Keysight M8085A MIPI® D-PHY Editor

User Guide

KEYSIGHT TECHNOLOGIES

Notices

Trademarks

MIPI® C-PHY™ and MIPI® D-PHY™ are registered trademarks of the MIPI Alliance.

Manual Part Number

M8085-91040

Edition

Edition 6.2, June 2019

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.

Contents

3 Using MIPI D-PHY Editor User Interface

4 SCPI Programming

1 Introduction

This chapter gives an overview on the high-speed serial interface called MIPI D-PHY, which provides high throughput performance over bandwidth limited channels for connecting to peripherals, along with an overview of the Keysight M8085A MIPI D-PHY Editor plug-in, the Data File and Sequence File definitions used in the MIPI D-PHY Editor plug-in.

Overview

The M8070B software has an add-on **MIPI D-PHY Editor** that generates MIPI D-PHY signals so as to test the DUTs that are compatible to this standard.

MIPI D-PHY describes a source synchronous, high speed, low power, low cost PHY, especially suited for mobile applications. The MIPI D-PHY specification (http://mipi.org/) has been written primarily for the connection of camera and display applications to a host processor. Nevertheless, it can be applied to many other applications.

The MIPI D-PHY provides a synchronous connection between Master and Slave. A practical PHY Configuration consists of a clock signal and one or more data signals. The clock signal is unidirectional, originating at the Master and terminating at the Slave. The data signals can either be unidirectional or bi-directional depending on the selected options.

The PHY uses two wires per Data Lane plus two wires for the Clock Lane. This gives four wires for the minimum PHY configuration. In High-Speed mode, each Lane is terminated on both sides and driven by a low-swing, differential signal. In Low-Power mode all wires are operated single-ended and non-terminated.

The **MIPI D-PHY Editor** is a licensed feature. To enable the **MIPI D-PHY Editor**, the following licenses/options are required:

- System software license (“M8070B-0TP” or “M8070B-0NP”)
- MIPI D-PHY Editor license (“refer to Chapter 3, License Requirements on page 50)
- M8195A AWG, which uses the following options:
  - Option -001, -002, or -004: With these options the number of channels is selected. The M8195A is available in a one channel (-001), two channel (-002) or 4 channel (-004) version. A software upgrade from one to two channels is possible by installing option U02. A software upgrade from two to four channels is possible by installing option U04. In order to upgrade from one to four channels, first option –U02 and next –U04 must be installed.
  - Option -16G: This option offers 16384 MSa (=16 GSa) waveform memory for the M8195A. Option -16G is software upgradeable.
  - Option -SEQ: This option offers extensive sequencing capabilities. Option -SEQ is software upgradeable.
  - Option -FSW: This option enables the M8195A to externally select or step through segments or sequences faster than every 500 µs. Option -FSW is export controlled and is software upgradeable.

- Option -1A7, -Z54: Calibration options.

For more details on how to install these licenses, refer to *Keysight M8000 Series of BER Test Solutions User Guide*.

NOTE **You require the AWG large memory option (0G2) to test large patterns using the MIPI D-PHY Editor plug-in.**

Modes supported by the MIPI D-PHY Editor

The MIPI D-PHY Editor plug-in supports two different mode groups:

- The Burst mode: In the Burst Modes (Burst, Burst Continuous Clock, Pure HS) a block of data is repeated infinitely. This block can contain LP Data and HS Data, pure LP data or pure HS data depending on the content of the data given for HS and LP. If either HS or LP data is empty (an empty text box for that particular data type), the software generates pure LP/HS data. If Pure HS is selected, no LP11 transitions are included and all LP data is neglected. The burst mode requires two sequences of data, one sequence for HS and other for LP. Each sequence can be set manually or be read by the software from a data file.A block consists of a name followed by one or more LP blocks:
  - LP Blocks: The format is LPxyNn. x and y can be 0 or 1, depending on the desired LP state. n represents the number of LP states.
- The Frame mode: In these modes (Frames and Frames, Continuous HS Clock), a sequence file allows running a sequence of blocks ending with an infinite loop over the complete number of blocks or an selection of the last blocks. Each individual block can be repeated (“looped”) N-times and the number of repetitions N can be selected for each block separately. In addition to the data files, the frame modes require a sequence file. It specifies the data rates and the sequence of blocks. The M8085A MIPI D-PHY Editor plug-in supports sequence files conforming to the CSI and DSI protocol standards. While the section below describes sequence file formats in general, refer to Sequence File Definition for CSI and DSI on page 17 to understand sequence file formats for CSI and DSI protocols respectively.

Data File Format (*.dat or *.txt)

For data definition in the MIPI D-PHY Editor plug-in, the hexadecimal (HEX) format is required. Bytes are represented in two digits, each ranging from 0 to 9 and A to F. The leading string “0x” is optional. Supported separators between data bytes are:

, (comma)

; (semicolon)

space (blank)

tab

line feed

nothing

Some examples:

- 0x01, 0xF3, 0x23
- 0134E734FF
- 32 FF E5 44

In addition to the pure HEX data, special commands are abbreviations of lists of hex bytes:

- 0x<HEX code>N<count>: repeat the byte <HEX code> <count> times.

For example: 0xABN5 is equal to AB AB AB AB AB.

- 0x<HEX code>S<count> repeat the byte <HexCode> N times for each data lane. In a one-data lane configuration the macro is similar to 0x<HEX code>N<count>, but in a multi-lane setup, it will repeat <HexCode> at each data lane <count> times.

For example, 0xABS4 is equal to AB AB AB AB in a 1-lane configuration, but will become AB AB AB AB at each data lane for a multi-lane setup.

An example for 2 data lanes:

0x02c05 is equal to D0: 02 03 04 05 and D1: 02 03 04 05.

For the counter with the “x” and two data lanes, 0x02x05 would lead to D0: 02 04 and D1: 03 05.

NOTE **The special commands require the leading “0x”, otherwise they will not be recognized.**

Sequence File Format(*.seq)

For waveform generation in the MIPI D-PHY Editor plug-in, you may use sequence files. In the sequence file the data rate, data blocks and sequence are defined. Note that all parameters are even integers. The structure of a sequence file is shown in Figure 1.

HSFreq: <frequency in bits/s>

Blocks:

<BlockName 1>: <Block Definition 1>, ..., <Block Definition n1>;

...

<BlockName M>: <Block Definition 1>, ...,<Block Definition nM>;

Sequence:

1. <BlockName J>, <Loop Count R>; - First block

...

<N>. <BlockName K>, <Loop Count S>; - Nth block

...

<P>. <BlockName L>, <Loop Count T>; - Pth block

[LoopTo N]

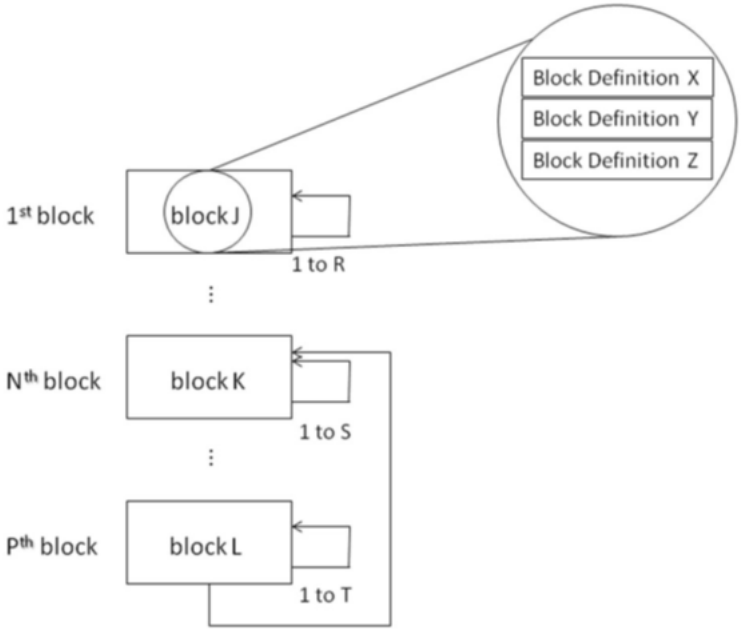


Figure 1 Block Diagram of a Sequence Example

Each block can comprise multiple sub-blocks (1 to n). Sub-blocks can be used in multiple blocks. In the sequence, blocks can be used as often as needed. Within the sequence, the LoopTo expression starts an infinite loop from block <N> to the last block <P>. If no LoopTo expression is specified, an infinite loop is created from block 1 to block P. Valid block definitions are:

- LP00, LP01, LP10, LP11: for a single LP state.
- LP<LP state 00,01,10, or 11>[<LP state 00,01,10, or 11>...] Only available in the Manual LP Framing mode and for LP pattern. It allows a sequence of LP states to be defined in one statement.

  For example: LP1101001011
- LPB"<filename>": for generating LP data specified in the file with the name <filename>. The file should be in the same folder as the sequence file. At the beginning of the data an escape trigger for LP or ULP data mode is sent before the data and a Mark-0/1 sequence is sent after the end of the data. For the data, the data file format given above must be used.

- LP<00, 01, 10, or 11>N<number of bits>: The LP state is sent <number of [HS] bits> times.
- LP<00, 01, 10, or 11>E<number of bits>: The LP state is sent until the block size reaches the number of HS bits given in <number of bits>.
- LPHSE<number of bits>: The block is filled with LP11 states and a LP–HS transition until the number of HS bits <number of bits> is reached.
- LPHS: The macro adds a LP–HS transition. It is mainly for influencing the block ending. If for example the following block starts with HS data, then the LP-HS transition will done at the end of the actual block. Without this macro the LP-HS transition would be added to the beginning of the following block. You do not need the macro for the following B macros, since they will trigger a LP-HS transition automatically if the previous block description contained LP states.
- B"<filename>": for generating HS data given in the file <filename>. The file should be in the same folder as the sequence file. If necessary, a LP-to-HS transition is generated before the data. A HS-to-LP transition is added if the following block contains LP states.
- BL<number of blanking bytes>: for generating HS blanking packets with a number of blanking bytes given in <number of blanking bytes>. The id (0x19), word counter, ECC (Error-Correcting Code) and CRC (Cyclic Redundancy Check) are calculated and added such that the number of bits is equal to (<number of blanking bytes>+6)*8;
- C<3 hex bytes>: For generating short packets like they are described in the CSI or DSI specifications. The resulting short packet will have the first byte as Data ID of this packet and the following two bytes as Packet DATA followed by the ECC for that packet. If necessary, a LP-to-HS transition is generated before the data. A HS-to-LP transition is added if the following block contains LP states.
- C<1 hex byte>"<filename>": for generating a long packet. The content in the filename will be taken as payload. The header will have the given hex byte as ID followed by a two byte word counter, followed by a ECC for that header data. At the end of the payload, a two byte CRC is added. If necessary, an LP-to-HS transition is generated before the data. An HS-to-LP transition is added if the following block contains LP states.

- PRBS<no.>(<seed1>|<seed2>| ... | <seedN>): for generating a PRBS of the polynomial <no.> with a seed of <seed1-N> for each lane. The <no.> is just a decimal number (only 9 is allowed), and the seeds are given in a hex number (example 0x789A). The number of input seeds should be the same as active lanes (example for 3 lanes <seed1> goes to D0, <seed2> goes to D1 and <seed3> goes to D2). If #seeds > #lanes the latest seeds will be ignored. If #seeds < #lanes an exception will be thrown. As a special case, if only one seed is provided but more than one lane is active, the pattern is distributed among all lanes.
- ULPEntry: Adds the ULP Entry escape sequence to the block. After the ULPEntry LP00 states plus finally an ULPExit should follow to create a specification conform ULP sequence.
- ULPExit<number of LP10 states>: Creates a ULP exit sequence. It is not allowed to combine this block definitions with other definitions, which means in this case the block must only contain this macro and no other.
- InitialSkewCal: adds an HS clock pattern to all data lanes to generate the skew calibration block at the beginning of the data transmission (see *MIPI Alliance Specification for D-PHY^{SM} version 2.1*). It is not allowed to combine this block definition with other definitions, which means in this case the block must only contain this macro and no other. The block before needs to contain a LPHSE or LPHS block as last entry to force the frame generator to put the special LP-HS transition for the sync pattern at the end of the previous block. In this case, the sync word is replaced by 0xFF. In the sequence definition the loop counter value for the block will be replaced by a suitable value to generate the clock pattern for duration **TX-InitialSkewCal Duration** parameter given via the Editor's user interface.
- PeriodicSkewCal: similar to InitialSkewCal, adds an HS clock pattern for line de-skew. This block is meant to be within the looped part of the pattern, and the same restriction as for InitialSkewCal apply. The duration of the line de-skew will be set by the **TX-PeriodicSkewCal Duration** parameter in the Editor's user interface.

NOTE

- If blocks are looped, the beginning of the block should have the same kind of data mode (LP or HS) as the block following it, otherwise the block loop will result in invalid LP to HS transitions.
- Video Frames which contain LP11 blanking periods should be rotated so that the block definition always ends with a LP11E command.
- If only the header contains LP11 states, the header block should end with LPHSE to start the HS transmission at the end of the header block.
- In case of PureHS mode, an initial LP to HS transition is added in the form of a hidden intro block in the waveform generation, before an infinite loop of pure HS data stream is generated.

In any case to each sequence, an LP11 block is added to the beginning of the sequence. If required, an LP-HS transition is added if the first block starts with HS data. Also, a sync block is added for the synchronization of the ParBERT LP and HS subsystems. These blocks need not to be added explicitly to the sequence. They are added automatically for all sequences, that is, even if a sequence with pure HS blocks is given.

Automatic Clock Generation

Depending on the data lane pattern, a clock lane signal is generated which adheres to the MIPI D-PHY standard. For every high speed burst on the data lane a transition from LP to HS will be done on the clock lane. The HS clock will be active as long as the HS burst on the data lane is active. After the HS data burst ended the clock lane will also transition back to its stop state in case the next HS burst on the clock is far enough away to fit in the transition segments. In case this LP mode is active for a shorter duration on the data lane the HS clock will stay active for both bursts and not transition back to clock LP mode.

2 Sequence File Definition for CSI and DSI

The following sections describe the various elements of a sequence file for the CSI and DSI protocols, such that the sequence file definition generates a waveform that conforms to the MIPI Specification for Camera Serial Interface (CSI) and MIPI Alliance Specification for Display Serial Interface (DSI), respectively.

Generally, a sequence file consists of three elements that form together a sequence:

- HS Data Rate
- Blocks
- Sequence

Following is a real time example of a sequence file definition:

Example of a Sequence File Definition:

HSFreq: 200MBit/s;

Blocks:

LPInit1: LPB"Esc0ms.txt",LP11E13728;

LPPause: LP11N1024;

LPInit2: LPB"Esc100ms.txt",LP11E13728;

LPInit3: LPB"Esc200ms.txt",LP11E13728;

Header: B"FirstHsLine.txt",LP11E6016;

Video: B"VideoLine.txt",LP11E6016;

Sequence:

1 LPInit1,1;
2 LPPause,20000;
3 LPInit2,1;
4 LPPause,20000;
5 LPInit3,1;
6 LPPause,20000;
7 Header,1;
8 Video,319;

LoopTo 6;

Following are real time examples of DSI and CSI sequence file definitions, respectively:

Example of a DSI Sequence File Definition:

```
HSFreq: 432.432MBit/s;

Blocks:

LPInit: LPB"HSyncEnd.txt",LP11E13728;

HSync: B"HSyncEnd.txt",LP11E12736,B"HSyncStart.txt",LP11E13728;

VSyncStart: B"HSyncEnd.txt",LP11E12736,B"VSyncStart.txt",LP11E13728;

VSyncEnd: B"HSyncEnd.txt",LP11E12736,B"VSyncEnd.txt",LP11E13728;

Video: B"HSyncEnd.txt",LP11E960,B"Video480pHSyncStart.txt",LP11E13728;


Sequence:
1   LPInit,1;
2   HSync,5;
3   VSyncEnd,1;
4   HSync,29;
5   Video,480;
6   HSync,9;
7   VSyncStart,1;
LoopTo 2;
```

Example of a CSI Sequence File Definition:

```
HSFreq: 158 MBit/s;


Blocks:

FrameStart: B"FrameStart.txt",LP11E10880;

Blanking: LP11E10880;

Video: C1E"compliance640_480.txt",LP11E10880;

FrameEnd: B"FrameEnd.txt",LP11E2048;
```

```
Sequence:
1   FrameStart,1;
2   Blanking,1;
3   Video,480;
4   Blanking,2;
5   FrameEnd,1;
```

Sequence File Definition for CSI

Overview

CSI is a MIPI Alliance standard for serial interface between a camera module and host processor. CSI adheres to the Low-Level Protocol (LLP), which is a byte orientated, packet based protocol that supports the transport of arbitrary data using Short and Long packet formats. Two packet structures are defined for low-level protocol communication: Long packets and Short packets. The format and length of Short and Long Packets depends on the choice of physical layer (MIPI C-PHY or MIPI D-PHY). For each packet structure, exit from the low power state followed by the Start of Transmission (SoT) sequence indicates the start of the packet. The End of Transmission (EoT) sequence followed by the low power state indicates the end of the packet. However, in CSI implementation, one burst consists of only one packet and LP11 state must be inserted before the start of a burst. Since it requires to go to LP state always, an explicit EoT packet is not required.

NOTE **A sequence file used for MIPI D-PHY or MIPI C-PHY conformance testing cannot be used for CSI/DSI conformance testing unless the header, payload and checksum data is included in the CSI/DSI block definitions in the sequences else the device rejects the packet.**

Long and Short Packet Formats

Long Packet

For MIPI D-PHY, a Long Packet shall be identified by Data Types 0x10 to 0x37. A Long Packet for the MIPI D-PHY physical layer option shall consist of three elements: a 32-bit Packet Header (PH), an application specific Data Payload with a variable number of 8-bit data words, and a 16-bit Packet Footer (PF). The Packet Header is further composed of three elements: an 8-bit Data Identifier, a 16-bit Word Count field and an 8-bit ECC. The Packet footer has one element, a 16-bit checksum (CRC).

