrument:LOCation[?]	:PLUGin:LFPSgen:INSTrument:LOCation[?] on page 507
:PLUGin:LFPSgen:CYCLes[?]	:PLUGin:LFPSgen:CYCLes[?] on page 508
:PLUGin:LFPSgen:DRATe?	:PLUGin:LFPSgen:DRATe? on page 508
:PLUGin:LFPSgen:DUTYCycle[?]	:PLUGin:LFPSgen:DUTYCycle[?] on page 508
:PLUGin:LFPSgen:EIDuration[?]	:PLUGin:LFPSgen:EIDuration[?] on page 509
:PLUGin:LFPSgen:GENerate	:PLUGin:LFPSgen:GENerate on page 509
:PLUGin:LFPSgen:LINE:CODing?	:PLUGin:LFPSgen:LINE:CODing? on page 509
:PLUGin:LFPSgen:PATTern?	:PLUGin:LFPSgen:PATTern? on page 510
:PLUGin:LFPSgen:PROTocol[?]	:PLUGin:LFPSgen:PROTocol[?] on page 510
:PLUGin:LFPSgen:PERiod[?]	:PLUGin:LFPSgen:PERiod[?] on page 510
:PLUGin:LFPSgen:PREData[?]	:PLUGin:LFPSgen:PREData[?] on page 511
:PLUGin:LFPSgen:PREData:ADJusted?	:PLUGin:LFPSgen:PREData:ADJusted? on page 511
:PLUGin:LFPSgen:TEST:MODE[?]	:PLUGin:LFPSgen:TEST:MODE[?] on page 511

:PLUGin:LFPSgen:CATalog?

Syntax	:PLUGin:LFPSgen:CATalog?
Description	This command returns a list of all created LFPS Generator plugin names currently available for measuring. This SCPI is applicable for M8042A and M8045A.
Example	Assume the following is a list of created LFPS Generator plugins: :PLUG:LFPS:NEW 'LFPS_1' :PLUG:LFPS:NEW 'LFPS_2' :PLUG:LFPS:NEW 'LFPS_3' The command and returned list would look like the following: :PLUG:LFPS:CAT? 'LFPS_1,LFPS_2,LFPS_3'

:PLUGin:LFPSgen:DELete

Syntax	:PLUGin:LFPSgen:DELete 'identifier'
Input Parameters	'identifier': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'.
Description	This command creates a new LFPS Generator plugin with the given name. This SCPI is applicable for M8042A and M8045A.
Example	The following example deletes LFPS plugin addressed by the plugin name identifier called 'LFPS Generator 1': :PLUG:LFPS:DEL 'LFPS Generator 1'

:PLUGin:LFPSgen:NEW

Syntax	:PLUGin:LFPSgen:NEW 'identifier'
Input Parameters	'identifier': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'.
Description	This command creates a new LFPS Generator plugin with the given name. This SCPI is applicable for M8042A and M8045A.

Example — The following example creates a LFPS generator plugin called 'LFPS Generator 1':

:PLUG:LFPS:NEW 'LFPS Generator 1'

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:LFPS:NEW 'LFPS Generator 1',0

:PLUGin:LFPSgen:RESet

Syntax — :PLUGin:LFPSgen:RESet 'identifier'

Input Parameters — 'identifier': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'.

Description — This command resets the LFPS Generator plugin addressed by the plugin name identifier.

This SCPI is applicable for M8042A and M8045A.

Example — The following example resets the LFPS Generator plugin addressed with the plugin name identifier called 'LFPS Generator 1':

:PLUG:LFPS:RES 'LFPS Generator 1'

:PLUGin:LFPSgen:STARt

Syntax — :PLUGin:LFPSgen:STARt 'identifier'

Input Parameters — 'identifier': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'.

Description — This command starts a LFPS Generator plugin addressed by the LFPS generator plugin name identifier.

This SCPI is applicable for M8042A and M8045A.

Example — The following example starts a LFPS Generator plugin addressed by the LFPS Generator plugin name identifier called 'LFPS Generator 1':

:PLUG:LFPS:STAR 'LFPS Generator 1'

:PLUGin:LFPSgen:STOP

Syntax — :PLUGin:LFPSgen:STOP 'identifier'

Input Parameters — 'identifier': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'.

Description — This command stops LFPS Generator plugin addressed by the LFPS Generator plugin identifier.

This SCPI is applicable for M8042A and M8045A.

Example — The following example stops LFPS Generator plugin addressed by the LFPS Generator plugin identifier called 'LFPS Generator 1':

:PLUG:LFPS:STOP 'LFPS Generator 1'

:PLUGin:LFPSgen:TABS?

Syntax — :PLUGin:LFPSgen:TABS?

Description — This query returns a list of all results tabs which can be exported as a CSV file.

This SCPI is applicable for M8042A and M8045A.

Example — :PLUG:EDI:TABS? -> "Calculated Results"

:PLUGin:LFPSgen:RUN:LOG?

Syntax — :PLUGin:LFPSgen:RUN:LOG? 'identifier'

Input Parameters — 'identifier': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'.

Description — This command returns logs for the addressed LFPS Generator plugin.

This SCPI is applicable for M8042A and M8045A.

Example — The following example returns logs for the LFPS Generator plugin identifier called 'LFPS Generator 1':

:PLUGin:LFPSgen:RUN:LOG? 'LFPS Generator 1'

The format of log returned is in the following way:

`#XY Message Log`

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

For example:

#10 means, there is 1 digit after 1 and there is 0 character in the log.

#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

:PLUGin:LFPSgen:RUN:MESSage?

Syntax	:PLUGin:LFPSgen:RUN:MESSage? 'identifier'
Input Parameters	'identifier': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'.
Description	This query returns a string describing the state of LFPS Generator plugin addressed by the LFPS Generator plugin name identifier. Possible states include NotStarted, Running, Finished, Error, or Stopped. This SCPI is applicable for M8042A and M8045A.
Example	The following example returns the state of LFPS Generator plugin with the LFPS Generator plugin name identifier called 'LFPS Generator 1': :PLUG:LFPS:RUN:MESS? 'LFPS Generator 1' Running

:PLUGin:LFPSgen:RUN:PROGress?

Syntax	:PLUGin:LFPSgen:RUN:PROGress? 'identifier'
Input Parameters	'identifier': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'.
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 LFPS Generator plugin addressed by the LFPS Generator plugin name identifier. A 0.0 indicates that the LFPS generation has not

started and 1.0 indicates the LFPS generation is finished.

This SCPI is applicable for M8042A and M8045A.

Example: The following example returns the progress of LFPS Generator plugin with the LFPS Generator plugin name identifier called 'LFPS Generator 1':

:PLUG:ERAT:RUN:PROG? 'LFPS Generator 1'
0.51

:PLUGin:LFPSgen:RUN[:STATus]?

Syntax: :PLUGin:LFPSgen:RUN[:STATus]? 'identifier'

Input Parameters: 'identifier': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'.

Return Range: 0|1

Description: This command returns the running status of LFPS Generator plugin addressed by the LFPS Generator plugin name identifier. A 0 indicates the LFPS Generator plugin is not running and a 1 indicates the LFPS Generator plugin is running.

This SCPI is applicable for M8042A and M8045A.

Example: The following example returns the running status of LFPS Generator plugin with the LFPS Generator plugin name identifier called 'LFPS Generator 1':

:PLUG:LFPS:RUN 'LFPS Generator 1', 1

:PLUGin:LFPSgen:INSTrument:LOCation[?]

Syntax: :PLUGin:LFPSgen:INSTrument:LOCation, '<Identifier> ', '<location-string>'

:PLUGin:LFPSgen:INSTrument:LOCation? '<Identifier>'

Input Parameters: '<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'

<location-string>: 'M*.DataOut*'

Description: This command sets the LFPS Generator data out location which is used to communicate with DUT RX.

Example: :PLUG:LFPS:INST:LOC 'LFPS Generator 1','M1.DataOut1'

:PLUG:LFPS:INST:LOC? 'LFPS Generator 1'

:PLUGin:LFPSgen:CYCLes[?]

Syntax	:PLUGin:LFPSgen:CYCLes '<Identifier>', <Cycles>
	:PLUGin:LFPSgen:CYCLes? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'
	<Cycles>: Number of LFPS cycles
Range	1 to 100
Description	This command sets the number of LFPS cycles.
Example	:PLUG:LFPS:CYCL 'LFPS Generator 1',10
	:PLUG:LFPS:CYCL? 'LFPS Generator 1'

:PLUGin:LFPSgen:DRATe?

Syntax	:PLUGin:LFPSgen:DRATe? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'
Description	This query returns the data rate used for LFPS generation based on the selected protocol.
Example	:PLUG:LFPS:DRAT? 'LFPS Generator 1'

:PLUGin:LFPSgen:DUTYCycle[?]

Syntax	:PLUGin:LFPSgen:DUTYCycle '<Identifier>', <Duty Cycle>
	:PLUGin:LFPSgen:DUTYCycle? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'
	<Duty Cycle>: Number of LFPS duty cycle
Range	45% - 55%
Description	This command sets the number of LFPS duty cycle.
Example	:PLUGin:LFPSgen:DUTYCycle 'LFPS Generator 1',50
	:PLUGin:LFPSgen:DUTYCycle? 'LFPS Generator 1

:PLUGin:LFPSgen:EIDuration[?]

Syntax	:PLUGin:LFPSgen:EIDuration '<Identifier>', <EI Duration> :PLUGin:LFPSgen:EIDuration? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1' <EI Duration>: Electrical ideal duration in ns.
Range	20-80 ns
Description	This command sets the electrical ideal duration in ns.
Example	:PLUG:LFPS:EID 'LFPS Generator 1',2E-08 :PLUG:LFPS:EID? 'LFPS Generator 1

:PLUGin:LFPSgen:GENerate

Syntax	:PLUGin:LFPSgen:GENerate '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin name, e.g., 'LFPS Generator 1'
Description	This command downloads the LFPS patterns and sequence to the selected generator location.
Example	:PLUG:LFPS:GEN 'LFPS Generator 1

:PLUGin:LFPSgen:LINE:CODing?

Syntax	:PLUGin:LFPSgen:LINE:CODing? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'
Description	This query returns the line coding to be used for LFPS generation based on the selected protocol. For USBG2 and USBG3 protocols, this query returns NRZ and for USBG4 protocol, this query returns PAM3.
Example	:PLUG:LFPS:LINE:COD? 'LFPS Generator 1

:PLUGin:LFPSgen:PATTern?

Syntax	:PLUGin:LFPSgen:PATTern? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'
Description	This query returns the name of the test pattern.
Example	:PLUG:LFPS:PATT? 'LFPS Generator 1

:PLUGin:LFPSgen:PROTocol[?]

Syntax	:PLUGin:LFPSgen:PROTocol '<Identifier>', <Protocol>
	:PLUGin:LFPSgen:PROTocol? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'
	< Protocol >: USBG2 | USBG3 | USBG4
Description	This command sets the USB4 protocol to be used for LFPS.
Example	:PLUG:LFPS:PROT 'LFPS Generator 1',USBG4
	:PLUG:LFPS:PROT? 'LFPS Generator 1

:PLUGin:LFPSgen:PERiod[?]

Syntax	:PLUGin:LFPSgen:PERiod '<Identifier>',<Period>
	:PLUGin:LFPSgen:PERiod? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'
	<Period>: Period in ns
Range	20-80 ns
Description	This command sets the LFPS period in ns.
Example	:PLUG:LFPS:PER 'LFPS Generator 1',5E-08
	:PLUG:LFPS:PER? 'LFPS Generator 1

:PLUGin:LFPSgen:PREData[?]

Syntax	:PLUGin:LFPSgen:PREData '<Identifier>', <Pre Data>
	:PLUGin:LFPSgen:PREData? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'
	<Pre Data>: LFPS pre data duration in ns.
Range	80 to 120 ns
Description	This command sets the LFPS pre-data duration in ns.
Example	:PLUG:LFPS:PRED 'LFPS Generator 1',1E-07
	:PLUG:LFPS:PRED? 'LFPS Generator 1

:PLUGin:LFPSgen:PREData:ADJusted?