For C-PHY, the Long Packet structure for the C-PHY physical layer option shall consist of four elements: a Packet Header (PH), an application specific Data Payload with a variable number of 8-bit data words, a 16-bit Packet Footer (PF), and zero or more Filler bytes (FILLER). The Packet Header is 6N x 16-bits long, where N is the number of C-PHY physical layer Lanes. The Packet Header consists of two identical 6N-byte halves, where each half consists of N sequential copies of each of the following

fields: a 16-bit field containing eight Reserved bits plus the 8-bit Data Identifier (DI); the 16-bit Packet Data Word Count (WC); and a 16-bit Packet Header checksum (PH-CRC) which is computed over the previous four bytes. The value of each Reserved bit shall be zero. The Packet Footer consists of a 16-bit checksum (CRC) computed over the Packet Data using the same CRC polynomial as the Packet Header CRC and the Packet Footer used in the MIPI D-PHY physical layer option. Packet Filler bytes are inserted after the Packet Footer, if needed, to ensure that the Packet Footer ends on a 16-bit word boundary and that each MIPI D-PHY physical layer Lane transports the same number of 16-bit words (i.e. byte pairs).

For both physical layer options, the 8-bit Data Identifier field and the 16-bit Word Count (WC) field contain identical data. The CSI receiver reads the next WC 8-bit data words of the Data Payload following the Packet Header. The length of the Data Payload shall always be a multiple of 8-bit data words. For both physical layer options, once the CSI receiver has read the Data Payload, it then reads the 16-bit checksum (CRC) in the Packet Footer and compares it against its own calculated checksum to determine if any Data Payload errors have occurred.

In either case, Packet Data length = Word Count (WC) * Data Word Width (8-bits).

Short Packet

For each option (MIPI C-PHY and MIPI D-PHY), the Short Packet structure matches the Packet Header of the corresponding Low Level Protocol Long Packet structure with the exception that the Packet Header Word Count (WC) field shall be replaced by the Short Packet Data Field. A Short Packet shall be identified by Data Types 0x00 to 0x0F. A Short Packet shall contain only a Packet Header; neither Packet Footer nor Packet Filler bytes shall be present. For Frame Synchronization Data Types, the Short Packet Data Field shall be the frame number. For Line Synchronization Data Types, the Short Packet Data Field shall be the line number.

For the MIPI D-PHY physical layer option, the Error Correction Code (ECC) byte allows single-bit errors to be corrected and 2-bit errors to be detected in the Short Packet.

For the C-PHY physical layer option, the 16-bit Checksum (CRC) allows one or more bit errors to be detected in the Short Packet but does not support error correction.

Short Packet Data Types shall be transmitted using only the Short Packet format. Refer to *Table 6 Synchronization Short Packet Data Type Codes* of the *MIPI Alliance Specification for Camera Serial Interface (CSI)*, which indicates that Data Type for Frame Start Code is 0x00 and Data Type for Frame End Code is 0x01.

NOTE Between Low Level Protocol packets, there must always be an HS-LP or an LP-HS transition.

Frame and Line Synchronization Packets

Frame Synchronization Packets

Each image frame shall begin with a Frame Start (FS) Packet containing the Frame Start Code. The FS Packet shall be followed by one or more long packets containing image data and zero or more short packets containing synchronization codes. Each image frame shall end with a Frame End (FE) Packet containing the Frame End Code. For FS and FE synchronization packets, the Short Packet Data Field shall contain a 16-bit frame number. This frame number shall be the same for the FS and FE synchronization packets corresponding to a given frame.

Line Synchronization Packets

Line synchronization packets are optional. For Line Start (LS) and Line End (LE) synchronization packets, the Short Packet Data Field shall contain a 16-bit line number. This line number shall be the same for the LS and LE packets corresponding to a given line.

Frame Blanking and Line Blanking

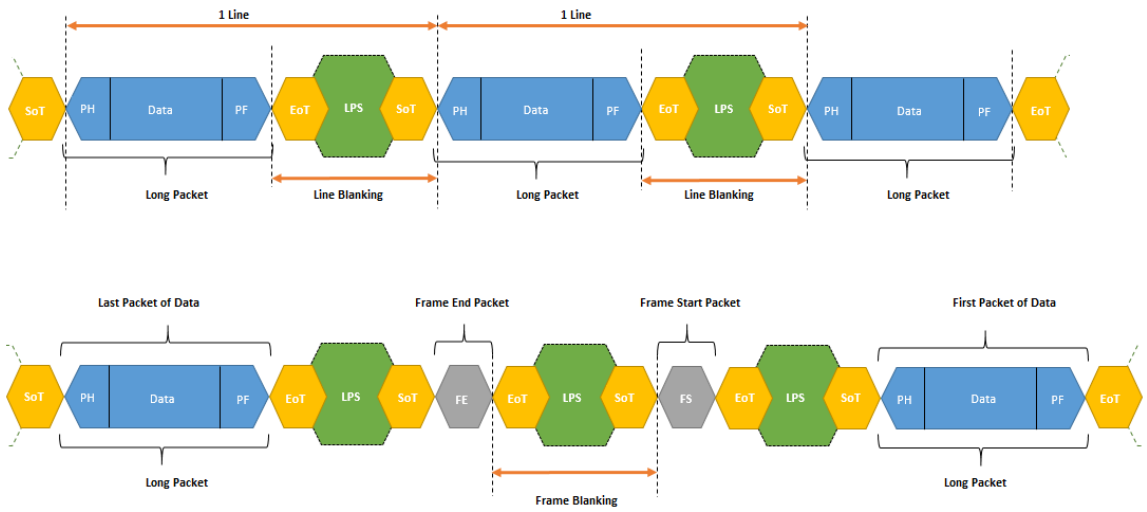


Figure 2 Block Diagram depicting packet structure for CSI sequence

Frame Blanking – The period between the Frame End packet in frame N and the Frame Start packet in frame N+1 is called the Frame Blanking Period.

Line Blanking – The period between the end of the Packet Footer (or the Packet Filler, if present) of one long packet and the Packet Header of the next long packet is called the Line Blanking Period.

Packet Data Payload Size Rules

For YUV, RGB or RAW data types, one long packet shall contain one line of image data. The total size of payload data within a long packet for all data types shall be a multiple of eight bits. The packet payload data format shall agree with the Data Type value in the Packet Header. Refer to *Section 11 Data Formats, Table 3 - Data Type Classes for eight different data type classes* and *Table 8 - Primary and Secondary Data Formats Definitions of the MIPI Alliance Specification for Camera Serial Interface (CSI)*.

To understand the concept of sequences in CSI implementation, consider the block diagram of a video packet shown below.

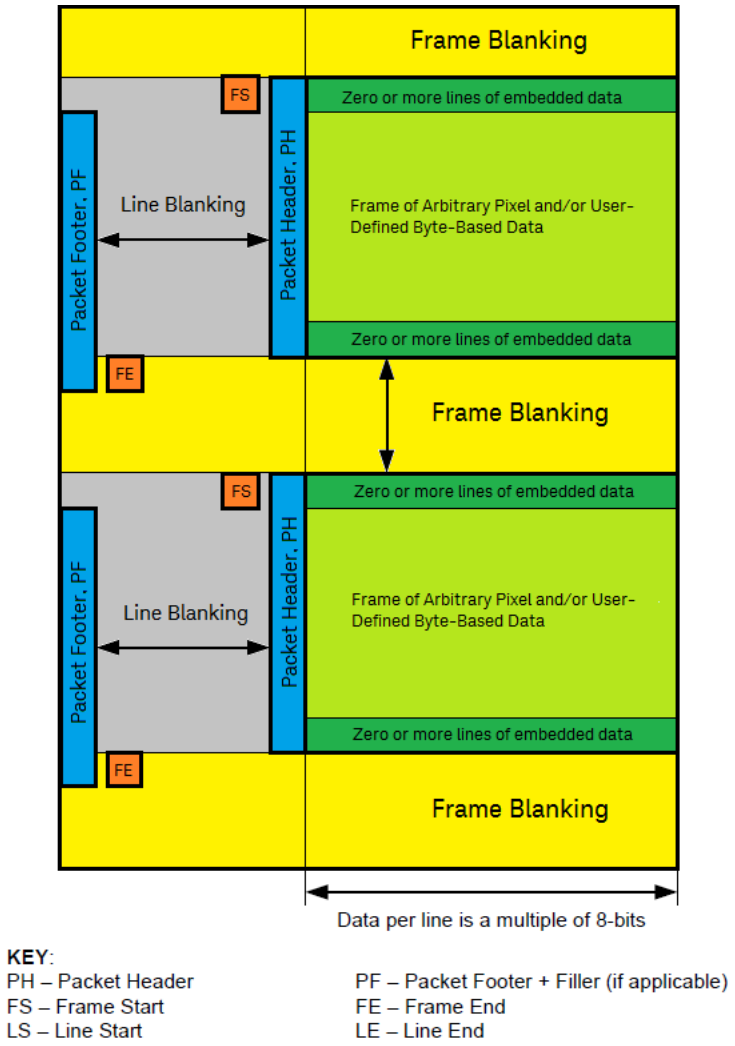


Figure 3 Block Diagram of a CSI Video Frame

For each video line transmission, the short packet or the FrameStart (FS) indicates start of transmission of the video packet. The payload data, which is contained in a long packet, consists of the Packet Header (PH),

followed by the actual arbitrary data and ending with the Packet Footer (PF). Another short packet or the FrameEnd (FE) indicates the end of transmission of the video packet. This data burst is in an HS state.

A Line Blanking (LP state) is transmitted between each video line, that is, after the end of a Packet Footer (PF) till the beginning of the next Packet Header (PH).

A Frame Blanking (LP state) is transmitted between each video frame, that is, after the end of a FrameEnd (FE) till the beginning of the next FrameStart (FS) short packet.

Understanding a CSI sequence file

The description of the block diagram corroborates the structure of the CSI sequence file shown below, for CSI implementation. Note that all the text files defined in the sequence must be placed in the same folder directory where the sequence file is located.

HSFreq: 158 MBit/s;

Blocks:

FrameStart: B"FrameStart.txt",LP11E10880;

Blanking: LP11E10880;

Video: C1E"compliance640_480.txt",LP11E10880;

FrameEnd: B"FrameEnd.txt",LP11E2048;

Sequence:

1. FrameStart,1;

2. Blanking,1;

3. Video,480;

4. Blanking,2;

5. FrameEnd,1;

The sequence definition in the given example contains an intrinsic looping. In the sequence, the blanking lines generate the frame blanking, and the video lines contain the line blanking, which is generated by the LP states. The number of lines for the video data and the associated blanking is defined by the device manufacturer. The sequence begins with the

FrameStart block running once, followed by a Blanking line. Then, the Video block runs for 480 lines ending with another Blanking line, followed by the FrameEnd block running once. This sequence loops over until manually aborted.

If considered closely, the FrameStart block generates the Frame Start (FS) short packet, with the Frame Start Code 0x00 as its header. This follows an HS-LP transition using LP11, where a line blanking is performed with an LPE marker, which fills the LP states until 10880 HS states are attained. The Blanking block generates a frame blanking packet of LP11, which fills the LP states until 10880 HS states are attained and is generated to provide for the LP-HS transition before the actual video payload begins transmitting. In the Video block, the C-Macro with a 1-byte data type of 0x1E generates the Packet Header of the long packet, followed by the actual payload video data (in Hex format) in the compliance640_480.txt file.

Since the video data is High-Speed, the end of the video packet follows an HS to LP transition with a line blanking packet of LP11, which fills the LP states until 10880 HS states are attained. Two blanking lines are sent to indicate the end of the video payload data and to save energy. The FrameEnd block generates the Frame End (FE) short packet, with the Frame End Code 0x01 as its header. This follows an HS-LP transition with a line blanking packet of LP11, which fills the LP states until 2046 HS states are attained.

Some other points to note are:

- Since you cannot send more than one packet per burst, Blanking (LP state) is inserted at the end of each burst to avoid HS data from concatenating.
- The length of the line and frame blanking is device dependent.
- The data type 0x1E corresponds to the pixel color code YUV422 8-bit used in the video. Refer to Section 11 Data Formats of the MIPI Alliance Specification for Camera Serial Interface (CSI) for more information about the other Data Types for various color codes.
- The line blanking length and bits per pixel of a specific color code helps you in determining the total line length in HS states. The E-marker is used for the LP states instead of N, such that it fills up the blocks until the total defined length of HS States is attained.

NOTE

To retain the same line rate for CSI implementation between MIPI D-PHY and MIPI C-PHY, it is recommended that you keep the number of HS states equal in the LP definition.

If all lines in a sequence file definition have an LPE statement in the end, you may calculate the Frame Rate:

1 Multiply the number of lines with the number defined in the LPE statement of the sequence file definition.

2 Repeat step 1 for all lanes in the frame and add the resulting values for each lane.

3 Multiply the sum of all lanes with the HS period length, which derives the Frame Rate.

Calculating HS Data Rate for CSI sequence

To calculate the minimum HS data rate required to run the sequence, you must be aware of at least the frame rate and the device's display resolution, which is provided by the device manufacturer. The HS Frequency is the first line in the definition of a sequence file.

For example, let us consider that the device under test has a Frame Rate of 30 Hz and a display resolution of 640 x 480 pixels, where 640 is the horizontal resolution (or the length of each line) and 480 is the vertical resolution (or the number of video lines).

1 Calculate the line rate using the equation:

Line Rate = Frame Rate * Vertical Resolution

However, the number of video lines has certain number of blanking lines preceding and following the video data, which must be considered as well for data rate calculation. The equation for line rate is, therefore, modified to:

Line Rate = Frame Rate * (Vertical Resolution + no. of blanking lines)

Let us assume that there are 10 blanking lines in a frame.

In this case, Line Rate = 30 Hz * (480 + 10) = 14700 Hz

2 Determine the total length of lines in HS states.

i Determine the number of bits required for transmission of the video data. To do so, check the pixel color coding for the Data Type in the video. In this case, the pixel color code is YUV422, which uses 8-bit per pixel.

Total no. of bits per line = Bit-size per pixel * Horizontal resolution

In this case, Total no. of bits = 8 * 640 = 5120 bits per line.

ii The number of bits for video transmission is not sufficient for determining the total length of lines in HS states, since extra time is required for the LP states in the line blanking. Therefore,

you must consider the LP states and accordingly extend the bits per line. Considering these factors, a total length of 10880 lines in HS states can be safely used for calculation of the HS data rate.

3 Calculate the HS Data Rate using the equation:

Data Rate = Line Rate * Total length of lines in HS state

In this case, Data Rate = 14700 Hz x 10880 = 159.936 Mbps

Therefore, you can define the HS Data Rate (HSFreq) in the beginning of the sequence file, as shown in the example above.

For information on the CSI implementation in the MIPI D-PHY and MIPI C-PHY physical layer and detailed understanding of the protocol layer, refer to the *MIPI Alliance Specification for Camera Serial Interface (CSI)*.

Sequence File Definition for DSI

Overview

DSI specifies the interface between a host processor and a peripheral such as a display module. It builds on existing MIPI Alliance specifications by adopting pixel formats and command set specified in DPI-2, DBI-489 2 and DCS standards. Some significant differences between DSI and CSI are:

- CSI uses unidirectional high-speed Link, whereas DSI is half-duplex bidirectional Link
- CSI makes use of a secondary channel, based on I2C, for control and status functions
- CSI data direction is from peripheral (Camera Module) to host processor, while DSI's primary data direction is from host processor to peripheral (Display Module)
- CSI sequence file structure is different from that of the DSI sequence file structure. The former consists of only of the FrameStart and FrameEnd packet along with some blanking lines, whereas the latter consists of HSync, VSync and Blanking packages.

At the lowest level, DSI protocol specifies the sequence and value of bits and bytes traversing the interface. It specifies how bytes are organized into defined groups called packets. The protocol defines required headers for each packet, and how header information is generated and interpreted.

On the transmitter side of a DSI Link, parallel data, signal events, and commands are converted in the Protocol layer to packets, following the packet organization. The Protocol layer appends packet-protocol information and headers, and then sends complete bytes through the Lane Management layer to the PHY. Packets are serialized by the PHY and sent across the serial Link. The receiver side of a DSI Link performs the converse of the transmitter side, decomposing the packet into parallel data, signal events and commands.

If there are multiple Lanes, the Lane Management layer distributes bytes to separate PHYs, one PHY per Lane, as described in Section 6 of the MIPI Alliance Specification for Display Serial Interface (DSI). Packet protocol and formats are independent of the number of Lanes used. The DSI protocol permits multiple packets to be concatenated, which substantially boosts effective bandwidth. This is useful for events such as peripheral initialization, where many registers may be loaded with separate write commands at system startup. There are two modes of data transmission, HS and LP transmission modes, at the PHY layer. Before an HS transmission can be started, the transmitter PHY issues a SoT sequence to

the receiver. After that, data or command packets can be transmitted in HS mode. Multiple packets may exist within a single HS transmission and the end of transmission is always signaled at the PHY layer using a dedicated EoT sequence. To enhance the overall robustness of the system, DSI defines a dedicated EoT packet (EoTp) at the protocol layer for signaling the end of HS transmission. In HS mode, time gaps between packets shall result in separate HS transmissions for each packet, with a SoT, LPS, and EoT issued by the PHY layer between packets. This constraint does not apply to LP transmissions.

Long and Short Packet Formats

Two packet structures are defined for low-level protocol communication: Long packets and Short packets. For both packet structures, the Data Identifier (DI) is always the first byte of the packet, which includes information specifying the type of the packet.

Long Packet

A Long packet shall consist of three elements: a 32-bit Packet Header (PH), an application-specific Data Payload with a variable number of bytes, and a 16-bit Packet Footer (PF). The Packet Header is further composed of three elements: an 8-bit Data Identifier, a 16-bit Word Count, and 8-bit ECC. The Packet Footer has one element, a 16-bit checksum. Long packets can be from 6 to 65,541 bytes in length.

After the end of the Packet Header, the receiver reads the next Word Count multiplied by the bytes of the Data Payload.

Once the receiver has read the Data Payload it reads the Checksum in the Packet Footer. The host processor shall always calculate and transmit a Checksum in the Packet Footer. Peripherals are not required to calculate a Checksum. Also, note the special case of zero-byte Data Payload: if the payload has length 0, the Checksum calculation results in (0xFFFF). If the Checksum is not calculated, the Packet Footer shall consist of two bytes of all zeros (0x0000). In the generic case, the length of the Data Payload shall be a multiple of bytes. In addition, each data format may impose additional restrictions on the length of the payload data, e.g. multiple of four bytes.

Short Packet

A Short packet shall contain an 8-bit Data ID followed by two command or data bytes and an 8-bit ECC; a Packet Footer shall not be present. Short packets shall be four bytes in length. The Error Correction Code (ECC) byte allows single-bit errors to be corrected and 2-bit errors to be detected in the Short packet. Some short packets may also contain some data in the payload.

Long and Short packets have several common elements. The first byte of any packet is the DI (Data Identifier) byte. The Error Correction Code allows single-bit errors to be corrected and 2-bit errors to be detected in the Packet Header. The host processor shall always calculate and transmit an ECC byte. Peripherals shall support ECC in both forward- and reverse-direction communications.

DSI Sequence Format Description

Sync Event (H Start, H End, V Start, V End), Data Type = XX 0001 (0xX1)

Sync Events are Short packets and, therefore, can time-accurately represent events like the start and end of sync pulses. As “start” and “end” are separate and distinct events, the length of sync pulses, as well as position relative to active pixel data, e.g. front and back porch display timing, may be accurately conveyed to the peripheral. The Sync Events are defined as follows:

- Data Type = 00 0001 (0x01) V Sync Start
- Data Type = 01 0001 (0x11) V Sync End
- Data Type = 10 0001 (0x21) H Sync Start
- Data Type = 11 0001 (0x31) H Sync End

To represent timing information as accurately as possible a V Sync Start event represents the start of the VSA. It also implies an H Sync Start event for the first line of the VSA. Similarly, a V Sync End event implies an H Sync Start event for the last line of the VSA.

Sync events should occur in pairs, Sync Start and Sync End, if accurate pulse-length information must be conveyed. Alternatively, if only a single point (event) in time is required, a single sync event (normally, Sync Start) may be transmitted to the peripheral. Sync events may be concatenated with blanking packets to convey inter-line timing accurately and avoid the overhead of switching between LPS and HS for every event. Display modules that do not need traditional sync/blanking/pixel timing should transmit pixel data in a high-speed burst then put the bus in Low Power Mode, for reduced power consumption.

EoTp, Data Type = 00 1000 (0x08)

This short packet is used for indicating the end of a HS transmission to the data link layer. Therefore, detection of the end of HS transmission may be decoupled from physical layer characteristics. The main objective of the EoTp is to enhance overall robustness of the system during HS transmission mode. Therefore, DSI transmitters should not generate an EoTp when transmitting in LP mode. The Data Link layer of DSI receivers

shall detect and interpret arriving EoTps regardless of transmission mode (HS or LP modes) to decouple itself from the physical layer. Unlike other DSI packets, an EoTp has a fixed format as follows:

- Data Type = DI [5:0] = 0b001000
- Virtual Channel = DI [7:6] = 0b00
- Payload Data [15:0] = 0x0F0F
- ECC [7:0] = 0x01

Blanking Packet (Long), Data Type = 01 1001 (0x19)

A Blanking packet is used to convey blanking timing information in a Long packet. Normally, the packet represents a period between active scan lines of a Video Mode display, where traditional display timing is provided from the host processor to the display module. The blanking period may have Sync Event packets interspersed between blanking segments. Like all packets, the Blanking packet contents shall be an integer number of bytes. Blanking packets may contain arbitrary data as payload.

The Blanking packet consists of the DI byte, a two-byte WC, an ECC byte, a payload of length WC bytes, and a two-byte checksum.

Packed Pixel Stream, 16-bit Format, Long Packet, Data Type 00 1110 (0x0E)

This long packet (shown in the sequence example) is used to transmit image data formatted as 16-bit pixels to a Video Mode display module. The packet consists of the DI byte, a two-byte WC, an ECC byte, a payload of length WC bytes and a two-byte checksum.

Packet Header Error Detection/Correction

The host processor in a DSI-based system shall generate an error-correction code (ECC) and append it to the header of every packet sent to the peripheral. The ECC takes the form of a single byte following the header bytes. The ECC byte shall provide single-bit error correction and 2-bit error detection for the entire Packet Header.

Checksum Generation for Long Packet Payloads

Long packets are comprised of a Packet Header protected by an ECC byte and a payload of 0 to 216-1 bytes. To detect errors in transmission of Long packets, a checksum is calculated over the payload portion of the data packet. Note that, for the special case of a zero-length payload, the 2-byte checksum is set to 0xFFFF.

Checksum generation and transmission is mandatory for host processors sending Long packets to peripherals. It is optional for peripherals transmitting Long packets to the host processor. However, the format of Long packets is fixed; peripherals that do not support checksum generation shall transmit two bytes having value 0x0000 in place of the checksum bytes when sending Long packets to the host processor. The host processor shall disable checksum checking for received Long packets from peripherals that do not support checksum generation.

NOTE **An ECC byte can be applied to both Short and Long packets. Checksum bytes shall only be applied to Long packets.**

Transmission Packet Sequences

DSI supports several formats, or packet sequences, for Video Mode data transmission. The peripheral timing requirements dictate which format is appropriate. In the following sections, Burst Mode refers to time-compression of the RGB pixel (active video) portion of the transmission.

Non-Burst Mode with Sync Pulses

This mode enables the peripheral to accurately reconstruct original video timing, including sync pulse widths. Normally, periods shown as HSA (Horizontal Sync Active), HBP (Horizontal Back Porch) and HFP (Horizontal Front Porch) are filled by Blanking Packets, with lengths (including packet overhead) calculated to match the period specified by the peripheral data sheet. Alternatively, if there is sufficient time to transition from HS to LP mode and back again, a timed interval in LP mode may substitute for a Blanking Packet, thus saving power. During HSA, HBP and HFP periods, the bus should stay in the LP-11 state.

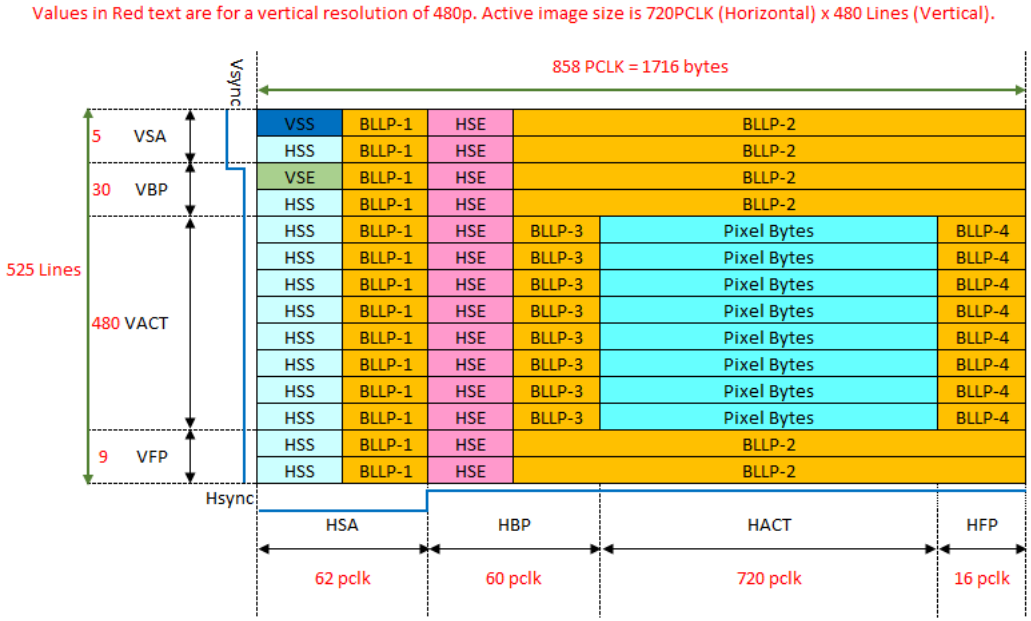


Figure 4 Block Diagram for Video transmission in Non-Burst mode with Sync Pulses

Non-Burst Mode with Sync Events

This mode functions in the same manner as the previous mechanism, but accurate reconstruction of sync pulse widths is not required, so a single Sync Event is substituted. Here, only the start of each synchronization pulse (VSyncSHSyncStart and HSyncStart) is transmitted. The peripheral may regenerate sync pulses as needed from each Sync Event packet received. Pixels are transmitted at the same rate as they would in a corresponding parallel display interface such as DPI-2. As with the previous Non-Burst Mode, if there is sufficient time to transition from HS to LP mode and back again, a timed interval in LP mode may substitute for a Blanking Packet, thus saving power.

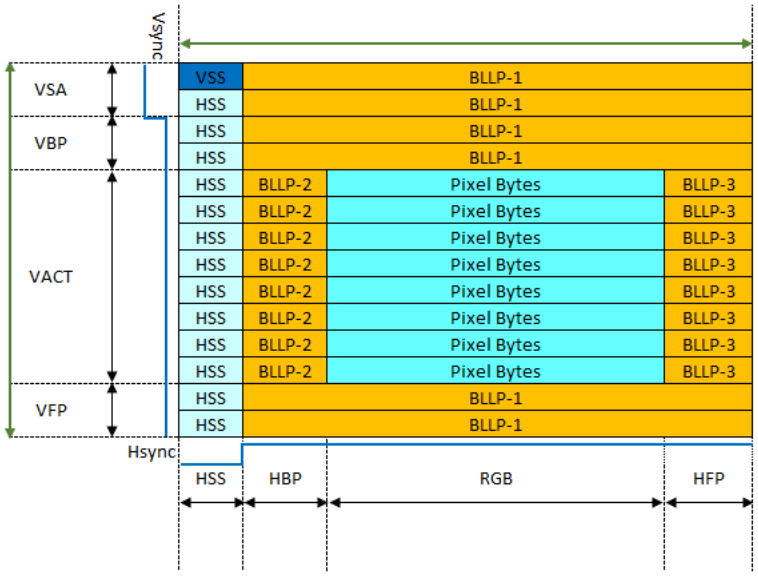


Figure 5 Block Diagram for Video transmission in Non-Burst mode with Sync Events

Burst Mode

RGB pixel packets are time-compressed, leaving more time during a scan line for LP mode (saving power) or for multiplexing other transmissions onto the DSI link. In this mode, blocks of pixel data can be transferred in a shorter time using a time-compressed burst format. This is a good strategy to reduce overall DSI power consumption, as well as enabling larger blocks of time for other data transmissions over the Link in either direction. In the same manner as the Non-Burst Mode scenario, if there is sufficient time to transition from HS to LP mode and back again, a timed interval in LP mode may substitute for a Blanking Packet, thus saving power.

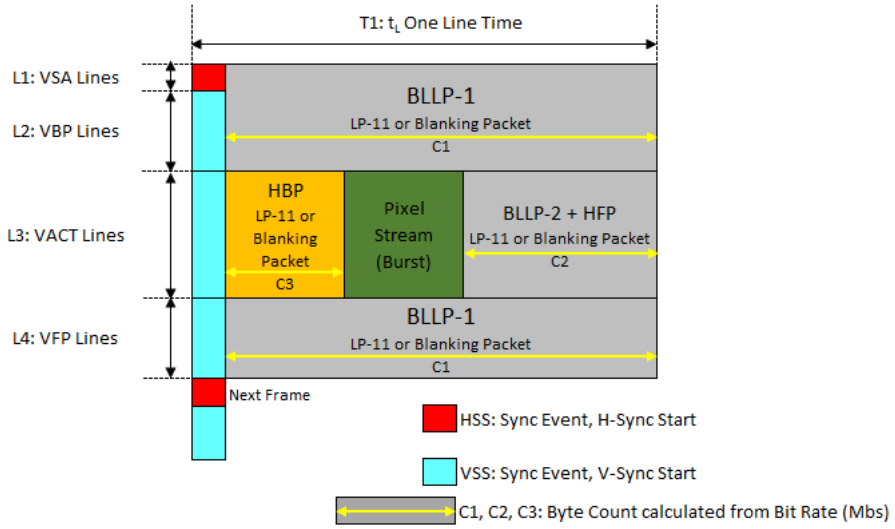


Figure 6 Block Diagram for Video transmission in Burst mode

Note that for accurate reconstruction of timing, packet overhead including Data ID, ECC, and Checksum bytes should be taken into consideration.

To enable PHY synchronization the host processor should periodically end HS transmission and drive the Data Lanes to the LP state. This transition should take place at least once per frame; The host processor should return to LP state once per scan-line during the horizontal blanking time. Regardless of the frequency of BLLP periods, the host processor is responsible for meeting all documented peripheral timing requirements. Note, at lower frequencies BLLP periods will approach, or become, zero, and burst mode will be indistinguishable from non-burst mode.

The sequence of packets within the BLLP or RGB portion of a HS transmission is arbitrary. The host processor may compose any sequence of packets, including iterations, within the limits of the packet format definitions. For all timing cases, the first line of a frame shall start with VSS; all other lines shall start with VSE or HSS. Note that the position of synchronization packets, such as VSS and HSS, in time is of utmost importance since this has a direct impact on the visual performance of the display panel.

Replacing B-Macros with C-Macros in a sequence file

Table 1 shows how a sequence file written originally using the B-Macros can be alternatively written using C-Macros.

Table 1 Sequence file definition using B-Macros and C-Macros

Sequence File Definition using B-Macro	Sequence File Definition using C-Macro
HSFreq: 432.432MBit/s; Blocks: LPInit: LPB"HSyncEnd.txt",LP11E13728; HSync: B"HSyncEnd.txt",LP11E12736,B"HSyncStart.txt",LP11E13728; VSyncStart: B"HSyncEnd.txt",LP11E12736,B"VSyncStart.txt",LP11E13728; VSyncEnd: B"HSyncEnd.txt",LP11E12736,B"VSyncEnd.txt",LP11E13728; Video: B"HSyncEnd.txt",LP11E960,B"Video480pHSyncStart.txt",LP11E13728; Sequence: 1. LPInit,1; 2. HSync,9; 3. VSyncStart,1; 4. HSync,5; 5. VSyncEnd,1; 6. HSync,29; 7. Video,480; LoopTo 2;	HSFreq: 432.432MBit/s; Blocks: LPInit: LPB"HSyncEnd.txt",LP11E13728; HSync: C310000,C080F0F,LP11E12736,C210000,C080F0F,LP11E13728; VSyncStart: C310000,C080F0F,LP11E12736,C010000,C080F0F,LP11E13728; VSyncEnd: C310000,C080F0F,LP11E12736,C110000,C080F0F,LP11E13728; Video: C310000,C080F0F,LP11E960,C0E"Video480p.txt",BL20,C210000,C080F0F ,LP11E13728; Sequence: 1. LPInit,1; 2. HSync,9; 3. VSyncStart,1; 4. HSync,5; 5. VSyncEnd,1; 6. HSync,29; 7. Video,480; LoopTo 2;

Upon considering each block closely, we notice the following differences, otherwise the rest of the sequence definition remains the same.

- In the HSync block, the B“HSyncEnd.txt” is replaced by C310000,C080F0F and B“HSyncStart.txt” is replaced by C210000,C080F0F.
- In the VSyncStart block, the B“HSyncEnd.txt” is replaced by C310000,C080F0F and B“VSyncStart.txt” is replaced by C010000,C080F0F.
- In the VSyncEnd block, the B“HSyncEnd.txt” is replaced by C310000,C080F0F and B“VSyncEnd.txt” is replaced by C110000,C080F0F.
- In the Video block, the B“HSyncEnd.txt” is replaced by C310000,C080F0F and B“Video480pHSyncStart.txt” is replaced by C0E“Video480p.txt”,BL20,C210000,C080F0F.

To understand how the replacements were done, you must read the description given in the earlier sections about the B“<filename>” macro, C<3 hex bytes> and C<1 hex byte>” <filename>” macros.

Let us consider the contents of each text file closely.

Contents of HSyncEnd.txt

3100 0001

080F 0F01

As mentioned in the previous sections, the Data Type for the HSyncEnd signal is 0x31, which means this short packet has its Data ID as 31 and the following two bytes of Packet DATA as 00 00 followed by the ECC of 01.

Therefore, in a sequence, B“HSyncEnd.txt” can be written using the C<3 hex bytes> macro as C310000.

Contents of HSyncStart.txt

2100 0012

080F 0F01

As mentioned in the previous sections, the Data Type for the HSyncStart signal is 0x21, which means this short packet has its Data ID as 21 and the following two bytes of Packet DATA as 00 00 followed by the ECC of 12.

Therefore, in a sequence, B“HSyncStart.txt” can be written using the C<3 hex bytes> macro as C210000.

Contents of VSyncEnd.txt

1100 0014

080F 0F01

As mentioned in the previous sections, the Data Type for the VSyncEnd signal is 0x11, which means this short packet has its Data ID as 11 and the following two bytes of Packet DATA as 00 00 followed by the ECC of 14.

Therefore, in a sequence, B“VSyncEnd.txt” can be written using the C<3 hex bytes> macro as C110000.

Contents of VSyncStart.txt

0100 0007

080F 0F01

As mentioned in the previous sections, the Data Type for the VSyncStart signal is 0x01, which means this short packet has its Data ID as 01 and the following two bytes of Packet DATA as 00 00 followed by the ECC of 07.

Therefore, in a sequence, B"VSyncStart.txt" can be written using the C<3 hex bytes> macro as C010000.

Notice that within each text file considered so far, a hexadecimal value 080F 0F01 is mentioned. This is the EoT package, which is used at end of each HS transmission in a short packet. The EoT package has a fixed format with Data ID as 08, Payload Data as 0F0F and ECC of 01.

Therefore, at the end of each C<3 hex bytes> macro defined for the four files, you must add the C<3 hex bytes> macro for the EoT package as C080F0F, as shown in the sequence definition.

Contents of Video480pHSyncStart.txt

0EA0 0508

1084 1084

1084 1084

1084 1084

.

.

.

1084 1084

FA44 1914

001F 0000

0000 0000

0000 0000

0000 0000

0000 0000

0000 6F1D

2100 0012

080F 0F01

The elements of the actual video data in the file 'Video480pHSyncStart.txt' can be divided as follows:

- The header consists of Data ID: 0E followed by a two-byte word counter A005, followed by the ECC for that header data as 08.
- The actual video payload data starts from a hex value of 1084 and ends at a hex value of 1084. Note that the payload data is too long and has been truncated for documentation purpose.
- At the end of the payload, a two byte CRC is added, which is FA44.
- After the checksum, a Blanking packet is added to covey the blanking timing information in this video packet. This Blanking packet consists of the DI byte of 0x19, a two-byte Word Count of 1400, an ECC byte of 1F, a payload of 20 bytes, and a two-byte checksum of 6F1D. This blanking packet enables the HS-LP transition and corresponds to the HFP (Horizontal Front Porch) before the HSyncStart.
- In the end, a short packet in HS mode is generated, which has the HSyncStart data, which allows the HS-LP transition, before the device goes into a low power state.