Syntax	:PLUGin:LFPSgen:PREData:ADJusted? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'
Description	This command returns the LFPS pre-data duration after the pattern block granularity adjustment in ns.
Example	:PLUG:LFPS:PRED:ADJ? 'LFPS Generator 1

:PLUGin:LFPSgen:TEST:MODE[?]

Syntax	:PLUGin:LFPSgen:TEST:MODE '<Identifier>', <CAL | RXT>
	:PLUGin:LFPSgen:TEST:MODE? '<Identifier>'
Input Parameters	'<identifier>': Specify LFPS Generator plugin, e.g., 'LFPS Generator 1'
	CAL - Calibration
	RXT - RX Testing
Description	This command sets the USB4 protocol for LFPS. The options are 'Calibration' and 'RX Testing'.
Example	:PLUG:LFPS:TEST:MODE 'LFPS Generator 1', CAL
	:PLUG:LFPS:TEST:MODE? 'LFPS Generator 1

Advanced Measurement Package

The Advanced Measurement Package provides the following measurements when it is installed on M8070B System Software:

- Output timing measurement
- Output level measurement
- Jitter tolerance measurement
- Eye diagram measurement
- Parameter sweep
- BER Based Deemphasis

NOTE **Advanced Measurement Package requires a valid license for activation.**

For more details on the Advanced Measurement Package and its remote programming, please refer to *M8000 Series of Advanced Measurement Package User Guide* and *Online Help*.

Error Distribution Analysis Package

The Error Distribution Analysis package provides the following measurement when it is installed on M8070B System Software:

- Error Distribution Analysis

NOTE **Error Distribution Analysis Package requires a valid license for activation.**

For more details on the Error Distribution Analysis Package and its remote programming, please refer to *M8000 Series of Error Distribution Analysis Package User Guide* and *Online Help*.

M8070ISIB Adjustable ISI Package for M8000 Series

The M8070B system software provides integrated and adjustable Intersymbol Interference (ISI) capability to test next-generation high-speed digital designs. The ISI package is programmable for each M8042A and M8045A pattern generator channel and simplifies receiver test automation for data rates up to 64 GBd.

NOTE

M8070ISIB Adjustable ISI Package requires a valid license for activation.

For more details on the M8070ISIB Adjustable ISI Package and its remote programming, please refer to *M8070ISIB Adjustable ISI Package for M8000 Series User Guide* and *Online Help*.

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?
││└:DETails
││ ├:JSON?
││ ├:VALue?
││ └:XML?
│└:LICenses
│ ├:HOST[:LIST]?
│ └:MODules[:LIST]?
├:COMMunicate
│└:TCPip
│ └:CONTrol?
├:DCINterface
│├[:PROGram]
││├:LOAD
││├:UNLoad
││└:SELected?
│├:EXECute[:INIT]
│├:DEVice
││├:CONNect[?]
││└:MODE[?]
│└:LANE
│ └:MODE[?]
├:RESet
│├[:ALL]
│└:INSTrument
├:MODule
│└:UPDate
│ ├:APPLy
│ ├[:AUTOmatic][?]
│ └:PENDing[?]
└:SERVice
 └:MANagement
  └:CALibration
   └...
```

This subsystem has the following commands and a subnode:

Table 91

Name	Description under
:INSTrument:RESet	:SYSTem:INSTrument:RESet on page 518
:INSTrument:GROup:DEFine	:SYSTem:INSTrument:GROup:DEFine on page 518
:INSTrument:GROup:DELete	:SYSTem:INSTrument:GROup:DELete on page 518
:INSTrument:GROup:MEMBer?	:SYSTem:INSTrument:GROup:MEMBer? on page 519
:INSTrument:GROup[:LIST]?	:SYSTem:INSTrument:GROup[:LIST]? on page 519
:INSTrument:LTRaining[:LOG]?	:SYSTem:INSTrument:LTRaining[:LOG]? on page 519
:INSTrument:LTRaining[:LOG]:CLEar	:SYSTem:INSTrument:LTRaining[:LOG]:CLEar on page 521
:INSTrument:LTRaining[:LOG]:BLOCk?	:SYSTem:INSTrument:LTRaining[:LOG]:BLOCk? on page 521
:INSTrument:LTRaining:EQUalization[:STATus]?	:SYSTem:INSTrument:LTRaining:EQUalization[:STATus]? on page 521
:INFormation[:INSTrument]:IDENtifier?	:SYSTem:INFormation[:INSTrument]:IDENtifier? on page 522
:INFormation[:INSTrument]:LOCations?	:SYSTem:INFormation[:INSTrument]:LOCations? on page 522
:INFormation[:INSTrument]:GROups:USER?	:SYSTem:INFormation[:INSTrument]:GROups:USER? on page 523
:INFormation[:INSTrument]:DETails:JSON?	:SYSTem:INFormation[:INSTrument]:DETails:JSON? on page 523
:INFormation[:INSTrument]:DETails:VALue?	:SYSTem:INFormation[:INSTrument]:DETails[:VALue]? on page 524
:INFormation[:INSTrument]:DETails:XML?	:SYSTem:INFormation[:INSTrument]:DETails:XML? on page 524
:INFormation:LICenses:HOST[:LIST]?	:SYSTem:INFormation:LICenses:HOST[:LIST]? on page 525
:INFormation:LICenses:MODules[:LIST]?	:SYSTem:INFormation:LICenses:MODules[:LIST]? on page 525
:COMMunicate:TCPip:CONTrol?	:SYSTem:COMMunicate:TCPip:CONTrol? on page 525
:DCINterface[:PROGram]:LOAD	:SYSTem:DCINterface[:PROGram]:LOAD on page 526
:DCINterface[:PROGram]:UNLoad	:SYSTem:DCINterface[:PROGram]:UNLoad on page 526
:DCINterface[:PROGram]:SELected?	:SYSTem:DCINterface[:PROGram]:SELected? on page 527
:DCINterface:EXEcute[:INIT]	:SYSTem:DCINterface:EXEcute[:INIT] on page 527

Name	Description under
:DCINterface:DEVice:CONNect	:SYSTem:DCINterface:DEVice:CONNect[?] on page 527
:DCINterface:DEVice:MODE	:SYSTem:DCINterface:DEVice:MODE[?] on page 528
:DCINterface:LANE:MODE	:SYSTem:DCINterface:LANE:MODE[?] on page 529
:RESet[:ALL]	:SYSTem:RESet[:ALL] on page 529
:RESet:INSTrument	:SYSTem:RESet:INSTrument on page 530
:SYSTem:MODule:UPDate:APPLy	:SYSTem:MODule:UPDate:APPLy on page 530
:SYSTem:MODule:UPDate[:AUTOmatic][?]	:SYSTem:MODule:UPDate[:AUTOmatic][?] on page 530
:SYSTem:MODule:UPDate:PENDing[?]	:SYSTem:MODule:UPDate:PENDing[?] on page 531
SYSTem:TIME:STARtup?	:SYSTem:TIME:STARtup? on page 531
Subnode	
:SYSTem:SERVice:MANagement:CALibration Subnode	:SYSTem:SERVice:MANagement:CALibration Subnode on page 532

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 92

License	Description	Modules
M8009A-062	Clock Generator two channel 60 GHz	M8009A
M8009A-0G6	Reference Clock Multiplier	M8009A
M8041A-0A2	Second Channel for Analyzer	M8041A
M8041A-0G2	Second Channel for Generator	M8041A
M8041A-0S1	Interactive Link Training for PCI Express, Clockgroup-wide License	M8041A
M8041A-0S2	SER/FER Analysis for Coded and Retimed Loopback, Clockgroup-wide License	M8041A
M8041A-C08	BERT one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8042A-G32	32 GBd PG for NRZ and PAM-4	M8042A
M8043A-0A3	Equalization and De-Embedding capability	M8043A
M8043A-A32	Error analysis up to 32.4 GBd for NRZ and PAM4	M8043A
M8045A-G32	32 GBd Pattern Generator	M8045A
M8046A-0A3	Equalizer	M8046A
M8046A-0S1	Interactive Link Training for PCIe 8/16/32 GT/s	M8046A
M8046A-0S2	SKP OS Filtering for PCIe 8/16/32 GT/s and CCIX 20/25 Gb/s	M8046A
M8046A-A32	32GBd Error Detector	M8046A
M8062A-C32	32 Gb/s BERT front-end	M8062A
M8062A-G32	32 Gb/s Pattern generator front-end	M8062A
M8070ADVB	Advanced Measurement Package for M8000 Series BERT Test Solutions	M8046A, M8043A, M8041A

: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[:INSTrument]:DETails subnode

The following commands provide information about a specific part of the instrument.

The returned information contains

- Identifier
- Product number
- Connection status (Online/Offline)
- List of all options
- User friendly name of the instrument
- Serial number
- VISA resource string
- Software Version
- All the above information for sub-components like remote heads, or external clock recovery modules

:SYSTem:INFormation[:INSTrument]:DETails:JSON?

Syntax :SYSTem:INFormation[:INSTrument]:DETails:JSON? 'Identifier'

Description This command returns the instrument details in .JSON (JavaScript Object Notation) data structure.

Input Parameters Identifier - Any module name. For example; 'M1', 'M2'

Return Value Returns the sequence of following values:
Identifier, ProductNumber, Connection, Options, FriendlyName of module, SerialNumber, VisaResource, SoftwareVersion, Subcomponents

Example :SYST:INF:INST:DET:JSON? 'M2'
{"Identifier":"M2","ProductNumber":"M8046A","Connection":"Offline","Opti
ons":["A32","A64","0P3","0P6","0A9","U64","UP3","UP6","UA9"],"FriendlyN

ame":"M8046A 64GBd Error Analyzer","SerialNumber":"DE5250000003","VisaResource":"OFFLINE::211::INSTR","SoftwareVersion":"5.0.125.1","Subcomponents":[]}

:SYSTem:INFormation[:INSTrument]:DETails[:VALue]?

Syntax	:SYSTem:INFormation[:INSTrument]:DETails[:VALue]? 'Identifier'
Description	This command returns the information as SCPI expression. Individual fields are separated by comma, and sub-structures are enclosed in round brackets.
Input Parameters	Identifier - Any module name. For example; 'M1', 'M2'
Return Value	Returns the instrument details value in the following sequence: (identifier, product-number, connection-status, (installed-options), friendly-name, serial-number, VISA-resource, software-version, ((sub-component-info-1), (sub-component-info-2), ... (sub-component-info-n))
Example	:SYST:INF:INST:DET:VAL? 'M2' ('M2','M8046A','Offline',(A32,A64,0P3,0P6,0A9,U64,UP3,UP6,UA9),'M8046A 64GBd Error Analyzer','DE5250000003','OFFLINE::211::INSTR', '5.0.125.1',())

:SYSTem:INFormation[:INSTrument]:DETails:XML?

Syntax	:SYSTem:INFormation[:INSTrument]:DETails:XML? 'Identifier'
Description	This command returns the instrument details in XML format.
Input Parameters	Identifier - Any module name. For example; 'M1', 'M2'
Return Value	Returns the following instrument details in XML format: Identifier, Product Number, Connection type, License options, friendly name of module, Serial Number, Visa resource, Software version, Subcomponents
Example	:SYST:INF:INST:DET:XML? 'M2' <?xml version="1.0" encoding="utf-16"?><DetailedComponentInfo xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Identifier>M2</Identifier><ProductNumber>M8046A</ProductNumber><Connection>Offline</Connection><Options>A32,A64,0P3,0P6,0A9,U64,UP3,UP6,UA

9</Options><FriendlyName>M8046A 64GBd Error Analyzer</FriendlyName><SerialNumber>DE5250000003</SerialNumber><VisaResource>OFFLINE::211::INSTR</VisaResource><SoftwareVersion>5.0.125.1</SoftwareVersion><Subcomponents /></DetailedComponentInfo>

: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?

((PCSERNO,FH34484899,("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","")("M8070ADEM"," DEM","1.0","Demo license for M8000 Series, demo-use only, transportable perpetual license","Installed","Perpetual","12/31/9999 11:59:59 PM","")("M8070ADVB","","","Advanced Measurement Package for M8000 Series BERT Test Solutions","Not installed","","12/31/9999 11:59:59 PM","")("M8070EDAB","","","Error Distribution Analysis Package for M8000 Series BERT Test Solutions","Not installed","","12/31/9999 11:59:59 PM","")))

: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 M8070B 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

:SYSTem:MODule:UPDate:APPLy

Syntax :SYSTem:MODule:UPDate:APPLy "Location"

Input Parameters <identifier> - "Location" , e.g. "M1.System"

Description This command applies the currently configured software state into the module.

This command is applicable for M8194A, M8195A, M8196A & M8054A.

Example :SYST:MOD:UPD:APPL "M1.System"

:SYSTem:MODule:UPDate[:AUTOmatic][?]

Syntax :SYSTem:MODule:UPDate[:AUTOmatic] "Location", <ON|OFF|1|0>

:SYSTem:MODule:UPDate[:AUTOmatic]? "Location"

Input Parameters <identifier> - "Location" , e.g. "M1.System"

<ON|OFF|1|0>

Description This command automatically applies the changes made to the instrument. When the Automatic Module Update option is disabled, the time intensive operation’s execution defers for parameter changes. Depending on the module type, this may disable some or all the property updates. Additionally, it can also disable the dynamic parameter limit calculations, which can trigger the “Auto Correct Confirmation” dialog when the module updates.

This query returns the current settings.

This command is applicable for M8194A, M8195A, M8196A, M8053A and M8054A.

Example :SYST:MOD:UPD:APPL "M1.System", 1

:SYSTem:MODule:UPDate:PENDing[?]

Syntax :SYSTem:MODule:UPDate:PENDing <identifier>, <ON|OFF|1|0>

:SYSTem:MODule:UPDate:PENDing? <identifier>

Input Parameters <identifier> - "Location" , e.g. "M1.System"

Description This applied the pending changes that are not yet sent to the hardware or otherwise not updated.

This query returns the current settings.

This command is applicable for M8194A, M8195A, M8196A, M8053A, M8154A.

Example :SYST:MOD:UPD:APPL "M1.System", 1

:SYSTem:TIME:STARtup?

Syntax :SYSTem:TIME:STARtup?

Input Parameters None

Description This query returns the date and time stamp when the current instance of the M8070B software was launched.

Example -> :SYSTem:TIME:STARtup?

<- (2023-02-24 10:15:32)

:SYSTem:SERVice:MANagement:CALibration Subnode

This subnode allows you to set the calibration interval, configure calibration due reminders, check the calibration status, set/update passcode and set reminder for a calibration service.

This subnode has the following SCPI structure:

```
:SYSTem
└:SERVice
  └:MANagement
    └:CALibration
      ├:LIST?
      ├:SELect[?]
      ├:INFormation?
      ├:INTerval
      │ ├:DEFault
      │ ├:VALue[?]
      │ └:TYPE?
      ├:NOTification
      │ └:ENABle[?]
      ├:PASScode
      │ ├:CHANge
      │ └[:VALue]
      ├:PERiodic
      │ └:ENABle[?]
      └:REMinder[?]
```

This subnode has the following commands:

Table 93

Name	Description under
::SYSTem:SERVice:MANagement:CALibration:LIST?	:SYSTem:SERVice:MANagement:CALibration:LIST? on page 534
:SYSTem:SERVice:MANagement:CALibration:SELect[?]	:SYSTem:SERVice:MANagement:CALibration:SELect[?] on page 534
:SYSTem:SERVice:MANagement:CALibration:INFormation?	:SYSTem:SERVice:MANagement:CALibration:INFormation? on page 534
:SYSTem:SERVice:MANagement:CALibration:INTerval:DEFault	:SYSTem:SERVice:MANagement:CALibration:INTerval:DEFault on page 535
:SYSTem:SERVice:MANagement:CALibration:INTerval:VALue[?]	:SYSTem:SERVice:MANagement:CALibration:INTerval:VALue[?] on page 536
:SYSTem:SERVice:MANagement:CALibration:INTerval:TYPE?	:SYSTem:SERVice:MANagement:CALibration:INTerval:TYPE? on page 536
:SYSTem:SERVice:MANagement:CALibration:NOTification:ENABle[?]	:SYSTem:SERVice:MANagement:CALibration:NOTification:ENABle[?] on page 537
:SYSTem:SERVice:MANagement:CALibration:PASScode:CHANge	:SYSTem:SERVice:MANagement:CALibration:PASScode:CHANge on page 537
:SYSTem:SERVice:MANagement:CALibration:PASScode[:VALue]	:SYSTem:SERVice:MANagement:CALibration:PASScode[:VALue] on page 537
SYSTem:SERVice:MANagement:CALibration:PERiodic:ENABle[?]	:SYSTem:SERVice:MANagement:CALibration:PERiodic:ENABle[?] on page 538
SYSTem:SERVice:MANagement:CALibration:REMinder[?]	:SYSTem:SERVice:MANagement:CALibration:REMinder[?] on page 538

:SYSTem:SERVice:MANagement:CALibration:LIST?

Syntax	:SYSTem:SERVice:MANagement:CALibration:LIST?
Input Parameters	None
Description	This query returns the model(module) list in PathWave Calibration Advisor user interface.
Return Type	String format including all listed models
Example	:SYST:SERV:MAN:CAL:LIST? returns this format: M8057A_MY00000001, M8041A_MY00000002

:SYSTem:SERVice:MANagement:CALibration:SELect[?]

Syntax	:SYSTem:SERVice:MANagement:CALibration:SELect <value> :SYSTem:SERVice:MANagement:CALibration:SELect?
Input Parameters	<value> - “<Model>_<Serial>” is a quoted string
Description	This command sets the selected model(module) information. NOTE: In a multiple module system, send this command before querying more info. This query returns the selected model(module) information.
Return Type	String
Example	:SYST:SERV:MAN:CAL:SEL “M8057A_MY00000001”

:SYSTem:SERVice:MANagement:CALibration:INFormation?

Syntax: :SYSTem:SERVice:MANagement:CALibration:INFormation?

Input Parameters: None

Description: This command the instrument calibration information in JSON format. The returned value includes the following:

- CalID – Calibration ID number created by the service center when the instrument was calibrated.
- CalBy – Service center that calibrated this instrument – either "Keysight Technologies" or "Keysight Authorized Calibration Provider".

- CalDate – Date of last calibration.
- CalDueDate – Calibration due date. The due date is calculated as CalDate + Calibration Interval, unless a shelf life policy is applied.
- Status – returned value depends on:
  - If SystemTime≤CalDueDate, Status is "CalibrationValid".
  - If SystemTime>CalDueDate, Status is "CalibrationRequired".
  - If CalDueDate is empty, and CalDate is presented, Status is "CalibrationValid".
  - If SystemTime<CalDate, Status is "CalibrationUnknown".
  - If Cal Data is unavailable, Status is "CalibrationNotFound".
- SystemTime – Current system time.

Return Type: String of JSON format including all calibration info

Example: :SYST:SERV:MAN:CAL:INF?

returns this format:

{"CalId":"1-00000000000-1","CalBy":"Keysight Technonlogies","CalDate":"27-May-2020","CalDueDate":"27-May-2021","Status":"CalibrationValid","SystemTime":"2020- 11-18 10:38:08"}

:SYSTem:SERVice:MANagement:CALibration:INTerval:DEFault

Syntax: :SYSTem:SERVice:MANagement:CALibration:INTerval:DEFault

Input Parameters: None

Description: This command sets calibration interval to Keysight Recommended (according to the instrument model).

Example: :SYST:SERV:MAN:CAL:INT:DEF

:SYSTem:SERVice:MANagement:CALibration:INTerval:VALue[?]

Syntax :SYSTem:SERVice:MANagement:CALibration:INTerval:VALue <value>

:SYSTem:SERVice:MANagement:CALibration:INTerval:VALue?

Input Parameters <value> - Calibration interval in months. An integer between 1 and 120.

Description This command sets the calibration interval value (interval in months).

NOTE:

- If using this command to set to the calibration interval, the calibration interval type will be switched to user defined (CUSTomer).
- This command only returns the value of current saved calibration interval type. Any unsaved changes will not be returned.

This query returns the calibration interval value (interval in months).

Example :SYST:SERV:MAN:CAL:INT:VAL 12

:SYSTem:SERVice:MANagement:CALibration:INTerval:TYPE?

Syntax :SYSTem:SERVice:MANagement:CALibration:INTerval:TYPE?

Input Parameters None

Description This query returns the instrument calibration interval type. Three types may return:

- DEFault: When current calibration interval is "Keysight recommended", the returned value is DEFault.
- CUSTomer: When the current calibration interval is "user defined", the returned value is CUSTomer.
- EMPTY: After new calibration info is imported and the calibration interval is not confirmed, the returned value is empty.

Example :SYST:SERV:MAN:CAL:INT:TYPE?

returns this format:

“CUSTomer” or “DEFault"

:SYSTem:SERVice:MANagement:CALibration:NOTification:ENABle[?]

Syntax	:SYSTem:SERVice:MANagement:CALibration:NOTification:ENABle <bool> :SYSTem:SERVice:MANagement:CALibration:NOTification:ENABle?
Input Parameters	<bool> - OFF (or 0) Popup notification is NOT visible during instrument running. ON (or 1) Popup notification is visible during instrument running.
Description	This command sets the calibration status/due popup notification state during instrument running. If disabled, the notification will pop up only on power up. If enabled, the notification will also pop up at 12:00 AM every day when the current date is ≤ the Cal Due Reminder days set by SYST:SERV:MAN:CAL:REM. This query returns the calibration status/due popup notification.
Example	:SYST:SERV:MAN:CAL:NOT:ENAB 0 :SYST:SERV:MAN:CAL:NOT:ENAB OFF

:SYSTem:SERVice:MANagement:CALibration:PASScode:CHANge

Syntax	:SYSTem:SERVice:MANagement:CALibration:PASScode:CHANge <value>
Input Parameters	<value> - Password
Description	This command sets new passcode. The character requirements: 6-20 characters, limited to alpha (a-Z) and numeric (0-9) characters.
Example	:SYST:SERV:MAN:CAL:PASS:CHAN ‘1q2w3e4r’

:SYSTem:SERVice:MANagement:CALibration:PASScode[:VALue]

Syntax	:SYSTem:SERVice:MANagement:CALibration:PASScode[:VALue] <value>
Input Parameters	<value> - Latest saved password
Description	This command inputs existing passcode and enable all setting features.
Example	:SYST:SERV:MAN:CAL:PASS ‘1q2w3e4r’

:SYSTem:SERVice:MANagement:CALibration:PERiodic:ENABle[?]

Syntax	:SYSTem:SERVice:MANagement:CALibration:PERiodic:ENABle <bool>
	:SYSTem:SERVice:MANagement:CALibration:PERiodic:ENABle?
Input Parameters	<bool> - OFF (or 0) Periodic calibration is disable.
	ON (or 1) Periodic calibration is enable.
Description	This command sets the periodic calibration state.
	This query returns the periodic calibration state.
Example	:SYST:SERV:MAN:CAL:PER:ENAB 0
	:SYST:SERV:MAN:CAL:PER:ENAB OFF

:SYSTem:SERVice:MANagement:CALibration:REMinder[?]

Syntax	:SYSTem:SERVice:MANagement:CALibration:REMinder <value>
	:SYSTem:SERVice:MANagement:CALibration:REMinder?
Input Parameters	<value> - Calibration due reminder in days. Only one number of “30, 15, 7”.
Description	This command sets the calibration due reminder days.
	This query returns the calibration due reminder days.
Example	:SYST:SERV:MAN:CAL:REM 30

:SYSTem:SERVice:MANagement:CALibration:REMinder[?]

Syntax	:SYSTem:SERVice:MANagement:CALibration:REMinder <value>
	:SYSTem:SERVice:MANagement:CALibration:REMinder?
Input Parameters	<value> - Calibration due reminder in days. Only one number of “30, 15, 7”.
Description	This command sets the calibration due reminder days.
	This query returns the calibration due reminder days.
Example	:SYST:SERV:MAN:CAL:REM 30

Error Messages

Error #	Error String	Description of special cases
-200	Execution error	When KAC Service is still running, but none of above SCPI commands will have response
-221	Settings conflict	**1** Send wrong passcode to instrument **2** Change settings without right passcode sent to instrument firstly
-224	Illegal parameter value	**1** Try to set reminder days to one number which is not integer number 7 or 15 or 30 **2** Try to set calibration interval to one number which is not an integer number between 1 and 120 **3** Try to set new passcode whose length is less than six letters
-310	System error	PathWave Calibration Advisor Service is not running

Example

Example	Description	Example	Comments
1	**1** Change any existing settings, passcode is needed and must be sent firstly. **2** Right passcode is needed only once until instrument power off	*CLS	Clean all existing errors
		:SYST:SERV:MAN:CAL:PASS 'Key4Cal'	Send passcode (Key4Cal is default passcode, it could be changed)
		:SYST:ERR?	Query system errors
		<- 0	If passcode is wrong, it will return -221
2	Change calibration interval	:SYST:SERV:MAN:CAL:PASS 'Key4Cal'	Send right passcode to enable setting commands
		:SYST:SERV:MAN:CAL:INT:VAL?	
		<- 6	Query current calibration interval value
		:SYST:SERV:MAN:CAL:INT:VAL 12	Set new calibration interval value
		:SYST:SERV:MAN:CAL:INT:VAL?	Confirm new calibration interval value
		<- 12	
		:SYST:SERV:MAN:CAL:PER:ENAB?	Confirm subject to periodic cal is enabled
		<-ON	

Example	Description	Example	Comments
3	Change calibration interval to Keysight recommended	:SYST:SERV:MAN:CAL:PASS 'Key4Cal'	Send right passcode to enable setting commands
		:SYST:SERV:MAN:CAL:INT:TYPE?	Query current calibration interval type
		<- CUSTomer	
		:SYST:SERV:MAN:CAL:INT:DEF	Set calibration interval to Keysight recommended
		:SYST:SERV:MAN:CAL:INT:TYPE?	Confirm new calibration interval type
		<- DEFault	
		:SYST:SERV:MAN:CAL:INT:VAL?	Confirm new calibration interval value
		<- 12	
		->:SYST:SERV:MAN:CAL:PER:ENAB?	Confirm subject to periodic cal is enabled
		<-ON	
4	Change calibration due reminder	:SYST:SERV:MAN:CAL:PASS 'Key4Cal'	Send right passcode to enable setting commands
		:SYST:SERV:MAN:CAL:REM?	Query current calibration due reminder value
		<- 30	
		:SYST:SERV:MAN:CAL:REM 15	Set new calibration due reminder
		:SYST:SERV:MAN:CAL:REM?	Confirm new calibration due reminder value
		<- 15	
		:SYST:SERV:MAN:CAL:NOT:ENAB?	Confirm popup notification is enable
		<-ON	
		:SYST:SERV:MAN:CAL:PER:ENAB?	Confirm subject to periodic cal is enabled
		<-ON	
5	Change Passcode	*CLS	Clean all existing errors
		:SYST:SERV:MAN:CAL:PASS 'Key4Cal'	Send right passcode to enable setting commands
		:SYST:SERV:MAN:CAL:PASS:CHAN 'newpasscode'	Set new passcode
		:SYST:ERR?	Query system errors

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
│ └...
├:LINecoding
│ └...
├:FEC
│ └...
├:COMPare
│ └...
└:IDLepattern:VALue[?]
```