Looking back at the Video block, the B“Video480pHSyncStart.txt” is replaced by C0E“Video480p.txt”,BL20,C210000,C080F0F, which indicates that only the actual payload data has been extracted into another text file named Video480p.txt and the rest of the data is appended using the C-Macro to the beginning and to the end of the payload data file. Notice that the Blanking packet with Data ID 0x19, which was part of the initial payload data, has been rewritten using the BL<number of blanking bytes> macros as BL20.

Irrespective of whether you use the B-Macros or the C-Macros, you must ensure that the data is defined in the correct order and you can use the sequence file in either format for waveform generation in the plug-in.

Notice that there has been no change made to the LPInit block of the sequence file, even though the HSyncEnd.txt file is used. This is because even though the HSyncEnd.txt contains hexadecimal data for HS mode, the LPB“filename” macro is used for generating Low Power data specified in the file HSyncEnd.txt, such that the display device is powered on.

To know more about the Data Types required to construct the short or long packets or to define C-Macros in a sequence file, refer to the *MIPI Alliance Specification for Display Serial Interface*.

Understanding a DSI sequence file

The DSI sequence file corresponds to the package structure shown in the image below, which is based on the Non-Burst mode with Sync Pulses:

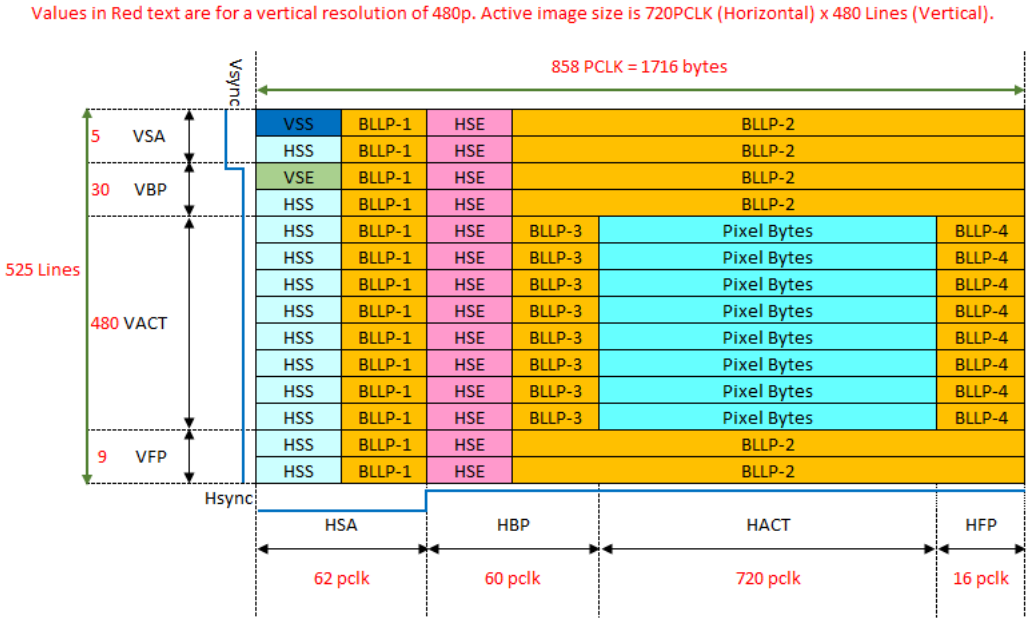


Figure 7 Block Diagram for Video transmission in Non-Burst mode with Sync Pulses

In general, the way the DVI functions is like the CRT mode. In a video frame of the CRT mode, an electron beam performed several horizontal traces and a vertical trace to begin the next frame. In the current technology, the HSS corresponds to the several horizontal traces whereas VSS corresponds to the vertical trace.

The VSyncStart (VSS) and HSyncStart (HSS) pulse are generated to mark the start of the video frame and define the timings of the device. For a video with resolution 720x480p, there are 720 horizontal active pixels, represented by HACT and 480 vertical active lines, represented by VACT. The device manufacturer provides the information about the horizontal blanking pixels and the vertical blanking lines that must be included to meet the timing and power saving requirements of the device, apart from the video transmission. In this case, a total of 138 horizontal blanking pixels are added to the 720 horizontal active pixels. Also, a total of 45

vertical blanking lines are added to the 480 vertical active lines. These HSync and VSync data is required for proper synchronization of the video data and to achieve proper timing in displaying the video data and is represented by HSyncActive (HSA) and VSyncActive (VSA), respectively. As described earlier, HSS, HSE, VSS and VSE are short packets containing High Speed data and blanking header information. They are part of the HSA and VSA.

Sync events are introduced such that only HSS have to be added. In this case, where Sync pulses are used, Blanking Lines can be added, which may be HS or LP, depending on the timing set between HSS and HSE.

The Horizontal Back Porch (HBP), Horizontal Front Porch (HFP), Vertical Back Porch (VBP), Vertical Front Porch (VFP) are indicated by BLLPs. The Blanking or Low-Power Interval (BLLP) is defined as a period during which video packets such as pixel-stream and sync event packets are not actively transmitted to the peripheral. The BLLP provides for the HS-LP and LP-HS transition when the device is powered on, during the switch to video mode and to save the device's power either in the idle state or just before the start of the video data. In this image, the HBP and HFP are indicated by BLLP-3 and BLLP-4, respectively whereas VBP and VFP are indicated by BLLP-1 and BLLP-2, respectively.

Now, let us consider the example given below of a sequence file for DSI implementation, derived from the video package displayed in the image above.

HSFreq: 432.432MBit/s;

Blocks:

LPInit: LPB"HSyncEnd.txt",LP11E13728;

HSync: C310000,C080F0F,LP11E12736,C210000,C080F0F,LP11E13728;

VSyncStart: C310000,C080F0F,LP11E12736,C010000,C080F0F,LP11E13728;

VSyncEnd: C310000,C080F0F,LP11E12736,C110000,C080F0F,LP11E13728;

Video:
C310000,C080F0F,LP11E960,C0E"Video480p.txt",BL20,C210000,C080F0F,LP11E13728;

Sequence:

1. LPInit,1;

2. HSync,9;

3. VSyncStart,1;

4. HSync,5;

5. VSyncEnd,1;

6. HSync,29;

7. Video,480;

LoopTo 2;

The LPInit block generates the HsyncEnd signal is a low power mode to power on the display device.

To achieve proper synchronization and timing for the video data, the HSync and VSync signals are used. The number of vertical blanking lines and horizontal blanking pixels are provided by the device manufacturer.

The HSync block, which loops over for 9 lines, generates the HSE signal, indicated by C310000,C080F0F before the device switches to low power mode until 12736 HS states are attained. The LP11E12736 corresponds to BLLP-2 on the image. Then, HSS signal is generated, indicated by C210000,C080F0F, after which the device enters into low power mode again until 13728 HS states are attained. The LP11E13728 corresponds to BLLP-1 on the image.

The VSyncStart block, which loops over once, generates the HSE signal followed by LP11E12736 state. Then, VSS signal is generated, indicated by C010000,C080F0F, after which the device enters into low power mode again until 13728 HS states are attained.

The HSync block loops over for 5 lines again before the VSyncEnd block loops over once. The VSyncEnd block generates the HSE signal followed by LP11E12736 state. Then, VSE signal is generated, indicated by C110000,C080F0F, after which the device enters into low power mode again until 13728 HS states are attained.

The HSync block loops over for 29 lines again before the Video block loops over for 480 lines. In the Video block, the HSE signal is generated followed by LP11E960 state. Here, in the duration between the Video line and HSyncEnd, there is a short LP state to allow the device to save power before the video starts. LP11E960, which forms the HBP, allows device to save power before switching to video mode. C0E“Video480p.txt” indicates the header information followed by the active payload data contained in the “Video480p.txt” file. The Video block displays BL20, which forms the HFP, before the HSS is generated again followed by switching into low power mode until 13728 HS states are attained.

Note that BLLP-1 and BLLP-2, that is, LP11E12736 and LP11E13728 are blanking lines. Instead of sending video data, long LP states are generated. The total number of HS states is driven by the E marker, which

means each line has a fixed duration, depending on the data rate. It includes the HS states before the LPE marker is defined and the time taken to switch to the HS mode.

You must always ensure that to define a proper sequence, each block must end with the same HS or LP state to avoid any unexpected loops.

Calculating HS Data Rate for the DSI sequence

To calculate the minimum HS data rate required to run the sequence, you must be aware of at least the frame rate, the Blanking Lines (HBlank and VBlank) and the device's display resolution, in other words, the pixels per line and lines per frame, which are provided by the device manufacturer. The HS Frequency is the first line in the definition of a sequence file.

For example, let us consider that the device under test has a Frame Rate of 60 Hz and a display resolution of 640 x 480 pixels, where 640 is the horizontal resolution (or the length of each line) and 480 is the vertical resolution (or the number of video lines). The number of header (blanking lines) is given as 45.

1 Calculate the total no. of lines using the equation:

   Total Lines = Video Lines + Header (Blanking) lines = 480 + 45 = 525 lines

2 Calculate the Line Rate using the equation:

   Line Rate = Frame Rate * Total Lines = 60 Hz x 525 lines = 31.5 kHz

3 Calculate the No. of pixels per line using the equation:

   No. of pixels per line = HBlanking + Horizontal Resolution

   where, the HBlanking data is given by the device manufacturer. Let us assume 160 HBlanking lines.

   Therefore, No. of pixels per line = 160 + 640 = 800 pixels per line.

4 Determine the number of bits required for transmission of the video data. To do so, check the pixel color coding for the Data Type in the video from the device's specification. In this case, the pixel color code is RGB, which uses 24-bits per pixel.

   Calculate the total no. of bits per line using the equation:

   Total no. of bits per line = Bit-size per pixel * No. of pixels per line

   In this case, Total no. of bits per line = 24 * 800 = 19.2 kbits per line

5 Calculate the HS Data Rate using the equation:

   Data Rate = Line Rate * Total no. of bits per line

   In this case, Data Rate = 31.5 kHz * 19.2 kbits per line = 604.8 Mbps

Therefore, for a VGA mode video with a frame rate of 60 Hz, you must set a data rate of 604.8 Mbps, which you can define as HSFreq in the beginning of the sequence file.

For information on the DSI implementation in the MIPI D-PHY and MIPI C-PHY physical layer and detailed understanding of the protocol layer, refer to the *MIPI Alliance Specification for Display Serial Interface*.

3 Using MIPI D-PHY Editor User Interface

Basic Requirements

Hardware Setup using M8195A AWG Module

Single-lane Setup

The required hardware setup for single-lane using M8195A module are following:

- One M8195A module
- An Embedded Controller (such as M9537A) or an external PC
- M9502A AXIe 2-Slot Chassis
- An Infiniium or SCPI compatible oscilloscope
- LAN or GPIB adapter

For hardware setup of a single lane (either Dual Channel or Four Channel mode):

1 Insert an M8195A and an Embedded Controller in the 2 slot frame AXIe chassis as shown in Figure 8.

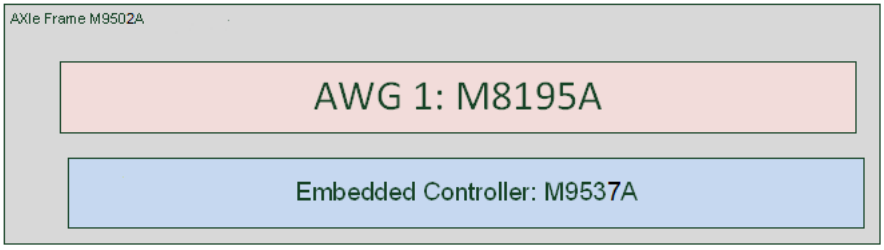


Figure 8 Hardware setup for Single-lane with module M8195A

2 M9502A contains only two slots. The Embedded Controller must be installed in the slot 1 of the AXIe chassis otherwise, it will not be able to connect to the internal PCIe interface of the frame.

3 The slot 2 contains the M8195A AWG module.

Multi-lane Setup

Following hardware setup is required for multi-lane:

- Two or three M8195A AWG modules to enable the multi-lane support (two data lanes in Dual Channel mode and up to four data lanes in Four Channel mode)
- One M8197A module to synchronize multi M8195A modules

- M9505A AXIe 5-Slot Chassis
  - for two lane structure - 2 slots for two M8195A modules, 1 slot for one M8197A module and 1 slot for embedded controller (optional)
  - for three lane structure - 3 slots for three M8195A modules, 1 slot for one M8197A module and 1 slot for embedded controller (optional)

NOTE **M8197A module is used to synchronize M8195A modules (Lane 1, Lane 2, Lane 3 and Lane 4). However, if you are using only single M8195A module (Lane 1), then M8197A is not required.**

Consider Figure 9 to achieve up to three lane hardware setup:

1 Put three M8195A and one M8197A in the 5 slot frame AXIe chassis.

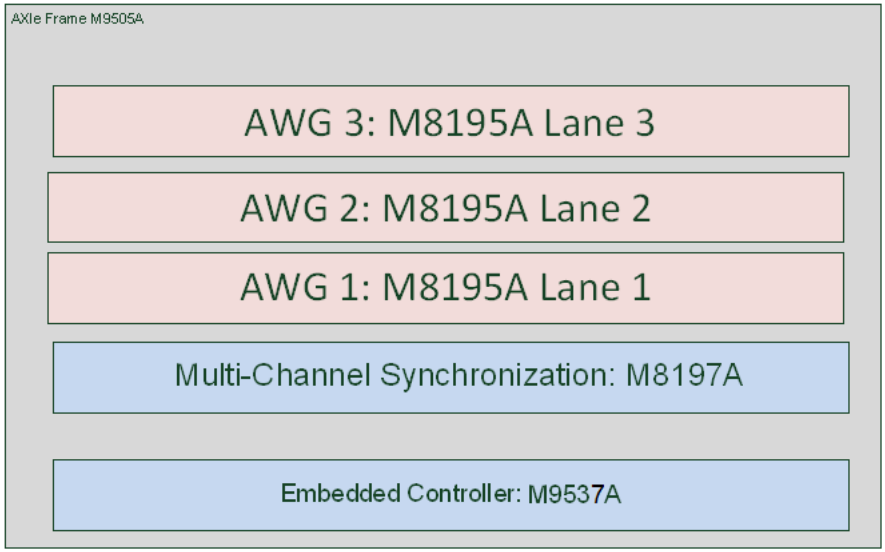


Figure 9 Hardware setup with two or three lane M8195A

2 The modules should be arranged inside a 5 slot chassis in the following order:
   - Slot 1: M8197A (Used for synchronization)
   - Slot 2: M8195A (Used for lane 1)
   - Slot 3: M8195A (Used for lane 2)

- Slot 4: M8195A (Used for lane 3)

NOTE You can also use an embedded controller for M8070B and/or the AWG SFPs or a remote PC as an external controller. The modules for multi-lane M8195A is also de-skewed itself.

Software Requirements

To install the MIPI D-PHY Editor plug-in, the M8070B software (version S6.0.100.2 or later) is required. You can download the software from the following link:

http://www.keysight.com/find/M8070B

License Requirements

The MIPI D-PHY Editor plug-in is a licensed feature. To enable it, Table 2 shows the required licenses:

Table 2 License required for MIPI D-PHY Editor plug-in

P/N	License	Description
M8085DE1A	-1TP	MIPI D-PHY 2.1 Editor for M819xA AWG, Transportable, Perpetual License
	-1NP	MIPI D-PHY 2.1 Editor for M819xA AWG, Network/Floating, Perpetual License
	-TRL	MIPI D-PHY 2.1 Editor for M819xA AWG, 30 Day Trial License
M8085DC1A	-1TP	MIPI D-PHY 2.1 Calibration, Conformance and Characterization Procedures for M819xA AWG, Transportable, Perpetual License
	-1NP	MIPI D-PHY 2.1 Calibration, Conformance and Characterization Procedures for M819xA AWG, Network/Floating, Perpetual License
	-TRL	MIPI D-PHY 2.1 Calibration, Conformance and Characterization Procedures for M819xA AWG, 30 Day Trial License

P/N	License	Description
M8085DUEA	-1TP	Upgrade MIPI D-PHY Editor from M8085A-DT1 to D-PHY 2.1, Transportable, Perpetual License
	-1NP	Upgrade MIPI D-PHY Editor from M8085A-DN1 to D-PHY 2.1, Network/Floating, Perpetual License
M8085DUCA	-1TP	Upgrade MIPI D-PHY Editor plus Calibration, Conformance and Characterization Procedures from M8085A-DT1 and M8085A-DTA to D-PHY 2.1, Transportable, Perpetual License
	-1NP	Upgrade MIPI D-PHY Editor plus Calibration, Conformance and Characterization Procedures from M8085A-DN1 and M8085A-DNA to D-PHY 2.1, Network/Floating, Perpetual License
N5990A	-010	Test Sequencer

Installing Plug-in

The MIPI D-PHY Editor plug-in must be installed separately other than the M8070B system software.

Please make sure that the system should already have M8070B (version 6.0.100.2 or later) software installed on it.

M8195A SFP should be installed from V3.6.0.0 and M8197A SFP should be installed from V3.6.0.0.

The installer for the MIPI D-PHY Editor plug-in is available either on CD or you may download it from the from the following Keysight web-page: www.keysight.com/find/m8085a.

For details on how to install the MIPI D-PHY Editor plug-in, refer to the *Keysight M8085A Plugins for MIPI Receiver Test Solutions Installation Guide*.

Related Documents

The plug-ins are installed separately from the plug-in manager.

For details on how to use the **Plug-in Manager** to install, uninstall and update the **M8085A** plug-in, refer to the *Installing M8085A Plugins for MIPI Receiver Test Solutions* section in *Keysight M8000 Series of BER Test Solutions Plugins for M8070B System Software Getting Started Guide*.

For M8070B plug-ins related documents, click **Start** > **Keysight M8070B** > **Keysight M8070B Documentation**.

Starting the MIPI D-PHY Editor Plug-in

To access the installed MIPI D-PHY Editor plug-in through the M8070B system software:

1. Click **Start** > **Keysight M8070B**. The user interface for the M8070B system software appears.
2. From the M8070B user interface menu, click **Application** to view the list of all installed plug-ins.
3. Select the **MIPI D-PHY Editor** plug-in.
4. The **MIPI D-PHY Editor** user interface appears as shown in Figure 10.

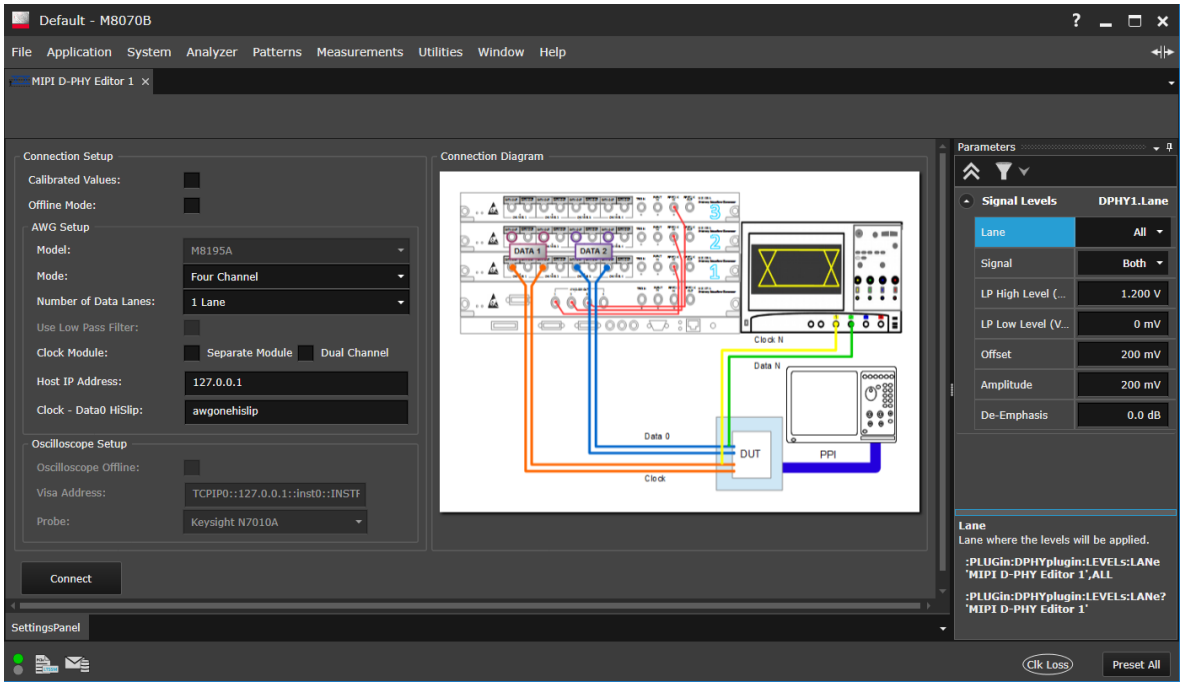


Figure 10 MIPI D-PHY Editor Plug-in User Interface

NOTE

If you are unable to see the MIPI D-PHY Editor option in the **Menu Bar** > **Applications**, you must have to install it separately. Ensure that you have a valid license to install the software.

Currently, the shortcut for Keysight M8085A plug-ins is not auto-generated during installation.

To manually create a shortcut to the Keysight M8070B software on your desktop to access the M8085A plug-ins, perform the following steps:

1 Navigate to *C:\Program Files\Keysight\M8070B\bin* folder on your machine.

2 Right-click *Keysight.M8070B.exe* and click **Create Shortcut**.

3 Click Yes on the 'Shortcut' prompt to place the shortcut on the desktop.

4 On the desktop, right-click *Keysight.M8070B – Shortcut* icon and click **Properties**.

5 In the **Properties** window, modify the 'Target' location to *"C:\Program Files\Keysight\M8070B\bin\Keysight.M8070B.exe" /IgnoreAwg*.

6 Click **Apply** and exit the **Properties** window.

You may launch the *Keysight.M8070B – Shortcut* to access the M8085A plug-ins.

Contacting Keysight Technologies

For more information on products, applications or services associated with Keysight Technologies, contact your local Keysight office. The complete list is available at: www.keysight.com/find/contactus.

MIPI D-PHY Editor User Interface

Beginning with version 2.7 of the M8085A plug-ins, the MIPI D-PHY Editor supports the M8195A AWG configuration only.

M8195A Configuration

The M8195A AWG supports up to 4 Data lanes and uses an M8197A synchronization module for automatic synchronization of the modules and there is no need to connect to an Oscilloscope for synchronization.

The **MIPI D-PHY Editor** user interface includes the following GUI elements:

Connection Setup

The connection setup is the initial step before starting to generate MIPI D-PHY signals. The connection setup shows the connection diagram with instructions on how to connect the system. All the configuration options, to properly set up the system, are available in this panel.

The connection panel provides two special options as following:

- Calibrated Values: Select this option to run the MIPI D-PHY Editor with calibrated parameters such as the signal levels. The calibrated values are available after running some of the calibration procedures that are part of the MIPI D-PHY CTS.
  When this option is not selected or when no calibration data is available the uncalibrated values are used.
- Offline Mode: Select this option to run the MIPI D-PHY Editor without connecting to the real instruments. This is intended to familiarize the user with the MIPI D-PHY Editor User Interface and for preliminary debugging without connecting to real instruments.

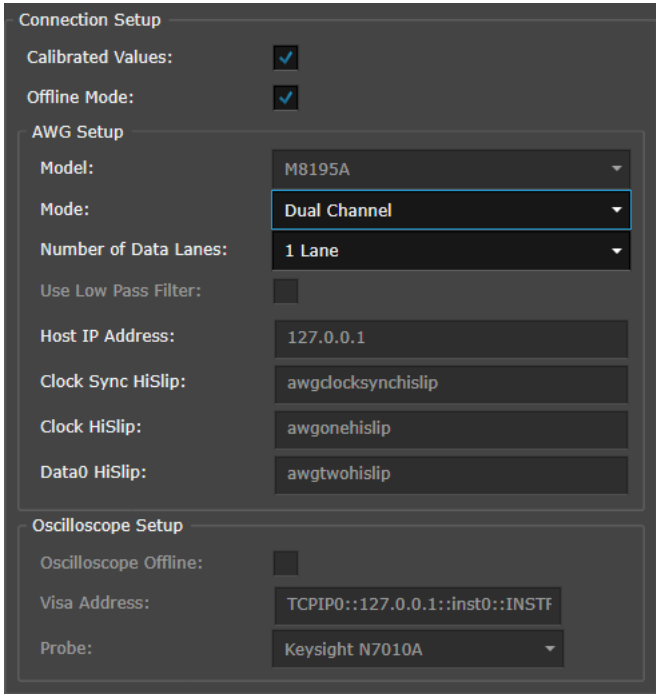


Figure 11 M8195A Connection Setup (for dual channel mode)

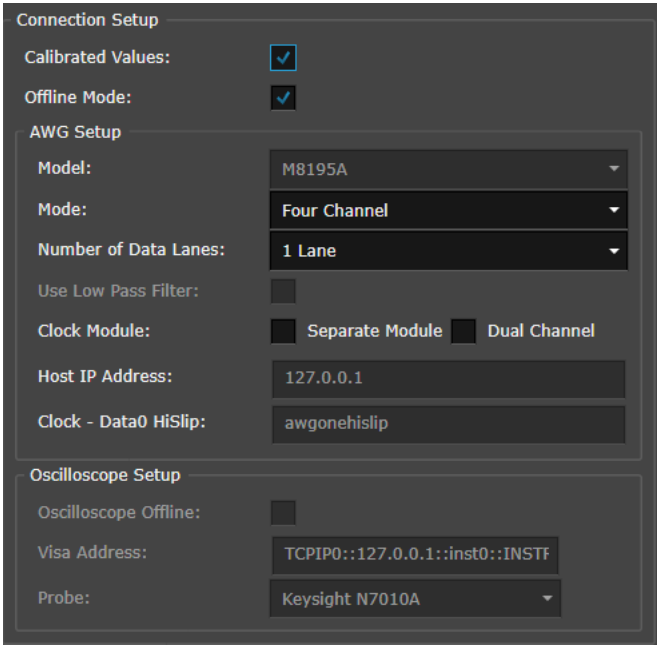


Figure 12 M8195A Connection Setup (for four channel mode)

- AWG Setup
  - Model: By default, M8195A is the only option available.
  - Mode: It provides two options:
    - Dual Channel - The dual channel option uses two channels of the AWG to generate data.
    - Four Channel - The four channel option uses four channels of the AWG to generate data.
  - Number of Data Lanes: Select the number of data lanes used in your connection setup. You can select up to two lanes for dual channel mode and up to 4 data lanes for four channel mode.
  - Clock Module: It has two options:
    - Separate Module: Forces the Clock lane to be generated separately from the Data lanes. Having the Clock lane in a separate module allows to skew the Clock lane in relation to the Data lane without the need to recalculate the waveform.
    - Dual Channel: Allows the clock to connect to the non-inverting pins of two separate channels of the AWG to generate signals.

- Host IP Address: The Host IP address is the network address of the controller computer where the M8195A is connected. It is expected that the IP should be given in the format “127.0.0.1”.
- Clock Sync HiSlip: When more than one lane is selected, the Clock Sync HiSlip identifier needs to be filled. The HiSlip should be given in the format “awgclocksynchislip”.
- Clock HiSlip: This is the HiSlip identifier of the AWG module that generates the MIPI D-PHY Clock lane signal. The value for Clock HiSlip is assigned in the format *hislip<number>*. For example, “hislip0”.
- Data<*m*> HiSlip: (<*m*> = 0 or 1 for dual channel) This is the HiSlip identifier of the AWG module that generates the associated MIPI D-PHY lane signal in dual channel mode. The value for Data0 HiSlip is assigned in the format *hislip<number>*. For example, “hislip0”. Similarly, Data1 HiSlip for “2 Lanes” should be in the same format.
- Data<*m*> - Data<*n*> HiSlip: (<*m*> = 0 or 2 and <*n*> = 1 or 3) This is the HiSlip identifier of the AWG module that generates the associated MIPI D-PHY lane signal in four channel mode.

- Oscilloscope Setup
  - Oscilloscope Offline and Visa Address: For automatic AWG module synchronization, clear the option “Oscilloscope Offline” and type the Oscilloscope VISA Address. For the VISA Address use the format: “TCPIP0::127.0.0.1::inst0::INSTR”.
  - Probe: The calibrations must be performed with the Keysight N7010A active termination adapter.

NOTE

The values for Host IP Address, Clock Sync HiSlip, Clock HiSlip, Data0 HiSlip, Data1 HiSlip, Data2 HiSlip; given above for M8195A, are just examples of the correct format. It is recommended to check and type the correct values for each of these setup characteristics on the respective device at your end.

Once all above mentioned connection settings are done, click the “Connect” button or execute the SCPI command to connect to the instruments. The main user interface appears. For details, refer to the section Main User Interface.

Connection Diagrams

The M8085A MIPI D-PHY Editor is a stand-alone software utility that allows you to set the DUT (Device Under Test) and test configuration. It provides a semi-automatic control of BERT based MIPI D-PHY Editor hardware for physical layer tests, including jitter sources. The MIPI D-PHY Editor is a flexible tool for troubleshooting and debugging. It complements the full Test Automation Software, which provides automated physical layer compliance tests and characterization. The software runs on a standard Windows PC and controls the hardware test resources through appropriate interfaces, for example; LAN.

The following figure represents a typical MIPI D-PHY Editor setup for M8195A for “Dual Channel” mode and it is applicable for all possible number of lanes.

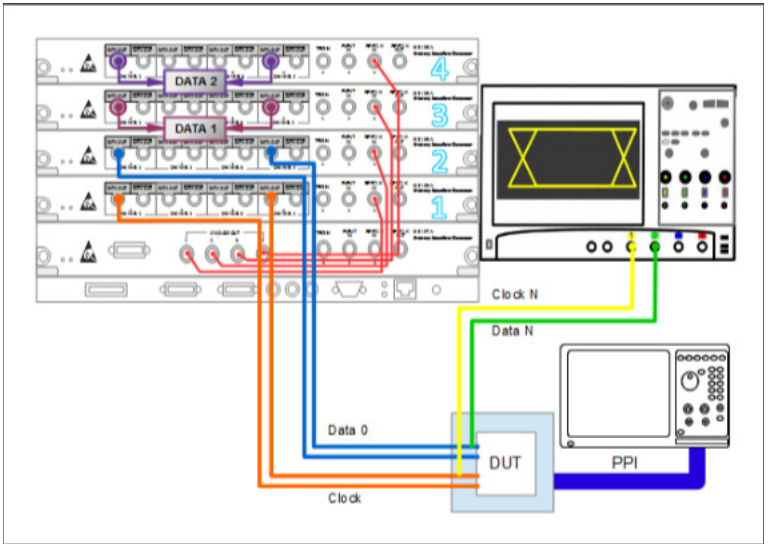


Figure 13 M8085A MIPI D-PHY Editor setup for M8195A (for dual channel mode)

The following figure represents a typical MIPI D-PHY Editor setup for M8195A for “Four Channel” mode with separate Clock module and without Dual Channel:

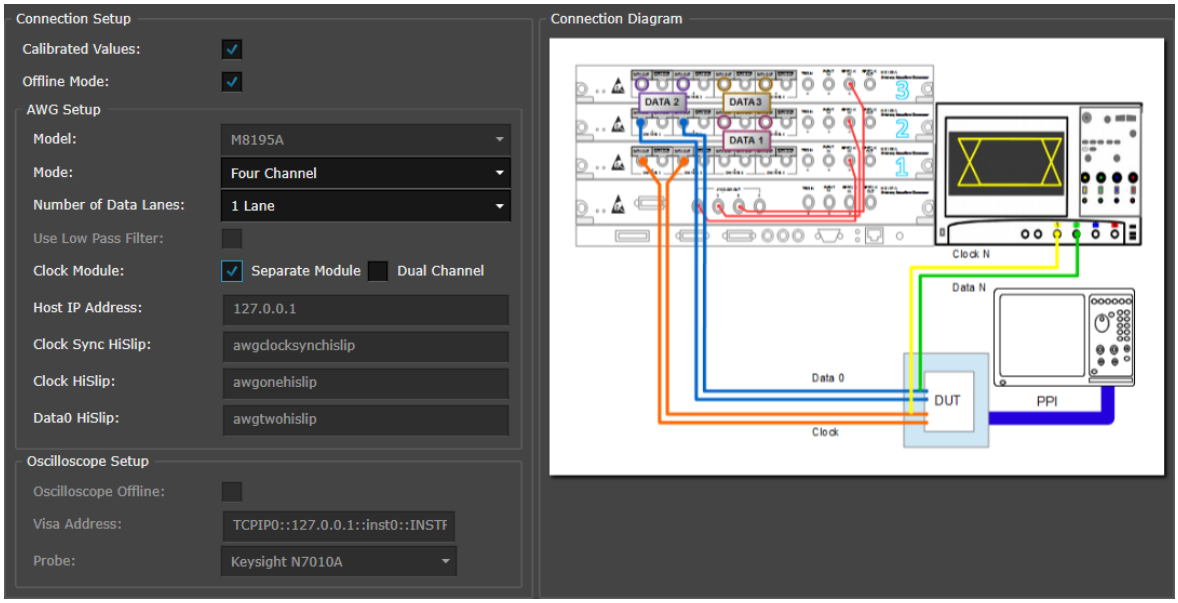


Figure 14 M8085A MIPI D-PHY Editor setup for M8195A (for four channel mode with separate Clock module and without Dual Channel)

NOTE

If you have selected “1 Lane” or “3 Lane” configuration, you will have the choice to select or clear the option “Separate Module”.

But if you have selected “2 Lane” or “4 Lane” configuration, you don’t have the choice to select or clear the option “Separate Module”. By default it is selected always.

The following figure represents a typical MIPI D-PHY Editor setup for M8195A for “Four Channel” mode without separate Clock module and dual channel:

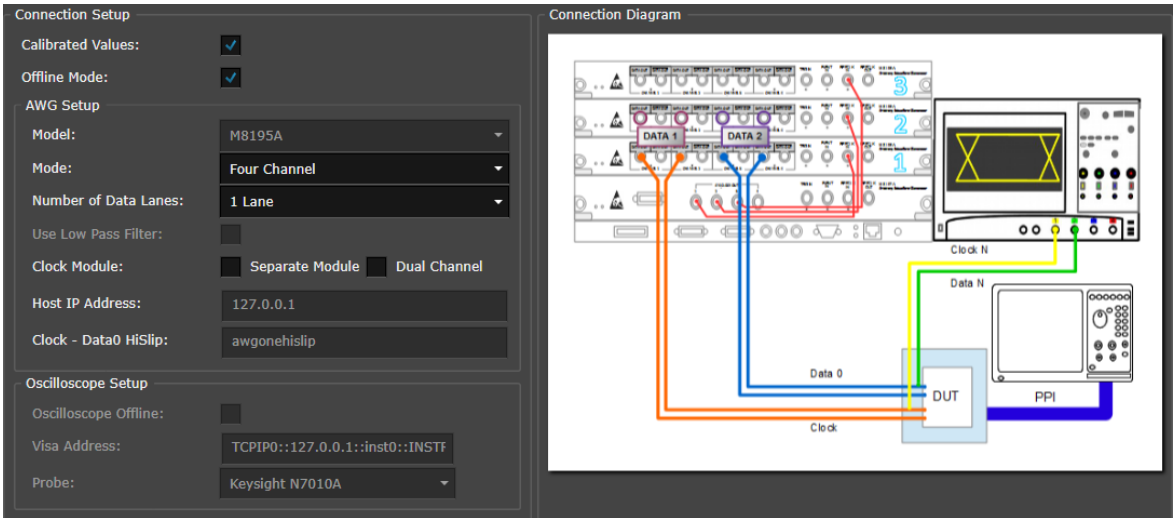


Figure 15 M8085A MIPI D-PHY Editor setup for M8195A (for four channel mode without separate Clock module and Dual Channel)

The following figure represents a typical MIPI D-PHY Editor setup for M8195A for “Four Channel” mode with separate Clock module and dual channel:

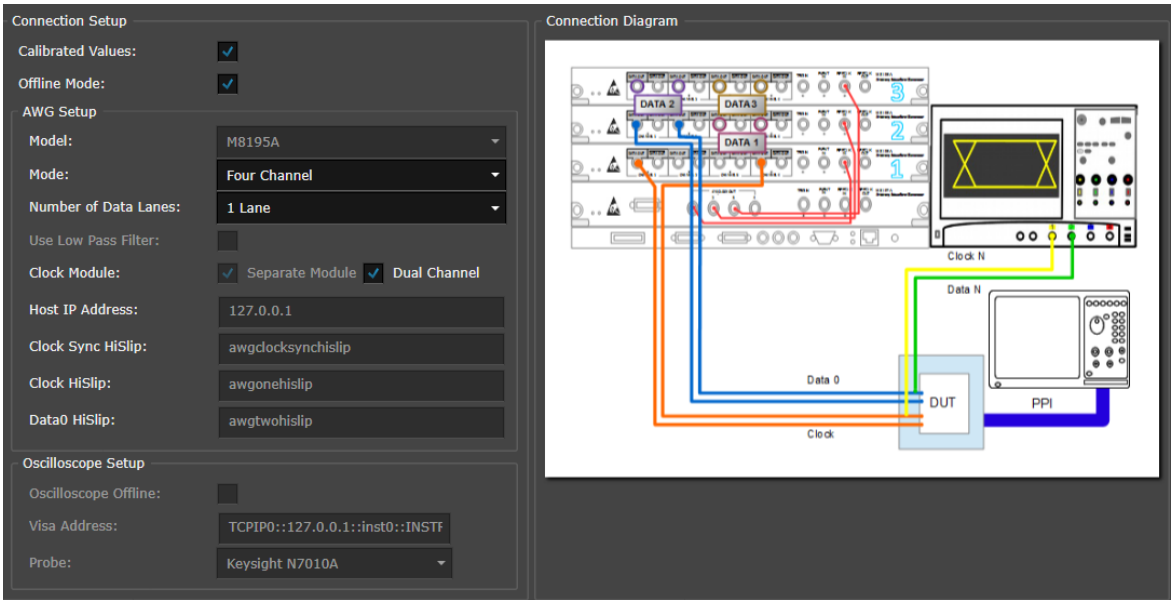


Figure 16 M8085A MIPI D-PHY Editor setup for M8195A (for four channel mode with separate Clock module and Dual Channel)

Main User Interface

The main user interface is shown in Figure 17 for M8195A configuration, and is available after connecting the MIPI D-PHY even in “Offline Mode”. The main user interface contains all the parameters of MIPI D-PHY.