The subsystem has the following subnodes and a command:

Table 94

Name	Description under
:PATTern	:DATA:PATTern Subnode on page 543
:SEQuence	:DATA:SEQuence Subnode on page 552
:SYNC	:DATA:SYNC Subnode on page 561
:LINecoding	:DATA:LINecoding Subnode on page 563
:FEC	:DATA:FEC Subnode on page 578
:COMPare	:DATA:COMPare Subnode on page 583
:IDLepattern:VALue[?]	:DATA:IDLepattern:VALue[?] on page 585

The SCPIs listed under this subsystem might require licenses for their operation. The following table list the licenses which you must install before executing these SCPIs.

Table 95

License	Description	Modules
M8009A-062	Clock Generator two channel 60 GHz	M8009A
M8009A-0G6	Reference Clock Multiplier	M8009A
M8041A-0A2	Second Channel for Analyzer	M8041A
M8041A-0G2	Second Channel for Generator	M8041A
M8041A-C08	BERT one Channel, Data Rate up to 8.5 Gb/s	M8041A
M8042A-G32	32 GBd PG for NRZ and PAM-4	M8042A
M8043A-A32	Error analysis up to 32.4 GBd for NRZ and PAM4	M8043A
M8045A-0G9	FEC encoding, module-wide license	M8045A
M8045A-0P3	PAM4 encoding up to 32 GBd, module-wide license	M8045A
M8045A-G32	32 GBd Pattern Generator	M8045A
M8046A-A32	32GBd Error Detector	M8046A
M8046A-A32	Analyzer, one channel, data rate up to 32 GBd, NRZ	M8046A
M8062A-C32	32 Gb/s BERT front-end	M8062A
M8062A-G32	32 Gb/s Pattern generator front-end	M8062A

:DATA:PATTern Subnode

For additional information about patterns, refer to Working with Patterns on page 46.

This subnode has the following SCPI structure:

```
:DATA
└:PATTern
  ├:USE[?]
  ├:IDATa[?]
  ├:BLENgth?
  ├:CATalog?
  ├:CINFormation?
  ├:DESCription[?]
  ├:DELete
  ├:PACKed[?]
  ├:IMPort
  ├:EXPort
  ├:ROTate:LEFT
  ├:ROTate:RIGHT
  └:MAPped:SAVE
```

This subnode has the following commands:

Table 96

Name	Description under
:USE[?]	:DATA:PATTern:USE[?] on page 544
:IDATa[?]	:DATA:PATTern:IDATa[?] on page 545
:BLENgth?	:DATA:PATTern:BLENgth? on page 546
:CATalog?	:DATA:PATTern:CATalog? on page 546
:CINFormation?	:DATA:PATTern:CINFormation? on page 546
:DESCription[?]	:DATA:PATTern:DESCription[?] on page 547
:DELete	:DATA:PATTern:DELete on page 547
:PACKed[?]	:DATA:PATTern:PACKed[?] on page 547
:IMPort	:DATA:PATTern:IMPort on page 548

Name	Description under
:EXPort	:DATA:PATTern:EXPort on page 549
:DATA:PATTern:ROTate:LEFT	:DATA:PATTern:ROTate:LEFT on page 549
:DATA:PATTern:ROTate:RIGHT	:DATA:PATTern:ROTate:RIGHT on page 550
:DATA:PATTern:MAPped:SAVE	:DATA:PATTern:MAPped:SAVE on page 550

: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>

Input Parameters <pattern-name>: Specify the pattern name.

<file-name>: Specify the location of the pattern file to be imported.

Description This command imports an N4903B J-BERT pattern or M8020A pattern file.

M8020A 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. For details on packing and its parameters, refer to :DATA:PATTern:PACKed[?]

on page 547.

The <data> in a separate line contains the sequence of bytes as specified by the packing. For more details on how to edit data, refer to the chapter “Setting up Patterns” of the *M8000 Series User Guide* (https://www.keysight.com/in/en/assets/9018-04784/user-manuals/9018-04784.pdf?id=3021945).

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>

Input Parameters <pattern-name>: Specify the pattern name.

<file-name>: Specify the location of the pattern file to be exported.

Description This command exports a pattern to a file in the M8020A import file format.

Example :DATA:PATT:EXP "current:x","C:/Users/jsmith/Desktop/test"

:DATA:PATTern:ROTate:LEFT

Syntax :DATA:PATTern:ROTate:LEFT <input pattern file name>, <output pattern file name>, <rotation length>

Input Parameters <input pattern file name>, <output pattern file name>, <rotation length>

Description This command shifts the amount of symbols that need to be shifted to the left.

The input file can be stored in any of the workspace locations, namely Factory, Current, or Shared. The output file can only be saved in Current or Shared. If the output file already exists, it will be overwritten; however, if it doesn't exist, then a new file with the provided name will be created. The <rotation length> parameter defines the number of symbols that need to be shifted. Therefore, the actual shift will be equal to the number specified in the <rotation length> parameter.

Example :DATA:PATTern:ROTate:LEFT "shared:PRBS7-1","shared:MappedPattern",2

:DATA:PATTern:ROTate:RIGHT

Syntax :DATA:PATTern:ROTate:RIGHT <input pattern file name>, <output pattern file name>, <rotation length>

Input Parameters <input pattern file name>, <output pattern file name>, <rotation length>

Description This command shifts the amount of symbols that need to be shifted to the right.

The input file can be stored in any of the workspace locations, namely Factory, Current, or Shared. The output file can only be saved in Current or Shared. If the output file already exists, it will be overwritten; however, if it doesn't exist, then a new file with the provided name will be created. The <rotation length> parameter defines the number of symbols that need to be shifted. Therefore, the actual shift will be equal to the number specified in the <rotation length> parameter.

Example :DATA:PATTern:ROTate:RIGHT "shared:PRBS7-1","shared:MappedPattern",2

:DATA:PATTern:MAPped:SAVE

Syntax :DATA:PATTern:MAPped:SAVE "<Source Pattern File>","<Destination Pattern File>","<Custom Mapping File>"

Input Parameters <Source Pattern File> - Name and path of the source pattern file

<Destination Pattern File> - Name and path of the destination pattern file

<Custom Mapping File> - Name and path of the custom mapping file

Description The command maps the source pattern file according to the mapping table defined in the custom mapping file and saves the result to the destination pattern file.

This SCPI is applicable for M8045A.

Example :DATA:PATT:MAP:SAVE "Factory:GDDR7/PRBS7-1_PAM3_7F_shift0", "Shared:PRBS7-1_PAM3_7F_shift0","Factory/Custom3Bit2SymbolMappingPam3.xml"

:DATA:SEQuence Subnode

With this subsystem sequencing can be changed programmatically. For additional information, refer to Creating Pattern Sequences on page 53.

This subnode has the following SCPI structure:

```
:DATA
└:SEQuence
  ├[:SET][?]
  ├[:SET]:VALue[?]
  ├:NEW
  ├:NAMes?
  ├:BIND[?]
  ├:DELete
  ├:DALL
  ├:BREak
  ├:RESTart
  │ └:IDENtifier
  ├:BLOCk?
  │ └:IDENtifier?
  └:GRANularity?
    └:IDENtifier?
```

This subnode has the following commands:

Table 97

Name	Description under
[:SET][?]	:DATA:SEQuence[:SET][?] on page 553
[:SET]:VALue[?]	:DATA:SEQuence[:SET]:VALue[?] on page 555
:NEW	:DATA:SEQuence:NEW on page 556
:NAMes?	:DATA:SEQuence:NAMes? on page 556
:BIND[?]	:DATA:SEQuence:BIND[?] on page 556
:DELete	:DATA:SEQuence:DELete on page 557
:DALL	:DATA:SEQuence:DALL on page 557
BREak	:DATA:SEQuence:BREak on page 557

Name	Description under
:RESTart	:DATA:SEQuence:RESTart on page 558
:RESTart:IDENtifier	:DATA:SEQuence:RESTart:IDENtifier on page 558
:BLOCk?	:DATA:SEQuence:BLOCk? on page 559
:BLOCk:IDENtifier?	:DATA:SEQuence:BLOCk:IDENtifier? on page 559
:GRANularity?	:DATA:SEQuence:GRANularity? on page 559
:GRANularity:IDENtifier?	:DATA:SEQuence:GRANularity:IDENtifier? on page 560

:DATA:SEQuence[:SET][?]

Syntax :DATA:SEQuence[:SET] <sequence-name | location | group-name>, <pattern type>,<pattern-value>

:DATA:SEQuence[:SET]? <sequence-name | location>

Input Parameters <sequence-name | location | group-name>,<pattern type>, <pattern-value>

<sequence-name | location | group-name> is a string name for which possible values are:

- <sequence-name> Name of the existing sequence, either default sequence name or user created sequence name.
- <location> Individual Analyzer or Generator location e.g. "M1.DataIn1"
- <group-name> A user created group name containing Analyzer(s) or Generator(s) locations. In this case the group name must only contain numbers and letters.

<pattern-type>

- PRBS: Sets a PRBS of specified length.
- PULSE: Sets a pulse of specified width.
- CLOCK: Sets a clock of specified divider.
- STATIC: Sets a static signal of specified value.
- MEMORY: Sets a memory pattern saved in the setting.
- SQUELCH: Selects a squelch from the Block Data.

<pattern-value>: String value parameter according to the pattern type.

- For PRBS possible values are: "2^7-1", "2^9-1", "2^10-1", "2^11-1", "2^13-1", "2^15-1", "2^23-1", "2^23-1p" or "2^23-1 PCIe", "2^31-1", "2^33-1", "2^35-1", "2^39-1", "2^41-1", "2^45-1", "2^47-1", "2^49-1", "2^51-1", "3^17", "3^19" and "3^23".
- For PULSE it's a specified width value e.g. "32"
- For CLOCK it's a specified divider value e.g. "2"
- For STATIC it's a signal value e.g. "0", "1", "low", "high"
- For MEMORY it's a name of the pattern. Name consists of root node ("factory:", "shared:" or "current:") plus folder names separated by '/' plus pattern name. e.g. "factory:CEI/CEIstress_bit"

Description

This command allows you to:

- set sequence for a specified location with the specified pattern.
- set or change a pattern for single location, sequence or multiple locations (exiting user group).
- create/modify a sequence for the given location with a single block pattern in an infinite loop.

For more verbose setting use, :DATA:SEQuence[:SET]:VALue command providing XML value.

This query returns the following:

- current pattern set for the given existing sequence or Analyzer/Generator location.
- pattern and value if the current sequence for the given location contains single data block pattern.

If there are more the one block or block targets are used in the sequence then use, :Data:SEQ:SET:VALue? query to retrieve xml sequence setting value.

Example

The following are few command examples:

:DATA:SEQ:SET "Generator",PRBS,"2^9-1" sets a pattern for given "Generator" sequence.

:DATA:SEQ:SET "M1.DataIn1",PULSE,"32" sets a pattern for given "M1.DataIn1" analyzer location only.

:DATA:SEQ:SET "M2.DataOut1", CLOCK,"6" sets a pattern for given "M1.DataOut1" generator location only.

:DATA:SEQ:SET "M2.DataOut1", SQUELCH sets a squelch block type.

:DATA:SEQ:SET "Analyzer", STATIC,"1" sets a pattern for given "Analyzer" sequence.

:DATA:SEQ:SET "M1.DataIn1",MEMORY,"factory:CEI/CEIstress_bit" sets a pattern for given "M1.DataIn1" analyzer location only.

The following are few query examples:

:DATA:SEQ:SET? "Generator"

> "Generator",PRBS,"2^9-1"

:DATA:SEQ:SET? "M1.DataIn1"

> "M1.DataIn1",PULSE,"32"

:DATA:SEQ:SET? "M1.DataOut1"

>"M2.DataOut1", CLOCK,"6"

:DATA:SEQ:SET? "Analyzer"

> "Analyzer", STATIC,"1"

:DATA:SEQ:SET? "M1.DataIn1"

> "M1.DataIn1",MEMORY,"factory:CEI/CEIstress_bit"

:DATA:SEQuence[:SET]:VALue[?]

Syntax	:DATA:SEQuence[:SET]:VALue <sequence-name>,<sequence-string>
	:DATA:SEQuence[:SET]: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.keysight.com/schemas/M8000/DataSequence"><description /><sequence><loop><block length="128"><prbs polynomial="2^7-1" /></block></loop></sequence></sequenceDefinition>

: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 applicable for M8041A, M8051A, M8194A, M8195A, M8196A M8043A and M8046A. However, for M8194A, M8195A and M8196A, it is only valid for Data Out locations.

Example :DATA:SEQ:BIND 'MySequence','M1.DataIn1'

: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'

For M8020A & M8040A, the identifier would be 'M1.System' as Pattern Generator module occupies the primary slot. However, for M8050A, the identifier would be 'M2.System' as the primary slot is occupied by the Clock module and the Pattern Generator occupies the secondary slot.

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

For M8020A & M8040A:

:DATA:SEQ:BRE 'M1.System'

For M8050A:

:DATA:SEQ:BRE 'M2.System'

:DATA:SEQuence:RESTart

Syntax

:DATA:SEQuence:RESTart [<sequence-name>[,<sequence-block-id>]]

Input Parameters

<sequence-name>: Specify sequence name.

<sequence-block-id>: Specify sequence block ID.

Description

This command restarts all sequences if no argument is specified.

This command restarts a specified sequence if <sequence-name> is specified.

This command jumps to a specific sequence block if <sequence-name> and <sequence-block-id> is specified.

- Limitations on sequence block ID selection:

  In case the sequencer block ID selected is contained inside an outer loop and it is not the starting segment the outer loop will not be executed.

NOTE Analyzer(s) are not able to do a break. So using this command, you can branch to any other block, also on analyzers.

Example

:DATA:SEQ:REST

:DATA:SEQ:REST 'Analyzer'

:DATA:SEQ:REST 'Generator', 3

: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:SEQuence:GRANularity?

Syntax :DATA:SEQuence:GRANularity? <sequence-name>

Input Parameters <sequence-name>: Specify sequence name, i.e., 'Sequence1'

Description This query returns the granularity in bits of sequence. It changes with the symbol width. The following table shows the granularity for various symbol widths for different modules:

Symbol Width	Granularity (M8041A, M8051A, M8194A, M8195A, M8196A)	Granularity (M8062A)	Granularity (M8045A, M8046A)
1	64	128	512
10	80	160	Not supported
130	130	260	Not supported
132	132	264	Not supported

Example :DATA:SEQ:GRAN? 'Sequence1'

:DATA:SEQuence:GRANularity:IDENtifier?

Syntax :DATA:SEQuence:GRANularity:IDENtifier? <identifier>

Input Parameters <identifier>: 'M*.DataIn1', 'M*.DataIn2', 'M*.DataIn', 'M*.DataOut', 'M*.DataOut1' and 'M*.DataOut2'

Description This query returns the current granularity in bits of location or group. It changes with the symbol width. Refer to the table shown in the command :DATA:SEQuence:GRANularity:IDENtifier? on page 560.

Example :DATA:SEQ:GRAN: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 98

Name	Description under
[:ONCE]	:DATA:SYNC[:ONCE] on page 561
:TYPE[?]	:DATA:SYNC:TYPE[?] on page 562
:THReshold[?]	:DATA:SYNC:THReshold[?] on page 562

:DATA:SYNC[:ONCE]

Syntax :DATA:SYNC[:ONCE] <identifier>

Input Parameters <identifier>: 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description This command initiates a pattern resynchronization.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :DATA:SYNC 'M1.DataIn1'

:DATA:SYNC:TYPE[?]