At any moment, it is possible to disconnect from the instruments by clicking the “Disconnect” button.

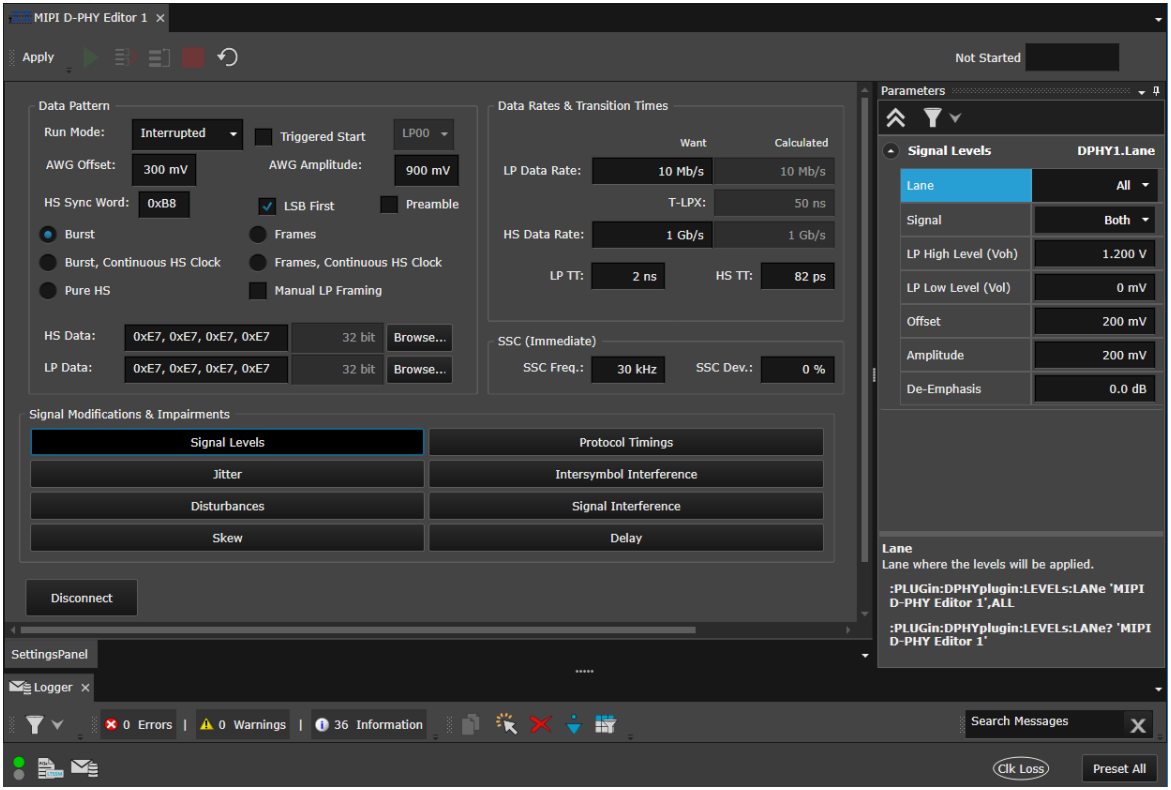


Figure 17 MIPI D-PHY Editor Main User Interface for M8195A

Main user interface consists of the following elements:

1 Toolbar
2 Data Pattern Selection
3 Data Rates and Transition Times
4 Signal Modifications & Impairments

5 Parameters
6 Status Indicator
7 Logger Window
8 SSC (Immediate)

NOTE Other MIPI D-PHY parameters can be accessed by toggling the “Signal Modifications & Impairments” category buttons. When a button is toggled, the corresponding parameters associated with the button category are displayed on the “Parameters” panel.

NOTE The “Parameters” panel is normally present on the right side of the User Interface unless the user has chosen to move it to different region of the User Interface.

Toolbar

The toolbar provides the following shortcuts:

Table 3 Toolbar shortcuts

Icons	Name	Description
	Apply/Abort	The “Apply” command calculates the waveform based on the configured settings. During a waveform calculation, this waveform changes to “Abort”. Pressing “Abort” causes an internal abort of the current calculation and a new waveform calculation starts. After the waveform calculation finishes, the AWGs are put into run mode. A change in the MIPI D-PHY settings does not cause a waveform recalculation. To apply changes in MIPI D-PHY settings apply you need to press “Apply” button.
	Start	The “Start” command puts the AWGs into run mode. This command is available only after a waveform is applied.
	Trigger	The “Trigger” command enables a jump from the initial loop block to next block. This option is available only if a pattern with the option “triggered startup” was programmed into the AWGs.

Icons	Name	Description
	Restart	The “Restart” command stops and starts the AWGs. It does not reprogram any waveform on the AWGs. It is available only if a waveform is applied on the AWGs.
	Stop	The “Stop” command puts the AWGs into stop mode. The command is available only after a waveform is applied and if the AWGs are running.
	Reset	The “Reset” command resets the settings to the default values and calculates the waveform with the default settings.

Data Pattern Selection

The supported data pattern selection options are shown in Figure 18:

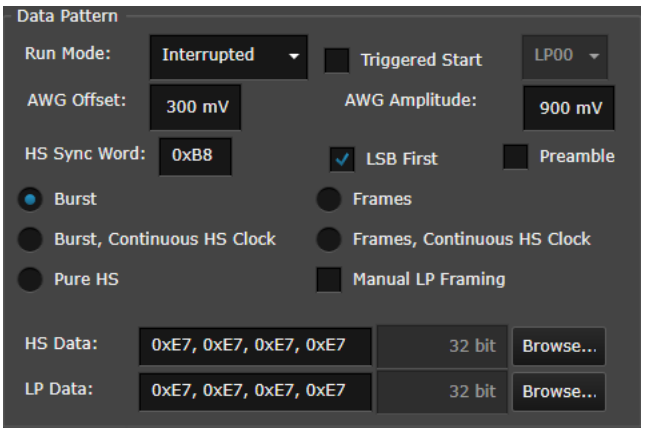


Figure 18 Data Pattern Selection

The MIPI D-PHY Frame Generator software supports two different mode groups—the Burst and the Frame modes. In the Burst Modes (Burst, Burst Continuous Clock, Pure HS), a block of data is repeated infinitely. This block can contain LP Data and HS Data, pure LP data or pure HS data, depending on the content of the data given for HS and LP. If either HS or LP data is empty (an empty text box for that particular data type), the software generates pure LP/HS data. If you select Pure HS, no LP11 transitions are included and all LP data is neglected. The burst modes require two sequences of data; one sequence for HS and another for LP. Each sequence can be set manually or be read by the software from a data file.

The second group of modes are the Frame modes. In these modes (Frames and Frames, Continuous HS Clock), a sequence file runs a sequence of blocks ending with an infinite loop over the complete number of blocks or a selection of the last blocks. Each individual block can be repeated ("looped") N-times and the number of repetitions N can be selected for each block separately. In addition to the data files, the frame modes require a sequence file. It specifies the data rates and the sequence of blocks.

MIPI D-PHY Editor provides the following options:

- Run Mode: The Run Mode enables switching between continuous (ping-pong) mode and interrupted mode.
  - In the Continuous mode, if the waveform calculation has already been performed, when you press "Apply" for the new settings, the AWGs are not stopped. In this case, the AWGs output the previous waveforms. After the new waveform is calculated, the AWGs activate the new waveform, without stopping the AWGs.

NOTE It is mandatory that for some combinations of settings, you must ensure that the AWGs are always stopped (for example, HS Symbol Range change, etc.).

  - In the Interrupted mode, the functionality continues to be the same, that is, applying a new waveform always triggers the AWGs to stop.
- AWG Offset: Sets the offset on the AWG output amplifiers. This is the output offset voltage when the AWGs are in the Stop state.
- AWG Amplitude: Sets the amplitude on the AWG output signal.

NOTE You may customize the AWG Output Amplifier for two modes:

Small Amplitude Mode - By default, AWG Offset is set to 180mV and AWG Amplitude is set to 450mV.

Large Amplitude Mode - By default, AWG Offset is set to 300mV and AWG Amplitude is set to 900mV.

- HS Sync Word: It sets the HS sync word. For LSB first it should be 0xB8, for MSB first it should be 0x1D or binary 00011101. Some devices may not follow the MIPI D-PHY specification and may need different sync words for detecting the start of the HS pattern.

- LSB first: Decides the byte transmission order. When LSB is selected, the least significant bit is sent first.
- Preamble: If HS Data Rate is set to 2.5 Gbps or higher, select this check box to insert a Preamble sequence in each burst of the generated HS data. The Preamble pattern consists of a 1010101010 pattern of $T_{PREAMBLE}$ duration (min. 32 UI to max. 512 UI) and is followed by an Extended Sync Pattern 11111111 (HS-1) in $T_{EXTSYNC}$ (of 8 UI).
- Burst: HS data and clock are embedded in a LP–HS–LP transition. In the Burst mode a block of data is repeated infinitely. This block can contain either LP data, HS data or both, which can be typed directly on the user interface or they can also be loaded by data files with the extension '.dat'.
- Frames and Frames, Continuous HS Clock: Sequences are only available if "Frames" or "Frames, Continuous HS Clock" is selected. For frame data, CSI and DSI sequences can be generated. A sequence file (*.seq) needs to be selected with the "Browse" button. Sequences can also can be used to send an initialization block before the infinite loop. During the data sequence, the clock can either be applied during the HS data transfer ("Frames") or continuously ("Frames, Continuous HS Clock").

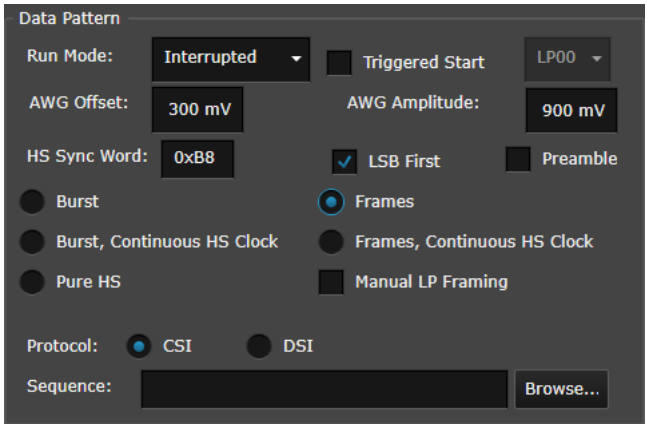


Figure 19 CSI/DSI Sequence File selection with Frames

- Burst, Continuous HS Clock: This is same as Burst data, but continuous HS clock.
- Pure HS: It is the continuous HS data and clock (no HS–LP transitions). Pure HS data can be typed directly on the user interface or they can also be loaded by data files with the extension '.dat'.

- Manual LP Framing: In this mode, no Esc and trigger sequence is added before the LP data and no Mark-1 at the end. It allows full control over the LP levels. In this mode, LP states are allowed in the LP Data text box.
- Triggered Start: The purpose of the “Triggered Startup” is to enable the MIPI D-PHY plug-in to “Triggered Start” mode with a static LP STOP signal.

  To configure a trigger sequence on the MIPI D-PHY Editor user interface,

  - select the “Triggered Start” check-box in the Data Pattern area.
  - select the Low Power state from the drop-down options that must be used in the initial looped block. See Figure 20.

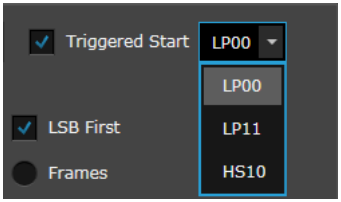


Figure 20 Activating Triggered Start on the user interface

Structure of the Sequence File

For details on the sequence file structures for generating CSI and DSI video frames, respectively, using the MIPI D-PHY Editor plug-in user interface, refer to Chapter 2 Sequence File Definition for CSI and DSI on page 17.

NOTE

MIPI D-PHY Editors cannot use a sequence that contains C-PHY specific macros such as three wire LP states (for example, LP000, LP011, etc.) or the P-Macro (for example, P"Patternfile.ptrn"). Otherwise, both MIPI C-PHY and MIPI D-PHY Editors use the same sequence structure.

Data Rates and Transition Times

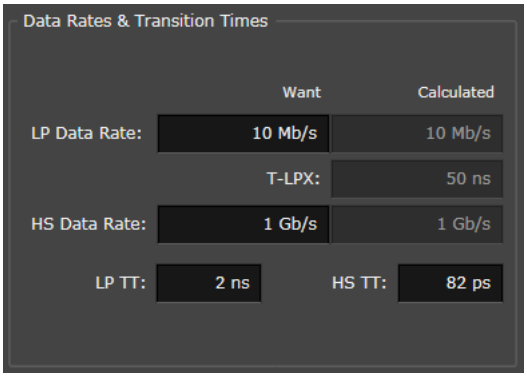


Figure 21 Data Rates and Transition Times.

The Data Rates & Transition Times provide the following options:

- LP Data Rate and HS Symbol Rate: The LP and HS stand for Low Power and High Speed respectively, which defines the data rates. In these fields you can adjust the low power data rate and the high speed symbol rate. However, in some specific situations, the exact data rate or symbol rate cannot be fulfilled. For these situations, you can use the "Calculated" data rates (as shown in Figure 21).

NOTE

Beginning with version 2.7 onwards, the M8085A MIPI D-PHY Editor plug-in supports a minimum LP Data Rate value of 1Mbps. However, using such lower data rate values can impact the pattern calculation time and memory usage. Therefore, the default LP Data Rate is set to 10 Mbps.

- LP TT and HS TT: The LP TT and HS TT stands for Low Power transition time and High Speed transition time respectively. They allow you to adjust the rise and fall time (both) of the Low Power and High Speed modes independently.

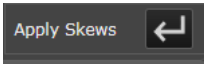
Signal Modifications & Impairments

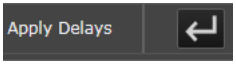
All the parameters of MIPI D-PHY Editor are accessible by the Signal Modification & Impairments panel. By selecting the parameter group from the "Signal Modification & Impairments" panel, the corresponding parameters of the group appears in the "Parameters" panel).

Changing the value of any parameter on the user interface does not perform waveform recalculation automatically anymore. You must explicitly request for a waveform recalculation by clicking the “Apply” button on the user interface.

NOTE

Changing the values of Skews” and “Delays” group parameters are not applied immediately. To apply the made changes

- Press 

button for Skews
- Press 

button for Delays
- You can also press global “Apply” button from the toolbar (for both Skews and Delays).

For M8195A pressing the “Apply” button (or Apply Skews/Delays button), restarts the sequence in all cases.

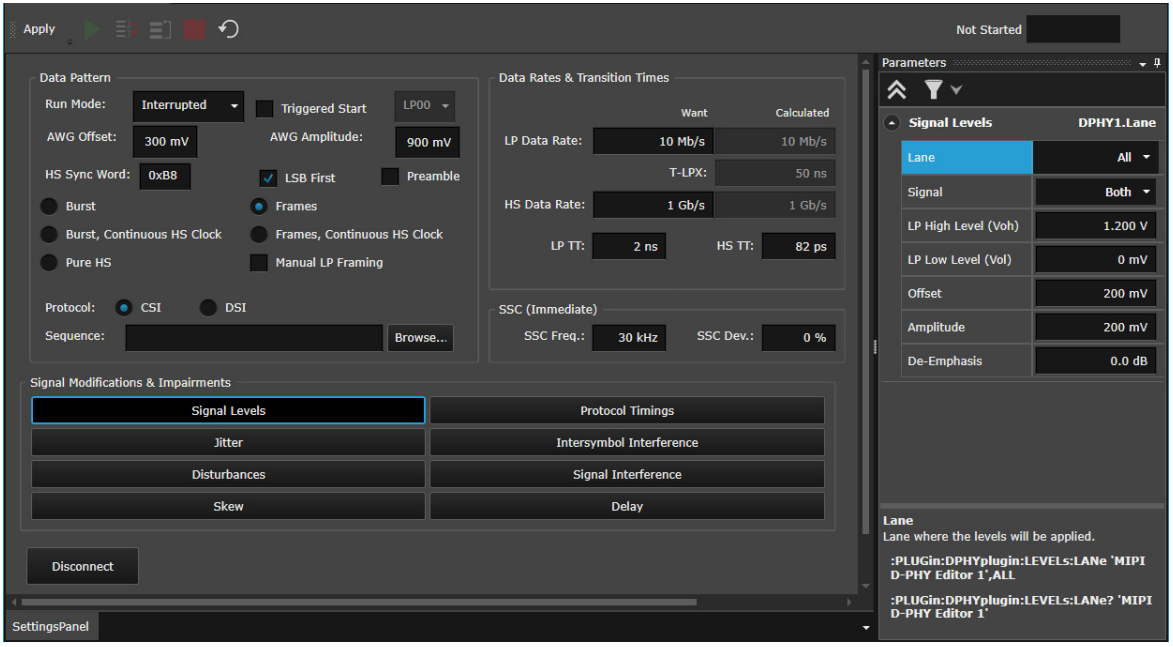


Figure 22 Signal Modifications & Impairments

Parameters

The Parameters window allows you to set the parameters for MIPI D-PHY Editor. It provides the following parameters group:

- Signal Levels Group
- Protocol Timings
- Jitter Group
- Intersymbol Interference Group
- Disturbances Group
- Signal Interference Group
- Skew Group
- Delay Group

For detail description of parameter refer the section Setting up MIPI D-PHY Editor Parameters on page 76.

NOTE To initiate waveform calculation, you must click the “Apply” button on the toolbar. Also, any change in the MIPI D-PHY settings does not trigger waveform calculation automatically anymore. Changes made to the MIPI D-PHY settings are only taken into account after you click the “Apply” button.

Status Indicator

The status indicator shows the current state of a waveform calculation based on the configured settings. It shows the following type of status:

- Not Started: Indicates that the waveform calculation activity is not started.
- Running: Indicates that the waveform calculation activity is currently running.
- Finished: Indicates that the waveform calculation activity is finished and the pattern file is sent.

The following figure shows the status indicator while the waveform calculation activity is in progress:

Figure 23 MIPI D-PHY Status Indicator

Logger Window

The Logger Window displays description of errors, warnings and information messages along with the applications from where they were generated and their time stamps.

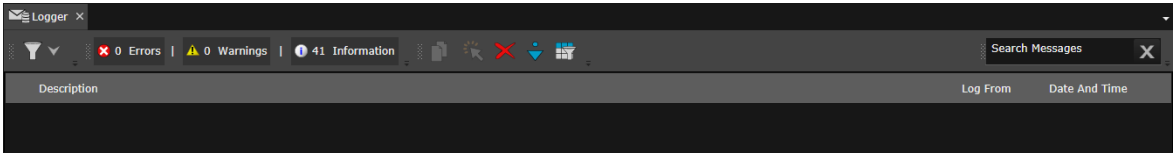


Figure 24 Logger Window

SSC (Immediate)

Modifying the values for the SSC parameters in the main user interface automatically changes the M8195A configuration. During this process, the AWGs automatically stop and restart. You may use the global “Apply” button on the Toolbar to set the SSC values.

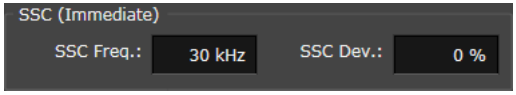


Figure 25 SSC Selection

The SSC feature has the following attributes:

SSC Frequency (SSC Freq.) – Define the value of SSC frequency (in Hertz) between the range from 1KHz to 50KHz. The default value is 30KHz.

SSC Deviation (SSC Dev.) – Define the minimum deviation of the data rate (in percentage) between the range from -0.6% to 0%. The default value is 0%.

Setting up MIPI D-PHY Editor Parameters

To change the value of any double parameters, click the value of the parameter as shown in figure (indicated by number 1). It opens a parameter dialog box, in which you can write the new desired value of parameter within the acceptable range (indicated by number 2).

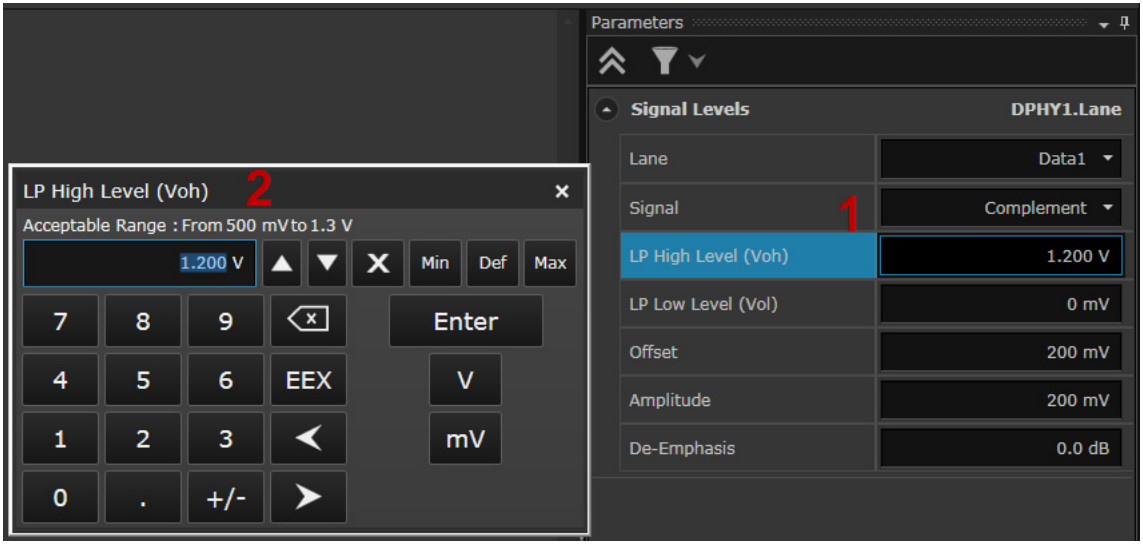


Figure 26 Changing values for Parameters

NOTE If you want to change the value of any parameter, every-time you must explicitly request for a waveform calculation by clicking the “Apply” button on the user interface. The user interface cannot perform waveform calculation automatically.

Signal Levels Group

To change the voltage levels of the signal, click the Signal Levels button in the “Signal Modifications & Impairments” area such that the corresponding settings are visible on the “Parameters panel (see Figure 27).

Other than setting values for Signal Levels in the Parameters panel of the user interface, you may also run the SCPI Command for Signal Levels Group on page 145.

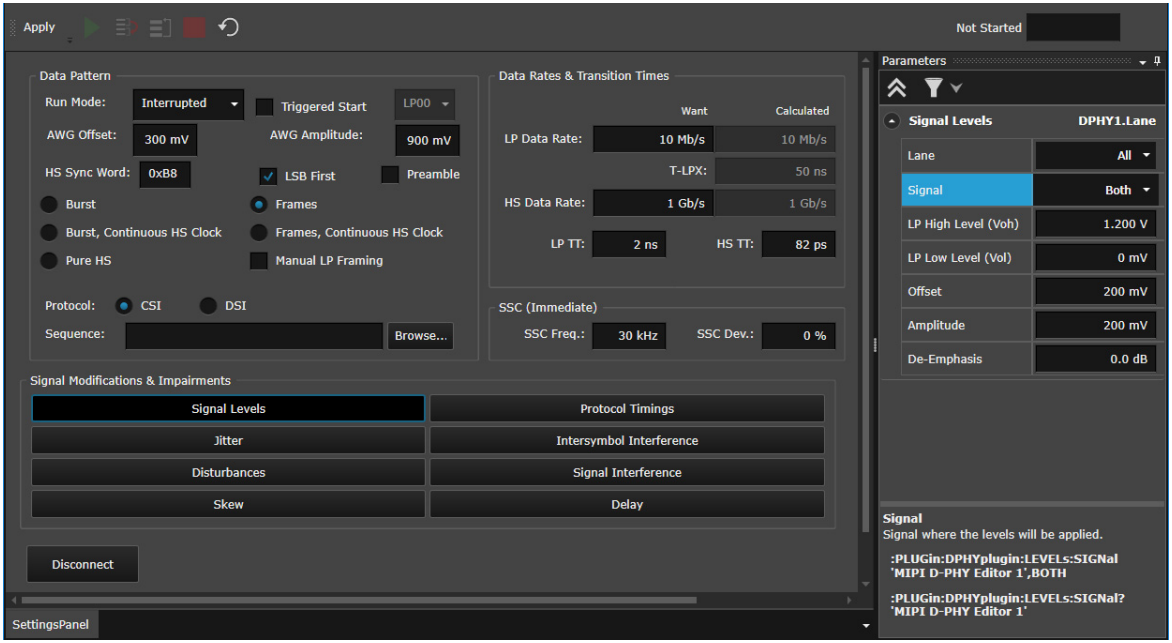


Figure 27 Setting the Signal Levels

Setting different signal levels for lanes

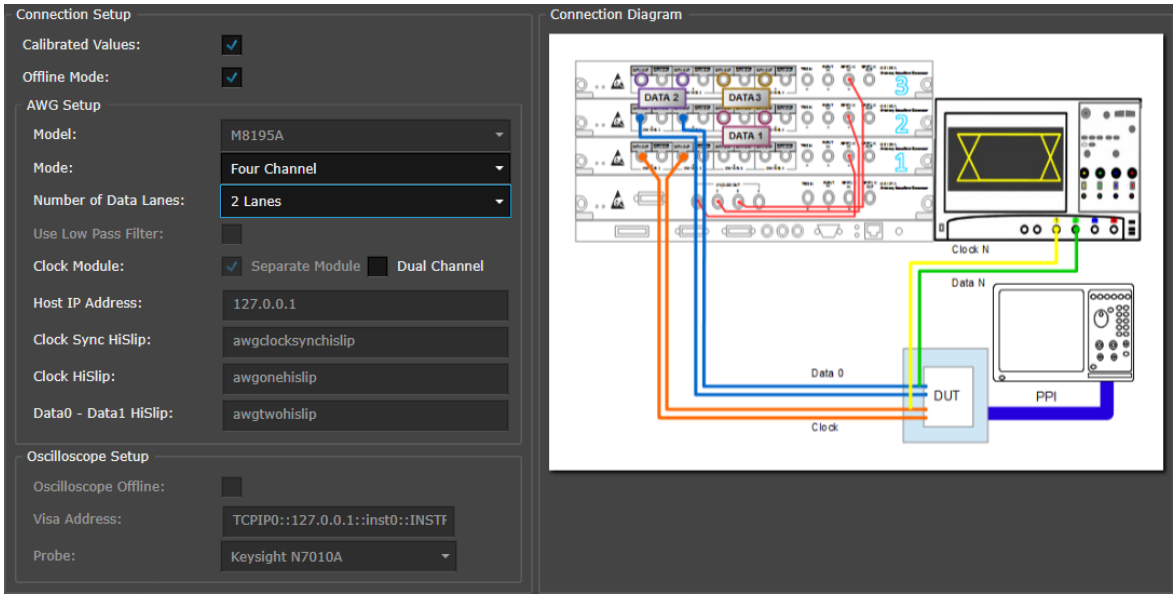
On the MIPI D-PHY Editor user interface, you may modify the voltage level for specific lane while keeping the signal levels for the rest of the data lanes same. This feature is helpful especially when you are running compliance tests on the DUT for M8195A configuration, where you may want to understand the impact of different voltage levels on specific data lanes.

Referring to Figure 27, notice in the “Parameters” panel that by default, common voltage levels are assigned for all lanes in a single instance of the MIPI D-PHY Editor plug-in. That is, with the “Lane” set to “All”, the values set for the signal level settings (Voh, Vol, Offset, Amplitude) are the same for each data lane.

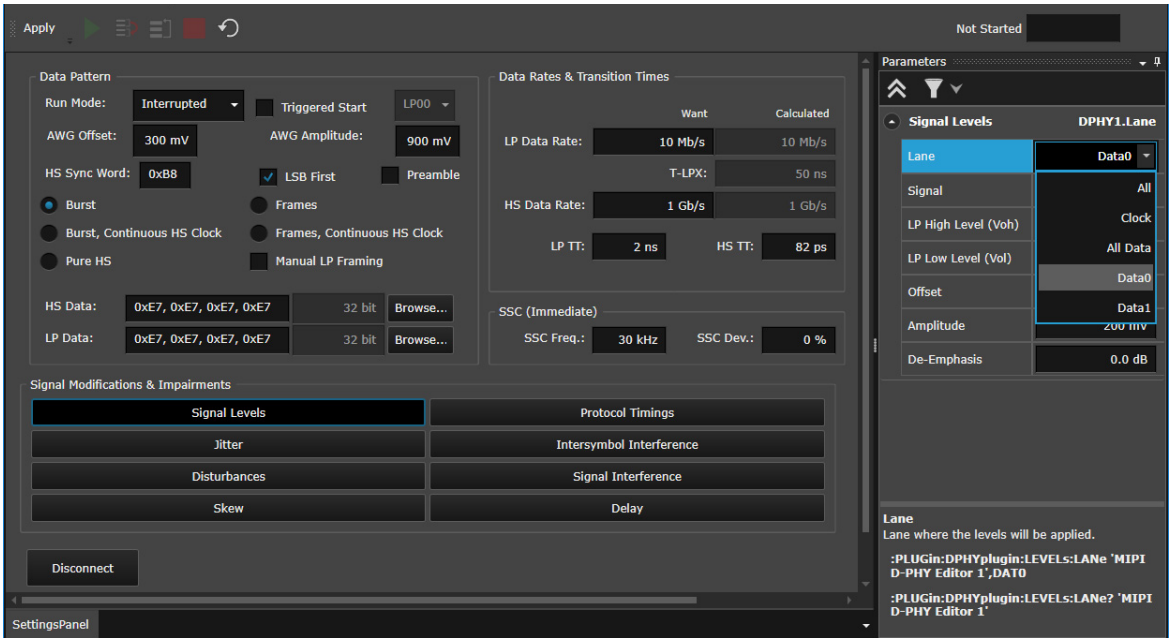
However, for waveform calculation, if required, you can modify the voltage levels for specific Lanes without impacting the voltage levels on other Lanes. The MIPI D-PHY Editor plug-in saves the values for the voltage levels that you modify for a certain specific Lane and does not change them automatically. However, if you perform a waveform calculation for “All” option in Lanes, in the same MIPI D-PHY Editor instance, the modified values for specific lanes are reset to the voltage level values set for the “All” option in Lanes.

To understand this concept in a better manner, consider the following example:

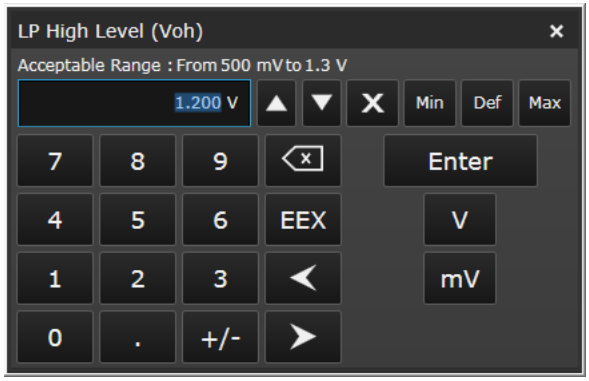
1 In the “Connection Setup” section, connect the MIPI D-PHY Editor plug-in to 2-Lanes of M8195A AWG.

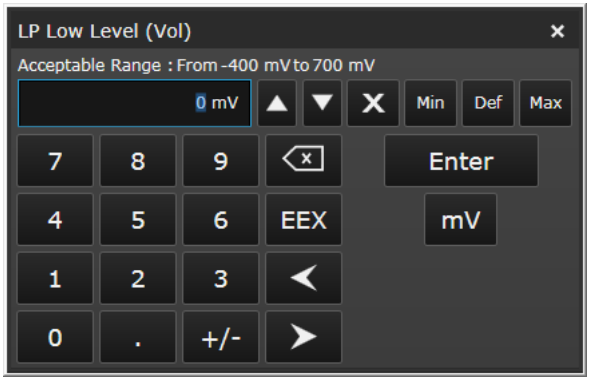


2 In the “Signal Modifications & Impairments” section, select “Signal” Levels and from the drop-down options for “Lane”, select a lane whose signal level you wish to modify. Fer example, you have selected “Data0”.

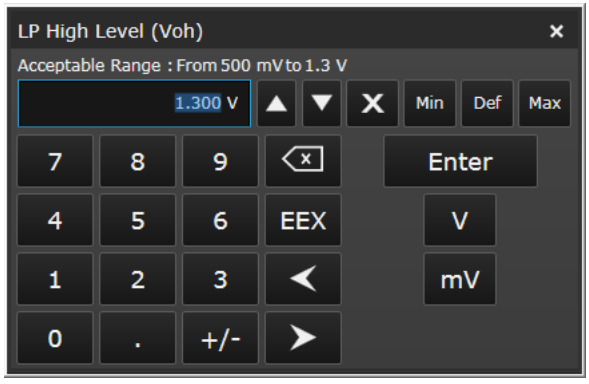


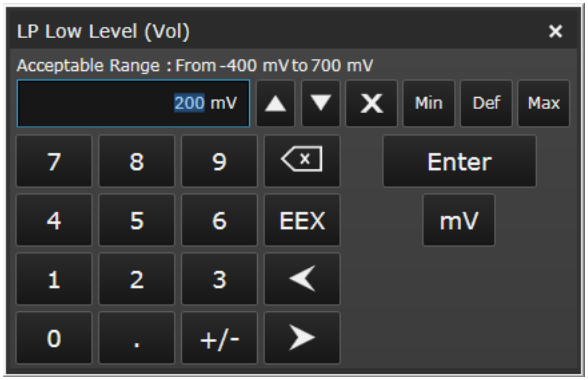
3 Modify one or more signal values as per your requirements. Note that by default, each field contains the compliant values.

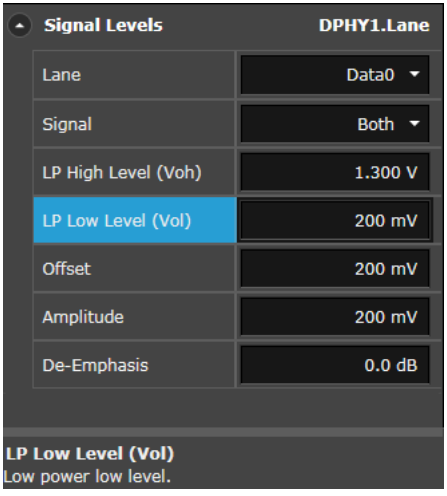




In the following image, the values for “Voh” and “Vol” for Lane “Data0” and have been modified to *1.3V* and *100mV*, respectively.