Syntax	:DATA:SYNC:TYPE 'identifier', AUTO|MANual
	:DATA:SYNC:TYPE? 'identifier'
Input Parameters	<identifier>: 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'
	AUTO|MANual: Enable/disable automatic resynchronization.
Return Range	AUTO|MAN
Description	This command enables/disables automatic resynchronization.
	This SCPI is applicable for M8041A, M8051A, M8062A, M8046A and M8043A.
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', 'M*.DataIn2' or 'M*DataIn', 'INF*.DataIn1'
	<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 applicable for M8041A, M8051A, M8062A, M8043A, M8046A and real-time oscilloscope based error detector channels.
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][?]
  ├─ :PAM3
  │  ├─:PRECoding[?]
  │  │ ├─:VALue[?]
  │  │ └─:FEEDback[?]
  │  ├─[:SYMBol]
  │  │ └─:LEVel(1)[?]
  │  ├─:MAPPing[?]
  │  ├─:MAPPing:CUSTom[?]
  │  └─:SYMBol
  │    ├─:ENABle[?]
  │    └─:COUNt[?]
  ├─ :PAM4
  │  ├─:MAPPing[?]
  │  ├─:MAPPing:CUSTom[?]
  │  ├─[:SYMBol]:LEVel(1|2)[?]
  │  └─:PRECoding[?]
  │    └─:VALue[?]
  ├─ :PAM6
  │  ├─:MAPPing[?]
  │  ├─:MAPPing:CUSTom:FILE[?]
  │  ├─:MAPPing:CUSTom:FILE:STATe[?]
  │  └─:[:SYMBol]:LEVel(1|2|3|4)[?]
  ├─ :PAM8
  │  ├─:MAPPing[?]
  │  ├─:MAPPing:CUSTom:FILE[?]
  │  ├─:MAPPing:CUSTom:FILE:STATe[?]
  │  └─:[:SYMBol]:LEVel(1|2|3|4|5|6)[?]
  └─ :INVerse:PREcoding[?]
```

This subnode has the following commands:

Table 99

Name	Description under
:DATA:LINecoding[:VALue][?]	:DATA:LINecoding[:VALue][?] on page 565
:DATA:LINecoding:PAM3:PRECoding[?]	:DATA:LINecoding:PAM3:PRECoding[?] on page 566
:DATA:LINecoding:PAM3:PRECoding:VALue[?]	:DATA:LINecoding:PAM3:PRECoding:VALue[?] on page 566
:DATA:LINecoding:PAM3:PRECoding:FEEDback[?]	:DATA:LINecoding:PAM3:PRECoding:FEEDback[?] on page 566
:DATA:LINecoding:PAM3[:SYMBol]:LEVel(1)[?]	:DATA:LINecoding:PAM3[:SYMBol]:LEVel(1)[?] on page 567
:DATA:LINecoding:PAM3[:SYMBol]:LEVel(1)[?]	:DATA:LINecoding:PAM3:MAPPing[?] on page 567
:DATA:LINecoding:PAM3:MAPPing:CUSTom[?]	:DATA:LINecoding:PAM3:MAPPing:CUSTom[?] on page 568
:DATA:LINecoding:PAM3:SYMBol:ENABle[?]	:DATA:LINecoding:PAM3:SYMBol:ENABle[?] on page 568
:DATA:LINecoding:PAM3:SYMBol:COUNt[?]	:DATA:LINecoding:PAM3:SYMBol:COUNt[?] on page 569
:DATA:LINecoding:PAM4:MAPPing[?]	:DATA:LINecoding:PAM4:MAPPing[?] on page 569
:DATA:LINecoding:PAM4:MAPPing:CUSTom[?]	:DATA:LINecoding:PAM4:MAPPing:CUSTom[?] on page 570
:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2)[?]	:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2)[?] on page 571
:DATA:LINecoding:PAM4:PRECoding[?]	:DATA:LINecoding:PAM4:PRECoding[?] on page 571
:DATA:LINecoding:PAM4:PRECoding:VALue[?]	:DATA:LINecoding:PAM4:PRECoding:VALue[?] on page 572
:DATA:LINecoding:PAM6:MAPPing[?]	:DATA:LINecoding:PAM6:MAPPing[?] on page 572
:DATA:LINecoding:PAM6:MAPPing:CUSTom:FILE[?]	:DATA:LINecoding:PAM6:MAPPing:CUSTom:FILE[?] on page 573
:DATA:LINecoding:PAM6:MAPPing:CUSTom:FILE:STATe[?]	:DATA:LINecoding:PAM6:MAPPing:CUSTom:FILE:STATe[?] on page 573
:DATA:LINecoding:PAM8:MAPPing[?]	:DATA:LINecoding:PAM8:MAPPing[?] on page 574
:DATA:LINecoding:PAM8:MAPPing:CUSTom:FILE[?]	:DATA:LINecoding:PAM8:MAPPing:CUSTom:FILE[?] on page 574
:DATA:LINecoding:PAM8:MAPPing:CUSTom:FILE:STATe[?]	:DATA:LINecoding:PAM8:MAPPing:CUSTom:FILE:STATe[?] on page 575

Name	Description under
:DATA:LINecoding:PAM6[:SYMBol]:LEVel(1|2|3|4)[?]	:DATA:LINecoding:PAM6[:SYMBol]:LEVel(1|2|3|4)[?] on page 575
:DATA:LINecoding:PAM8[:SYMBol]:LEVel(1|2|3|4|5|6)[?]	:DATA:LINecoding:PAM8[:SYMBol]:LEVel(1|2|3|4|5|6)[?] on page 576
DATA:LINecoding:INVerse:PREcoding[?]	:DATA:LINecoding:INVerse:PREcoding[?] on page 577

:DATA:LINecoding[:VALue][?]

Syntax :DATA:LINecoding[:VALue] 'identifier', NRZ | PAM3 | PAM4 | PAM6 | PAM8

:DATA:LINecoding[:VALue]? 'identifier'

Input Parameters <identifier>: Any valid data output or data input identifier that supports the line coding. M*.DataOut1, M*.DataOut2, M*.DataOut3, M*.DataOut4, M*DataIn and INF*.DataIn1

Description The command selects the line coding to be used. The availability of individual selections depends on the hardware capabilities.

PAM3 is supported by M8042A, M8045A and M8046A.

PAM4 is supported by M8042A, M8043A, M8045A, M8046A, M8194A, M8195A, and M8196A.

PAM6 and PAM8 are supported by M8042A PG module, and UXR series real-time oscilloscopes.

This command is not supported by instruments that only support NRZ, like M8041A, M8051A, M8062A.

This query returns the present state.

This SCPI is applicable for M8042A, M8043A, M8045A, M8046A, M8194A, M8195A, M8196A, and UXR series real-time oscilloscope based error detector channels.

Example :DATA:LIN 'M1.DataOut1', PAM4

:DATA:LIN 'M2.DataOut1', PAM6

:DATA:LINecoding:PAM3:PRECoding[?]

Syntax :DATA:LINecoding:PAM3:PRECoding 'identifier', ON|OFF|1|0

:DATA:LINecoding:PAM3:PRECoding? 'identifier'

Input Parameters <identifier>: Any valid data output identifier that supports PAM3 encoding. M*.DataOut1, M*.DataOut2

Description The command enables/disables the precoding of the PAM3 symbols. Precoding is an optional extra step between the symbol mapping and PAM3 level mapping.

PAM3 is only supported by M8045A.

Example :DATA:LIN:PAM3:PREC 'M1.DataOut1',ON

:DATA:LINecoding:PAM3:PRECoding:VALue[?]

Syntax :DATA:LINecoding:PAM3:PRECoding:VALue 'identifier', ON|OFF|SEQC

:DATA:LINecoding:PAM3:PRECoding:VALue? 'identifier'

Input Parameters <identifier>: Any valid data output identifier that supports PAM3 encoding. M*.DataOut1, M*.DataOut2

ON | OFF | SEQC

Description This command sets the pre-coding value on the Data Out channel that supports PAM3 encoding.

This query returns the present state.

This command is applicable for M8042A and M8045A.

Example :DATA:LIN:PAM3:PREC:VAL 'M1.DataOut1', SEQC

:DATA:LINecoding:PAM3:PRECoding:FEEDback[?]

Syntax :DATA:LINecoding:PAM3:PRECoding:FEEDback 'identifier', <Feedback-Value>

:DATA:LINecoding:PAM3:PRECoding:FEEDback? 'identifier'

Input Parameters <identifier>: Any valid data output identifier that supports PAM3 encoding. M*.DataOut1, M*.DataOut2

<Feedback-Value> - ZERO | ONE | TWO

Description This command sets the initial pre-coder feedback value on the Data Out channel that supports PAM3 encoding. The feedback value can be 0, 1, or 2.

This query returns the present state.

This command is applicable for M8045A.

Example :DATA:LIN:PAM3:PREC:FEED 'M1.DataOut1', ONE

:DATA:LINecoding:PAM3[:SYMBol]:LEVel(1)[?]

Syntax :DATA:LINecoding:PAM3[:SYMBol]:LEVel(1) <identifier>, <NRf>

:DATA:LINecoding:PAM3[:SYMBol]:LEVel(1)? 'identifier'

Input Parameters <identifier>: Any valid data output identifier that supports PAM3 encoding. M*.DataOut1, M*.DataOut2

Description The command controls the PAM-3 symbol level 1. It can be used to adjust the actual output level that is being generated by a data output for PAM-3 symbol 1.

The levels of PAM-3 symbol 0 and 2 cannot be adjusted and are always kept at the value of 0% and 100% of the actual output voltage swing.

This query returns the present state.

This SCPI is applicable for M8042A and M8045A.

Example :DATA:LIN:PAM3:SYMB:LEV1 "M1.DataOut1",35

:DATA:LINecoding:PAM3:MAPPing[?]

Syntax :DATA:LINecoding:PAM3:MAPPing 'identifier', NONE | CUSTom

:DATA:LINecoding:PAM3:MAPPing? 'identifier'

Input Parameters <identifier>: Any valid data output/input identifier. M*.DataOut1, M*.DataOut2, M*DataIn

Description The command selects how the voltage levels are mapped to symbol numbers.

- Select NONE for default mapping symbol0 to low level, symbol1 to middle level and symbol2 to upper level.
- Select CUSTom to use a user defined mapping definition.

This query returns the present state.

This SCPI is applicable for M8042A, M8045A and M8046A.

Example :DATA:LIN:PAM3:MAPP 'M1.DataOut1', NONE

:DATA:LINecoding:PAM3:MAPPing:CUSTom[?]

Syntax :DATA:LINecoding:PAM3:MAPPing:CUSTom 'identifier', <"mapping definition">

:DATA:LINecoding:PAM3:MAPPing:CUSTom? 'identifier'

Input Parameters <identifier>: Any valid data output/input identifier. M*.DataOut1, M*.DataOut2, M*DataIn

Description The command defines a symbol number to voltage level mapping. The mapping defined with this command will be used when CUSTom is selected as the active mapping. The mapping is defined as a string that contains a comma separated list of symbols. The position in the list corresponds to the voltage level. "x,y,z" is mapping to lower, middle, upper level.

This query returns the present state.

This SCPI is applicable for M8042A, M8045A and M8046A.

Example :DATA:LIN:PAM3:MAPP 'M1.DataOut1', '1,2,0'

This command defines that Symbol 0 is being used for '1', Symbol 1 is being used for '2', Symbol 2 is being used for '0'.

:DATA:LINecoding:PAM3:SYMBol:ENABle[?]

Syntax :DATA:LINecoding:PAM3:SYMBol:ENABle 'identifier', ON|OFF|1|0

:DATA:LINecoding:PAM3:SYMBol:ENABle? 'identifier'

Input Parameters <identifier>: Any valid data output identifier that supports PAM3 encoding. M*.DataOut1, M*.DataOut2

Description The command enables/disables the pattern shift and symbol mapper functionality in the hardware.

This query returns the present state.

This SCPI is applicable for M8045A.

Example :DATA:LIN:PAM3:SYMB:ENAB 'M1.DataOut1', 1

:DATA:LINecoding:PAM3:SYMBol:COUNt[?]

Syntax :DATA:LINecoding:PAM3:SYMBol:COUNt 'identifier', <Shift-Value>

:DATA:LINecoding:PAM3:SYMBol:COUNt? 'identifier'

Input Parameters <identifier>: Any valid data output identifier that supports PAM3 encoding. M*.DataOut1, M*.DataOut2

<Shift-Value> - Number of symbols by which the pattern needs to be rotated. The shift value range is between 0 to 21844.

Description The command specify the number of symbols by which the pattern needs to be rotated. This option becomes available when "Enable Symbol/Pattern Shift" is turned on. See :DATA:LINecoding:PAM3:SYMBol:ENABle[?] on page 568.

This query returns the present state.

This SCPI is applicable for M8045A.

Example :DATA:LIN:PAM3:SYMB:COUN 'M1.DataOut1', 1

:DATA:LINecoding:PAM4:MAPPing[?]

Syntax :DATA:LINecoding:PAM4:MAPPing 'identifier', NONE | GRAY | CUSTom

:DATA:LINecoding:PAM4:MAPPing? 'identifier'

Input Parameters <identifier>: Any valid data output or data input identifier. M*.DataOut1, M*.DataOut2, M*.DataOut3, M*.DataOut4, M*DataIn and INF*.DataIn1

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. This option is available for M8195A, M8045A, M8043A and M8046A.
- Select GRAY if consecutive pairs of bits shall be Gray coded in order to map to the PAM4 symbols. This option is available for M8195A, M8045A, M8043A and M8046A.
- Select CUSTom to use a user defined mapping definition. This option is only available for M8045A, M8043A and M8046A.

This query returns the present state.

This SCPI is applicable for M8194A, M8195A, M8196A, M8042A, M8043A, M8045A, M8046A and real-time oscilloscope based error detector channels.

Example :DATA:LIN:PAM4:MAPP 'M1.DataOut1', NONE

:DATA:LIN:PAM4:MAPP 'M2.DataIn', CUST

:DATA:LINecoding:PAM4:MAPPing:CUSTom[?]

Syntax :DATA:LINecoding:PAM4:MAPPing:CUSTom 'identifier', <"mapping definition">

:DATA:LINecoding:PAM4:MAPPing:CUSTom? 'identifier'

Input Parameters <identifier>: Any valid data output or data input identifier.

Description The command defines a custom bit sequence to symbol number mapping. The mapping defined with this command will be used when CUSTom is selected as the active mapping.

The mapping is defined as a string that contains a comma separated list of bit pairs. The position in the list corresponds to the symbol number.

"ww,xx,yy,zz" is mapping to Symbol 0 to Symbol 3 according to the following table.

Symbol Number	Bit Pair
Symbol 0	ww
Symbol 1	xx
Symbol 2	yy
Symbol 3	zz

This query returns the present state.

This SCPI is applicable for M8045A, M8043A, M8046A, DSO634A, DSA634A and real-time oscilloscope based error detector channels.

Example :DATA:LIN:PAM4:MAPP 'M1.DataIn1', '11,00,01,10'

This command defines that Symbol 0 is being used for '11', Symbol 1 is being used for '00', Symbol 2 is being used for '01' and Symbol 3 is being used for '10'.

:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2)[?]

Syntax	:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2) 'identifier', <NRf>
	:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2)? 'identifier'
Input Parameters	<identifier>: Any valid data output 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 PAM4 level mappings. It can be used to adjust the actual output level that is being generated by a data output for a specific PAM4 symbol.
	The levels of PAM4 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 query returns the present state.
	This SCPI is applicable for M8194A, M8195A, M8196A, M8042A and M8045A.
Example	:DATA:LIN:PAM4:LEV1 'M1.DataOut1', 35pct;LEVel2 'M1.DataOut1', 70pct

:DATA:LINecoding:PAM4:PRECoding[?]

Syntax	:DATA:LINecoding:PAM4:PRECoding 'identifier', <ON|OFF|1|0>
	:DATA:LINecoding:PAM4:PRECoding? 'identifier'
Input Parameters	<identifier>: Any valid data output or data input identifier. M*.DataOut1, M*.DataOut2
Description	The command enables/disables the pre-coding of the PAM4 symbols.
	Pre-coding is an optional extra step between the symbol mapping and PAM4 level mapping. It is normally used together with a Gray code for the symbol mapping.
	The pre-coder does not enforce using a Gray code for symbol mapping. This allows device specific custom symbol mapping (in case of LSB first devices), or using patterns that already contain the required symbol

mapping (e.g. already gray encoded data).

This query returns the present state.

This SCPI is applicable for M8042A and M8045A.

Example :DATA:LIN:PAM4:PREC 'M1.DataOut1', ON

:DATA:LINecoding:PAM4:PRECoding:VALue[?]

Syntax :DATA:LINecoding:PAM4:PRECoding:VALue 'identifier', ON|OFF|SEQC

:DATA:LINecoding:PAM4:PRECoding:VALue? 'identifier'

Input Parameters <identifier>: Any valid data output identifier that supports PAM4 encoding. M*.DataOut1, M*.DataOut2

ON | OFF | SEQC

Description This command set the pre-coding value on the Data Out channel that supports PAM4 encoding.

This query returns the present state.

This command is applicable for M8045A.

Example :DATA:LIN:PAM4:PREC:VAL 'M1.DataOut1', SEQC

:DATA:LINecoding:PAM6:MAPPing[?]

Syntax :DATA:LINecoding:PAM6:MAPPing 'identifier', NONE | CUSTom

:DATA:LINecoding:PAM6:MAPPing? 'identifier'

Input Parameters <identifier>: Any valid data output or data input identifier. UXR Real time Scope (e.g.: INF*.DataIn1)

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 CUSTom to use a user defined mapping definition.

This query returns the present state.

This SCPI is applicable for M8042A and real-time oscilloscope based error detector channels.

Example :DATA:LIN:PAM6:MAPP 'INF1.DataIn1', NONE

:DATA:LIN:PAM6:MAPP? 'INF1.DataIn1'

:DATA:LINecoding:PAM6:MAPPing:CUSTom:FILE[?]

Syntax
:DATA:LINecoding:PAM6:MAPPing:CUSTom:FILE 'identifier', '<Custom Mapping File>'

:DATA:LINecoding:PAM6:MAPPing:CUSTom:FILE? 'identifier'

Input Parameters
<identifier>: Any valid data output or data input identifier.
UXR Real time Scope (e.g.: INF*.DataIn1), M*.DataOut1 and M*.DataOut2

<Custom Mapping File> - custom mapping file to be used for symbol mapping

Description
The command selects the custom symbol mapping file for PAM6.
For M8042A, this mapping is applied on the pattern memory level and requires a waveform download when changed. This mapping is therefore applied before the regular symbol mapping which updates in the real-time with pattern re-download.

This query returns the name of the custom mapping file.

This SCPI is applicable for real-time oscilloscope based error detector channels and M8042A.

Example
:DATA:LIN:PAM6:MAPP:CUST:FILE 'INF1.DataIn1', “Factory/CustomSymbolMappingPam6.xml”

:DATA:LIN:PAM6:MAPP:CUST:FILE? 'INF1.DataIn1' -> CustomSymbolMappingPam6.xml

:DATA:LINecoding:PAM6:MAPPing:CUSTom:FILE:STATe[?]

Syntax
:DATA:LINecoding:PAM6:MAPPing:CUSTom:FILE:STATe 'identifier', ON|OFF|1|0

:DATA:LINecoding:PAM6:MAPPing:CUSTom:FILE:STATe? 'identifier'

Input Parameters
<identifier>: Any valid data output or data input identifier. M*.DataOut1 and M*.DataOut2

ON|OFF|1|0

Description
The command selects whether the memory pattern bits should be mapped according to the definition in the custom mapping file. This effect applies before regular Symbol Mapping.

This query returns the present state.

This SCPI is applicable for M8042A.

Example :DATA:LIN:PAM6:MAPP:CUST:FILE:STAT ' M2.DataOut1',1

:DATA:LIN:PAM6:MAPP:CUST:FILE:STAT? ' M2.DataOut1' -> 1

:DATA:LINecoding:PAM8:MAPPing[?]

Syntax :DATA:LINecoding:PAM8:MAPPing 'identifier', NONE | CUSTom

:DATA:LINecoding:PAM8:MAPPing? 'identifier'

Input Parameters <identifier>: Any valid data output or data input identifier.
UXR Real time Scope (e.g.: INF*.DataIn1)

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 CUSTom to use a user defined mapping definition.

This query returns the present state.

This SCPI is applicable for M8042A and real-time oscilloscope based error detector channels.

Example :DATA:LIN:PAM8:MAPP 'INF1.DataIn1', NONE

:DATA:LIN:PAM8:MAPP? 'INF1.DataIn1'

:DATA:LINecoding:PAM8:MAPPing:CUSTom:FILE[?]

Syntax :DATA:LINecoding:PAM8:MAPPing:CUSTom:FILE 'identifier', '<Custom Mapping File>'

:DATA:LINecoding:PAM8:MAPPing:CUSTom:FILE? 'identifier'

Input Parameters <identifier>: Any valid data output or data input identifier.
UXR Real time Scope (e.g.: INF*.DataIn1), M*.DataOut1 and M*.DataOut2

<Custom Mapping File> - custom mapping file to be used for symbol mapping

Description The command selects the custom symbol mapping file for PAM8.

For M8042A, this mapping is applied on the pattern memory level and requires a waveform download when changed. This mapping is therefore applied before the regular symbol mapping which updates in the real-time with pattern re-download.

This query returns the name of the custom mapping file.

This SCPI is applicable for real-time oscilloscope based error detector channels and M8042A.

Example :DATA:LIN:PAM8:MAPP:CUST:FILE 'INF1.DataIn1', “Factory/CustomSymbolMappingPam8.xml”

:DATA:LIN:PAM8:MAPP:CUST:FILE? 'INF1.DataIn1' -> “Factory/CustomSymbolMappingPam8.xml”

:DATA:LINecoding:PAM8:MAPPing:CUSTom:FILE:STATe[?]

Syntax :DATA:LINecoding:PAM8:MAPPing:CUSTom:FILE:STATe 'identifier', ON|OFF|1|0

:DATA:LINecoding:PAM8:MAPPing:CUSTom:FILE:STATe? 'identifier'

Input Parameters <identifier>: Any valid data output or data input identifier. M*.DataOut1 and M*.DataOut2

ON|OFF|1|0

Description The command selects whether the memory pattern bits should be mapped according to the definition in the custom mapping file. This effect applies before regular Symbol Mapping.

This query returns the present state.

This SCPI is applicable for M8042A.

Example :DATA:LIN:PAM8:MAPP:CUST:FILE:STAT ' M2.DataOut1',1

:DATA:LIN:PAM8:MAPP:CUST:FILE:STAT? ' M2.DataOut1' -> 1

:DATA:LINecoding:PAM6[:SYMBol]:LEVel(1|2|3|4)[?]

Syntax :DATA:LINecoding:PAM6[:SYMBol]:LEVel(1|2|3|4) 'identifier', <NRf>

:DATA:LINecoding:PAM6[:SYMBol]:LEVel(1|2|3|4)? 'identifier'

Input Parameters <identifier>: Any valid data output identifier. M*.DataOut1, M*.DataOut2, M*.DataOut3 and M*.DataOut4

Range
- Symbol 1: 1 to 39 %
- Symbol 2: 21 to 59 %
- Symbol 3: 41 to 79 %
- Symbol 4: 61 to 99 %

Description
The command controls the PAM6 level mappings. It can be used to adjust the actual output level that is being generated by a data output for a specific PAM6 symbol.

The levels of PAM6 symbol 0 and 5 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 4 in a combined remote command transaction (see example below) to minimize hardware reconfiguration duration.

This query returns the present state.

This SCPI is applicable for M8042A.

Example
:DATA:LIN:PAM6:LEV1 'M1.DataOut1', 25pct;LEVel2 'M1.DataOut1', 45pct;LEVel3 'M1.DataOut1', 65pct;LEVel4 'M1.DataOut1', 85pct

:DATA:LINecoding:PAM8[:SYMBol]:LEVel(1|2|3|4|5|6)[?]

Syntax
:DATA:LINecoding:PAM8[:SYMBol]:LEVel(1|2|3|4|5|6) 'identifier', <NRf>

:DATA:LINecoding:PAM8[:SYMBol]:LEVel(1|2|3|4|5|6)? 'identifier'

Input Parameters
<identifier>: Any valid data output identifier.
M*.DataOut1, M*.DataOut2, M*.DataOut3 and M*.DataOut4

Range
- Symbol 1: 1 to 27 %
- Symbol 2: 15 to 41 %
- Symbol 3: 29 to 55 %
- Symbol 4: 43 to 69 %
- Symbol 5: 57 to 83 %
- Symbol 6: 71 to 99 %

Description
The command controls the PAM8 level mappings. It can be used to adjust the actual output level that is being generated by a data output for a specific PAM8 symbol.

The levels of PAM8 symbol 0 and 7 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 6 in a combined remote command transaction (see example below) to minimize hardware reconfiguration duration.

This query returns the present state.

This SCPI is applicable for M8042A.

Example :DATA:LIN:PAM8:LEV1 'M1.DataOut1', 14pct;LEVel2 'M1.DataOut1', 28pct;LEVel3 'M1.DataOut1', 42pct;LEVel4 'M1.DataOut1', 56pct;LEVel5 'M1.DataOut1', 70pct;LEVel6 'M1.DataOut1', 85pct

:DATA:LINecoding:INVerse:PREcoding[?]

Syntax :DATA:LINecoding:INVerse:PREcoding 'identifier', <ON|OF|1|0>

:DATA:LINecoding:INVerse:PREcoding? 'identifier'

Input Parameters <identifier>: Any valid data output identifier.
M*.DataIn

Description This command enabled/disables inverse precoder for PAM3 line coding.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :DATA:LIN:INV:PRE 'M2.DataIn',ON

:DATA:LIN:INV:PRE? 'M2.DataIn' -> 1

:DATA:FEC Subnode

The DATA:FEC subnode describes programming specification for the FEC error insertion feature.

FEC support in M8070B is enabled by installing a module specific FEC license. FEC licenses are available for M8042A and M8045A modules. For details on the licenses, refer to the *M8000 Series User Guide* or *Online Help*.

For M8045A, the following startup options can be selected while launching the M8070B application:

- BERT with FEC (50G)
- BERT with FEC (100G KP4)
- BERT with FEC (200G KP4)
- BERT with FEC (400G KP4)
- BERT with FEC (2X400G KP4)

For M8042A, currently the 100G-1 KP4, 200G-2 KP4 and 2x400G-4 KP4 encodings are supported by the M8070B application.

FEC is enabled by selecting one of the FEC factory patterns.

- IEEE_802_3cd_RS_544_514_Scrambled_Idle
- IEEE_802_3cd_RS_544_514_Remote_Fault

Due to the differences between FEC frame length and sequencer word width, there will be phases of invalid FEC data when transitioning from a FEC encoded sequence block into a non-FEC sequence block and vice versa.

:DATA:FEC:EINSertion Subsystem

When a FEC pattern is used in the sequence, then the pattern generator offers a FEC specific error insertion.

The FEC error insertion is independent from the error insertion controlled with the :OUTPut:EINSertion subnode (see :OUTPut:EINSertion Subnode on page 305). Both error insertions can be used at the same time.

This subnode has the following SCPI structure:

```
:DATA
└:FEC
  ├:EINSertion
  │ ├:STATe[?]
  │ ├:ERRors[?]
  │ └:BIP:ONCE
  ├:ENCoding[?]
  ├:PCSLanes?
  ├:STATe?
  └:SETTings?
```

This subnode has the following commands:

Table 100

Name	Description under
:EINSertion:STATe[?]	:DATA:FEC:EINSertion:STATe[?] on page 580
:EINSertion:ERRors[?]	:DATA:FEC:EINSertion:ERRors[?] on page 580
:EINSertion:BIP:ONCE	:DATA:FEC:EINSertion:BIP:ONCE on page 581
:DATA:FEC:ENCoding[?]	:DATA:FEC:ENCoding[?] on page 581
:DATA:FEC:PCSLanes?	:DATA:FEC:PCSLanes? on page 582
:DATA:FEC:STATe?	:DATA:FEC:STATe? on page 582
:DATA:FEC:SETTings[?]	:DATA:FEC:SETTings[?] on page 582

:DATA:FEC:EINSertion:STATe[?]

Syntax	:DATA:FEC:EINSertion[:STATe] 'identifier', <ON|OFF|1|0>
	:DATA:FEC:EINSertion[:STATe]? 'identifier'
Input Parameters	<identifier>: Any valid data output identifier. 'M*.DataOut1','M*.DataOut2'
	ON | OFF | 1 | 0
Return Range	1 | 0
Description	This command enables or disables the continuous FEC error insertion. The inserted errors are after the FEC encoding.
	This query returns the present state.
	This SCPI is applicable for M8045A.
Example	:DATA:FEC:EINS:STAT 'M1.DataOut1',ON
	:DATA:FEC:EINS:STAT? 'M1.DataOut1'
	1

:DATA:FEC:EINSertion:ERRors[?]

Syntax	:DATA:FEC:EINSertion:ERRors 'identifier', <NRf>
	:DATA:FEC:EINSertion:ERRors ? 'identifier'
Input Parameters	<identifier>: Any valid data output identifier. 'M*.DataOut1','M*.DataOut2'
	ON | OFF | 1 | 0
Return Range	0 to 16
Description	This command specifies the number of symbols to be corrupted within a single FEC frame. The inserted errors are at random symbol positions. Each corrupted symbol will contain exactly one bit error at a random bit position.
	This query returns the present state.
	This SCPI is applicable for M8045A.
Example	:DATA:FEC:EINS:ERR 'M1.DataOut1',8
	:DATA:FEC:EINS:ERR? 'M1.DataOut1'
	8

:DATA:FEC:EINSertion:BIP:ONCE

Syntax	:DATA:FEC:EINSertion:BIP:ONCE 'identifier'
Input Parameters	<identifier>: Any valid data output identifier. 'M*.DataOut1','M*.DataOut2'
Return Range	1 | 0
Description	This command inserts a single Bit Interleaved Parirty (BIP) error. This is a pre-FEC error and can be inserted at any time, even if the FEF error insertion is disabled. This SCPI is applicable for M8045A.
Example	:DATA:FEC:EINS:BIP:ONCE 'M1.DataOut1'

:DATA:FEC:ENCoding[?]

Syntax	:DATA:FEC:ENC '<identifier>', <FEC Encoding> :DATA:FEC:ENC? '<identifier>'
Input Parameters	<identifier>: Any valid data output identifier. 'M*.DataOut1','M*.DataOut2' <FEC Encoding>: FEC100G1 | FEC200G2 | FEC2x400G4
Description	For M8042A, this command sets the FEC encoding according to the given standard. For M8042A and M8045A, this query returns the generated FEC encoding according to the given standard. This SCPI is applicable for M8042A and M8045A.
Example	:DATA:FEC:ENCoding 'M1.DataOut1', FEC100G1 :DATA:FEC:ENCoding? 'M1.DataOut1' -> FEC100G1

:DATA:FEC:PCSLanes?

Syntax	:DATA:FEC:PCSL? '<identifier>'
Input Parameters	<identifier>: Any valid data output identifier. 'M*.DataOut1','M*.DataOut2'
Description	This query returns the PCS lanes generated on the channel. This does not affect the patterns that are not FEC encoded. This SCPI is applicable for M8042A and M8045A.
Example	:DATA:FEC: PCSLanes? 'M1.DataOut1' -> PCS0TO3

:DATA:FEC:STATe?

Syntax	:DATA:FEC:STATe? 'Identifier'
Input Parameters	<identifier>: Any valid data output identifier. 'M*.DataOut1','M*.DataOut2'
Description	This query returns the FEC state. This reports if FEC generation used by the sequence or not. i.e. state will be on for FEC encoded patterns else it will be off. This SCPI is applicable for M8042A and M8045A.
Example	:DATA:FEC:STAT? 'M2.DataOut1'

:DATA:FEC:SETTings[?]

Syntax	:DATA:FEC: SETTings 'Identifier', <1 | 0 | ON | OFF> :DATA:FEC: SETTings? 'Identifier'
Input Parameters	<identifier>: Any valid data output identifier. 'M*.DataOut1','M*.DataOut2' 1 | 0 | ON | OFF
Description	This command configures the system for FEC generation to be used by the sequence. This SCPI is applicable for M8042A.
Example	:DATA:FEC:SETT 'M2.DataOut1', 1 :DATA:FEC:SETT? 'M2.DataOut1' -> 1

:DATA:COMPare Subnode

This subnode has the following SCPI structure:

```
:DATA
└:COMPare
  └:EVENt
    └:ERRor
      ├:MODE[?]
      ├:ECOunt[?]
      └:BCOunt[?]
```

This subnode has the following commands:

Table 101

Name	Description under
:ERRor:MODE[?]	:DATA:COMPare:EVENt:ERRor:MODE[?] on page 584
:ERRor:ECOunt[?]	:DATA:COMPare:EVENt:ERRor:ECOunt[?] on page 584
ERRor:BCOunt[?]	:DATA:COMPare:EVENt:ERRor:BCOunt[?] on page 584

:DATA:COMPare:EVENt:ERRor:MODE[?]

Syntax	:DATA:COMPare:EVENt:ERRor:MODE 'identifier', <Enum>
	:DATA:COMPare:EVENt:ERRor:MODE?
Input Parameters	Identifier: 'M*.DataIn'
	<Enum>: SINGle | MULTiple
Description	This command controls how detected errors are reported.
	This query returns the present state.
	This SCPI is applicable for M8043A and M8046A.
Example	:DATA:COMP:EVEN:ERR:MODE 'M1.DataIn', SINGle
	:DATA:COMP:EVEN:ERR:MODE? 'M1.DataIn'

:DATA:COMPare:EVENt:ERRor:ECOunt[?]

Syntax	:DATA:COMPare:EVENt:ERRor:ECOunt 'identifier', <error-count>
	:DATA:COMPare:EVENt:ERRor:ECOunt?
Input Parameters	Identifier: 'M*.DataIn'
	<error-count>: Number of errors.
Description	This command defines the number of errors that causes the generation of the error event.
	This query returns the present state.
	This SCPI is applicable for M8046A.
Example	:DATA:COMP:EVEN:ERR:ECO 'M1.DataIn', 2
	:DATA:COMP:EVEN:ERR:ECO? 'M1.DataIn'

:DATA:COMPare:EVENt:ERRor:BCOunt[?]

Syntax	:DATA:COMPare:EVENt:ERRor:BCOunt 'identifier', <error-count>
	:DATA:COMPare:EVENt:ERRor:BCOunt?
Input Parameters	Identifier: 'M*.DataIn'
	<error-count>: Number of error count.
Description	This command defines the number of bits that defines the reporting

interval.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :DATA:COMP:EVEN:ERR:BCO 'M1.DataIn', 5120

:DATA:COMP:EVEN:ERR:BCO? 'M1.DataIn'

:DATA:IDLepattern:VALue[?]

Syntax :DATA:IDLepattern:VALue 'identifier', PRBS | STATIC0

:DATA:IDLepattern:VALue? 'identifier'

Input Parameters Identifier: 'M1.DataOut1', 'M2.DataOut2'

PRBS | STATIC0

Description This command sets the idle pattern. The idle pattern can be set either PRBS or Static 0.

This query returns the present state.

This SCPI is applicable for M8045A.

Example :DATA:IDL:VAL M1.DataOut1, PRBS

:DATA:IDL:VAL? M1.DataOut1

M8047B Subsystem

This subsystem has the following SCPI structure:

```
:M8047B
 ├:INPut
 │ └:EQUalization
 │    ├:BOOSt[?]
 │    ├:EYEExpander[?]
 │    └:DCGain[?]
 └:OUTPut
   └:VOD[?]
```

This subsystem has the following commands:

Table 102

Name	Description under
:M8047B:INPut:EQUalization:BOOSt	:M8047B:INPut:EQUalization:BOOSt on page 587
:M8047B:INPut:EQUalization:EYEExpander[?]	:M8047B:INPut:EQUalization:EYEExpander[?] on page 587
:M8047B:INPut:EQUalization:DCGain[?]	:M8047B:INPut:EQUalization:DCGain[?] on page 588
:M8047B:OUTPut:VOD[?]	:M8047B:OUTPut:VOD[?] on page 588

:M8047B:INPut:EQUalization:BOOSt

Syntax	:M8047B:INPut:EQUalization:BOOSt 'identifier', <Value>
	:M8047B:INPut:EQUalization:BOOSt? 'identifier'
Input Parameters	Identifier: 'M*.DataIn'
	Value: 0-33
Description	This command sets the equalizer boost.
	This query returns the current value of equalizer boost.
Example	Command
	:M8047B:INP:EQU:BOOS 'M3.DataIn', 2
	Query
	:M8047B:INP:EQU:BOOS? 'M3.DataIn'

:M8047B:INPut:EQUalization:EYEExpander[?]

Syntax	:M8047B:INPut:EQUalization:EYEExpander 'identifier', <Value>
	:M8047B:INPut:EQUalization:EYEExpander? 'identifier'
Input Parameters	Identifier: 'M*.DataIn'
	Value: 0-3
Description	This command sets the value of the PAM4 eye expander (range: 0 to 3).
	This query returns the current value of PAM4 eye expander.
Example	Command
	:M8047B:INP:EQU:EYEE 'M3.DataIn', 2
	Query
	:M8047B:INP:EQU:EYEE? 'M3.DataIn'

:M8047B:INPut:EQUalization:DCGain[?]

Syntax	:M8047B:INPut:EQUalization:DCGain 'identifier', <Value>
	:M8047B:INPut:EQUalization:DCGain? 'identifier'
Input Parameters	Identifier: 'M*.DataIn'
	Value: 0-3
Description	This command sets the Equalizer DC Gain.
	This query returns the current value of Equalizer DC Gain.
Example	Command
	:M8047B:INP:EQU:DCG 'M3.DataIn', 3
	Query
	:M8047B:INP:EQU:DCG? 'M3.DataIn'

:M8047B:OUTPut:VOD[?]

Syntax	:M8047B:OUTPut:VOD 'identifier', <Value>
	:M8047B:OUTPut:VOD? 'identifier'
Input Parameters	Identifier: 'M*.DataOut'
	Value: 0-3
Description	This command sets the Output VOD.
	This query returns the current value of Output VOD.
Example	Command
	:M8047B:OUTP:VOD M3.DataOut, 1
	Query
	:M8047B:OUTP:VOD? 'M3.DataOut'

ARM Subsystem

The ARM subsystem has the following command:

:ARM:DELay[?]

Command	:ARM:DELay 'identifier', <delay>|MINimum|MAXimum
Query	:ARM:DELay? 'identifier'
Parameters	'identifier' – 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3' or 'M*.DataOut4'
	Delay - Specify delay in seconds
Description	This command sets or queries the data delay of a specific channel. The unit is in seconds.
	This SCPI is applicable for M8053A.
Example	:ARM:DELay 'M*.DataOut1', 10.0e-12
	:ARM:DELay? 'M*.DataOut1' -> 10.0e-12
	:ARM:DELay 'M*.DataOut1', MAX
	:ARM:DELay? 'M1.DataOut1' -> 2.5e-08

Index

L

M

N

O

P

Edition 19.1, December 2025

M8000-91B09

www.keysight.com