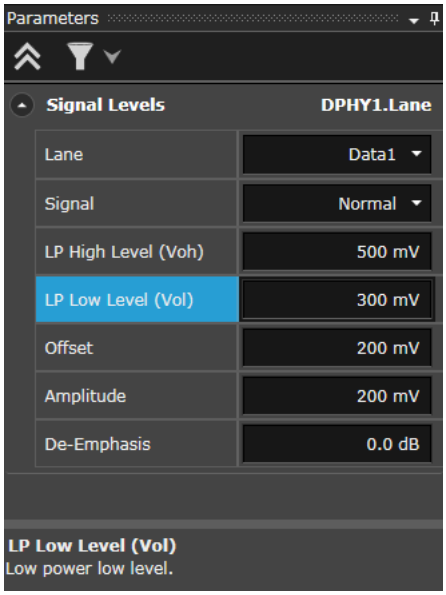






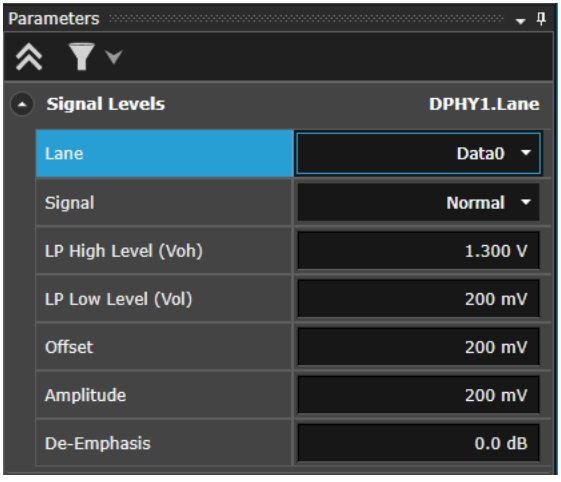
4 Repeat step 2 to 4 if you wish to modify the signal level values for other lanes.

In the following image, the values for “Voh” and “Vol” for Lane “Data1” have been modified to *500 mV* and *300mV*, respectively and “Signal” is set to “Normal”.

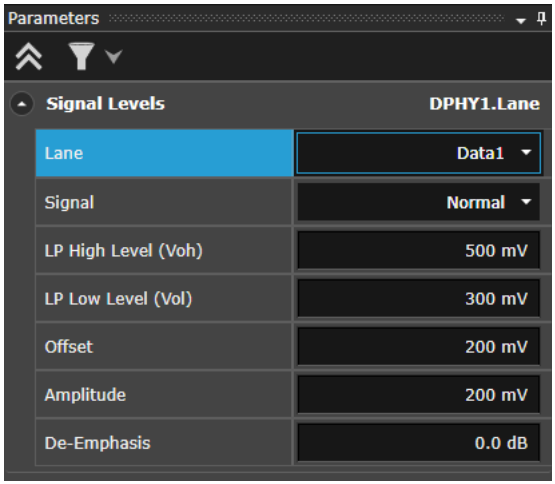


5 Click “Apply” to perform a waveform calculation for the second configuration.

6 Once the waveform generation is complete on Lane “Data1” with “Voh” set to 500mV and “Vol” set to 300mV, change the Lane to “Data0”. Notice that there is no change in the “Voh” and “Vol” values that you had customized for that combination.



7 However, after modifying the Signal Level values for any specific of Lane, if you select “All” for any “Lane” and perform a waveform calculation, the values for specific combinations of lanes are reset to the values set for the option “All” in “Lane”.



8 You may repeat the steps mentioned previously to customize any voltage level under Signal Levels.

The following figure shows the parameters for signal level group:

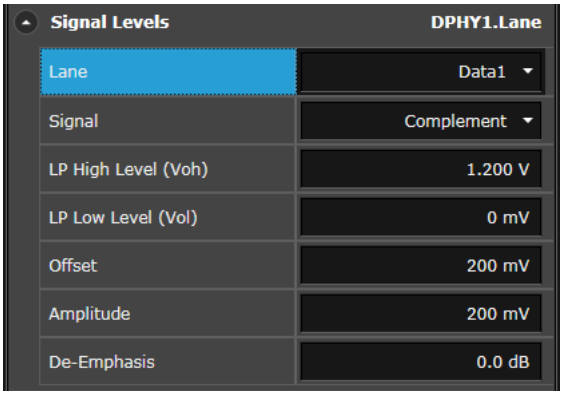


Figure 28 Signal Levels Group

Signal Levels group parameters:

- Lane: Select the lane where you want to apply the voltage levels.
- Signal – Lane's signal pair where the signal voltage levels are applied. It supports on the both Normal and Complement, on the Complement only or on the Normal only.
- LP High Level (Voh) – Low Power mode High voltage level.

NOTE

By default, the LP High Level Voltage parameter is set to 1.200 V, which corresponds to the Large Amplitude Mode. However, if you set the AWG Offset and AWG Amplitude for Small Amplitude Mode, make sure that the LP High Level Voltage parameter is set to 740mV.

- LP Low Level (Vol) – Low Power mode Low voltage level.
- Offset – Sets the offset voltage of the High Speed mode.
- Amplitude – Differential amplitude voltage of the High Speed mode.
- De-Emphasis: Shows the value of de-emphasis in dB. It should be always negative value.

Protocol Timings

To change the protocol settings, click the “Protocol Timings” button in the “Signal Modifications & Impairments” area such that the corresponding settings are visible on the “Parameters” panel. Other than setting values for “Protocol Timings” in the “Parameters” panel of the user interface, you may also run the SCPI Commands for Protocol Timings Group on page 149.

The parameters provided by Protocol setting are shown in Figure 29:

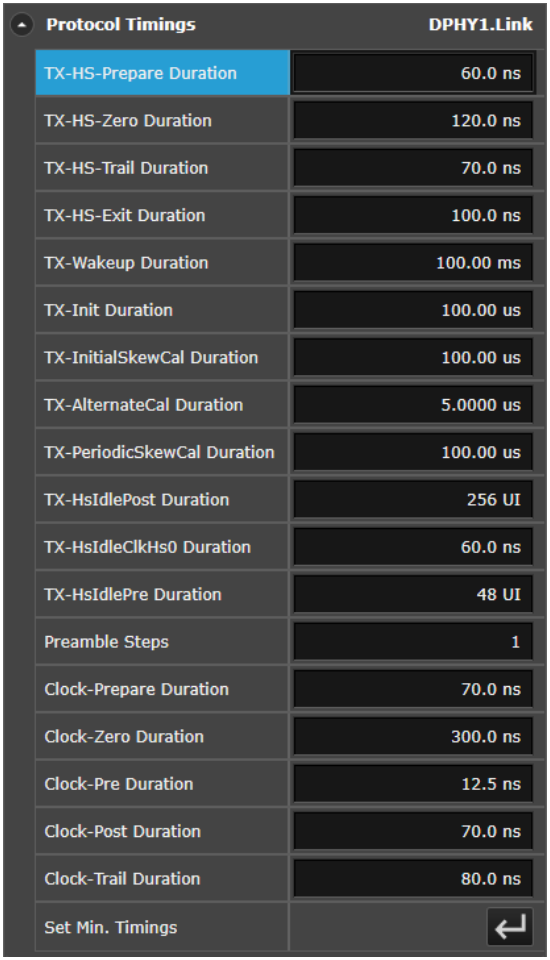


Figure 29 Protocol Timings

Protocol group parameters:

- TX-HS-Prepare Duration: Sets the time that the transmitter drives a LP-00 state immediately before the HS-0 line starts the high speed transmission.
- TX-HS-Zero Duration: Set the time that a transmitter drives HS-0 state until the TX-HS settle time expires in order to neglect transmission effects.
- TX-HS-Trail Duration: Set the time that a transmitter drives flipped differential state after last payload data bit of high speed transmission burst.
- TX-HS-Exit Duration: Set the time that a transmitter drives LP-11 following HS burst.
- TX-Wakeup Duration: Set the time that a transmitter drives a Mark-1 state prior to a Stop state in order to initiate an exit from ULPS.
- TX-Init Duration: Set the time that a transmitter drives a Stop state (LP-11).
- TX-InitialSkewCal Duration: Set the time that a transmitter drives a skew calibration pattern in the initial skew-calibration mode.
- TX-AlternateCal Duration: Set the duration when the transmitting signal drives the Alternate Calibration Pattern in the Alternate Calibration mode. To know more about Alternate Calibration sequence, see Understanding Alternate Calibration Sequence on page 114.
- TX-PeriodicSkewCal Duration: Set the time that a transmitter drives a skew calibration pattern in the periodic skew-calibration mode.
- TX-HsIdlePost Duration: Set the duration when the transmitting signal transitions from Data Burst to HS Idle state.
- TX-HsIdleClkHs0 Duration: Set the duration when the transmitting signal drives in HS-Zero state on Clock lane.
- TX-HsIdlePre Duration: Set the duration when the transmitting signal transitions from HS Idle state to next Data Burst state. To know more about HS-Idle sequence, see Understanding HS-Idle Sequence on page 117.
- Preamble Steps: Set the multiplier value for the required preamble steps to be inserted into the HS Burst. Each step is 32 UI in length. The parameter sets the multiplier value, which is multiplied by 32 UI, and the preamble sequence length is configured. The range for $T_{PREAMBLE}$ is from 32 UI (min.) to 512 UI (max.). To know more about Preamble sequence and its sub-states, see Understanding Preamble Sequence on page 116.
- Clock-Prepare Duration: Set the time that a high speed clock drives a bridge state (LP-00).

- Clock-Zero Duration: Set the time for HS-0 is driven while high speed driver is enabled and low speed driver is enabled.
- Clock-Pre Duration: Set the duration to drive high speed clock data rate UI before any data lane start up. UIs are relative to data channel rate and granularity is 2 as cock has granularity of two “bits”.
- Clock-Post Duration: Set the time that high speed clock is driven in data rate UI after last data lane goes in to low power mode. UIs are relative to data channel rate and granularity is 2 as clock has granularity of two “bits”.
- Clock-Miss Duration: Set the timeout for the receiver to detect absence of clock transition and disable the clock Lane HS-RX.
- Clock-Trail Duration: Set the time that a transmitter drives HS-0 state after the last payload clock bit of high speed transmission burst.
- Set Min. Timings: Set the minimum timing based on current data rates.

Jitter Group

To change the jitter settings, click the “Jitter” button in the “Signal Modifications & Impairments” area such that the corresponding settings are visible on the “Parameters” panel. Other than setting values for “Jitter” in the “Parameters” panel of the user interface, you may also run the SCPI Commands for Jitter Group on page 158.

The Jitter group contains the parameters to generate Sinusoidal Jitter and Bounded Uncorrelated Jitter (Random Jitter). For both types of jitter, it is possible to select the target lane were it needs to be generated.

The following figure shows parameters for jitter group:

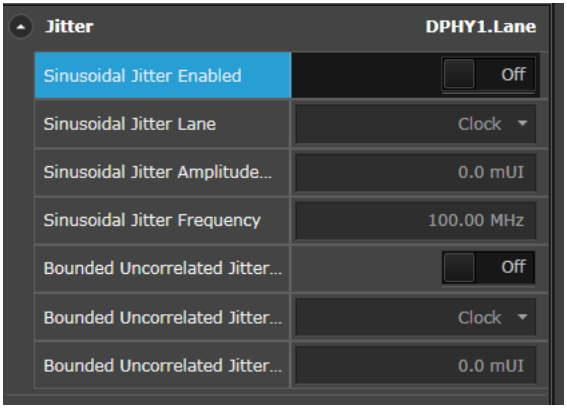


Figure 30 Jitter Group

Jitter Group Parameters:

- Sinusoidal Jitter Enabled: Enables or disables the Sinusoidal Jitter generation on all lanes.
- Sinusoidal Jitter Lane: Lane selection where the Sinusoidal Jitter is applied. It supports Sinusoidal Jitter on the clock lane, on all data lanes or alternatively on a specific data lane.
- Sinusoidal Jitter Amplitude (p-p): Amplitude of the Sinusoidal Jitter in peak-to-peak.
- Sinusoidal Jitter Frequency: Frequency of the Sinusoidal Jitter.
- Bounded Uncorrelated Jitter Enabled: Enables or disables the Bounded Uncorrelated Jitter generation on all lanes. It emulates random jitter.
- Bounded Uncorrelated Jitter Lane: Lane selection where the Bounded Uncorrelated Jitter is applied. It supports Bounded Uncorrelated Jitter on the clock lane, on all data lanes or alternatively on a specific data lane.
- Bounded Uncorrelated Jitter (RMS): Root Mean Square (RMS) Amplitude of the Bounded Uncorrelated Jitter.

Intersymbol Interference Group

To change the intersymbol interference settings, click the “Intersymbol Interference” button in the “Signal Modifications & Impairments” area such that the corresponding settings are visible on the “Parameters” panel. Other than setting values for “Intersymbol Interference” in the “Parameters” panel of the user interface, you may also run the SCPI Commands for Intersymbol Interference Group on page 162.

Intersymbol Interference can also be generated by the MIPI D-PHY Editor plug-in by using an S-Parameter file that modulates the targeted physical channel. After selecting an S-Parameter file some additional changes are allowed.

The following figure shows parameters for intersymbol interference group:

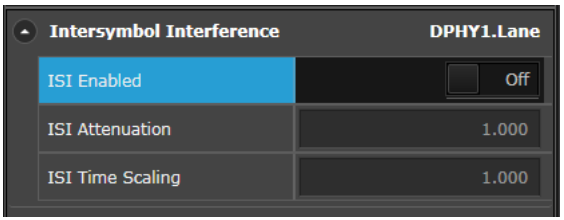


Figure 31 Intersymbol Interference Group

Intersymbol Interference Group Parameters:

- ISI Enabled: Enables or disables the Intersymbol Interference.
- ISI Attenuation: Allows to increase or decrease the overall attenuation of the S-Parameters contained on the S-Parameter file.
- ISI Time Scaling: Allows to change the time scale of S-Parameters contained on the S-Parameter file. A change here will cause in increasing or decreasing the ISI value.

Disturbances Group

To change the disturbances settings, click the “Disturbances” button in the “Signal Modifications & Impairments” area such that the corresponding settings are visible on the “Parameters” panel. Other than setting values for “Disturbances” in the “Parameters” panel of the user interface, you may also run the SCPI Commands for Disturbances Group on page 164.

The Disturbances Group parameters allow you to insert the e_{SPIKE} in the middle of a symbol with editable area and input voltages of logic 1(or 0). Also, these parameters allow you to modify the LP Pulse Width and the pattern disturbances on the high-speed entry and exit sequences of a burst. To clearly understand the functionality of the various parameters under the Disturbances group, a conceptual understanding of e-Spikes is required.

eSpike

Spikes are defined as transient signals, clearly distinguishable from the background activity with a pointed peak. The eSpike is described as an LP (low power) receiver’s ability to reject short-term glitches, i.e., narrow pulses with voltage levels outside of the current Logic state, but that must not change the receiver state, as their widths are sufficiently shorter than the nominal T_{LPX} interval.

The LP receiver should reject any eSpike up to the limit defined in the specification.

Activating eSpike generation introduces a single voltage spike per LP state. It is located in the center of the LP-symbol and has an area as defined in the corresponding property. Area calculation of eSpikes also depends on the receiver high/low level thresholds called “Logic 1 Input Voltage” and “Logic 0 Input Voltage” respectively. When having eSpikes in logic 1 states the glitch will go from LP high level to LP low level. As area calculation of the eSpike in a high state begins below the selected “Logic 1 Input Voltage” the width of the eSpike increases when increasing this

value as it will also be the case when increasing the area directly. “Logic 0 Input Voltage” has no effect on high level eSpikes. Behavior of the eSpikes in low levels is similar to the high level case with the difference that “Logic 0 Input Voltage” is then a width defining parameter. During LP <-> HS transitions “TX-HS-Request Duration”, “TX-HS-Prepare Duration” and “TX-HS-Exit Duration” are treated as a single LP symbol and therefore only contain a single eSpike each.

The conformance limit for eSpike is defined in units of Volts times picoseconds, (V*ps), which allows for multiple potential test cases (high voltage/short-duration, vs. low voltage/long duration).

The following diagram shows eSpike glitches.

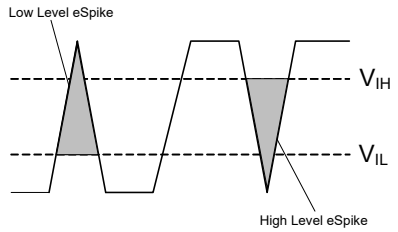


Figure 32 Input Glitch Rejection of Low-Power Receivers

The following table shows the LP Receiver DC specifications:

Table 4 LP Receiver DC Specifications.

Parameter	Description	Min	Nom	Max	Units	Notes
V_{IH}	Logic 1 input voltage	880			mV	1
		740			mV	2
V_{IL}	Logic 0 input voltage, not in ULP State			550	mV	
$V_{IL-ULPS}$	Logic 0 input voltage, ULP State			300	mV	

Note:
1. Applicable when the supported data rate <= 1.5 Gbps.
2. Applicable when the supported data rate > 1.5 Gbps.

The following table shows the LP Receiver AC specifications:

Table 5 LP Receiver AC Specifications.

Parameter	Description	Min	Nom	Max	Units	Notes
eSpike	Input pulse rejection			300	V.ps	1, 2, 3
T_{MIN-RX}	Minimum pulse width response	20			ns	4

The Low power receivers must have an ability to reject short-term glitches (any input signal smaller than e_{SPIKE}), which are the narrow pulses with voltage levels outside of the current logic state. However, the receiver state does not change as their widths are sufficiently shorter than the nominal T_{LPX} interval as shown in Figure 33.

Activating e_{SPIKE} generation introduces a single voltage spike per LP state. It is located in the center of the LP-state. Area calculation of e_{SPIKE} depends on the receiver high/low level thresholds called “Logic 1 Input. Voltage” and “Logic 0 Input Voltage” respectively.

The following figure shows e_{SPIKE} glitches where V_{IL} and V_{IH} represent e_{SPIKE} Logic 0 Input Voltage and e_{SPIKE} Logic 1 Input Voltage, respectively.

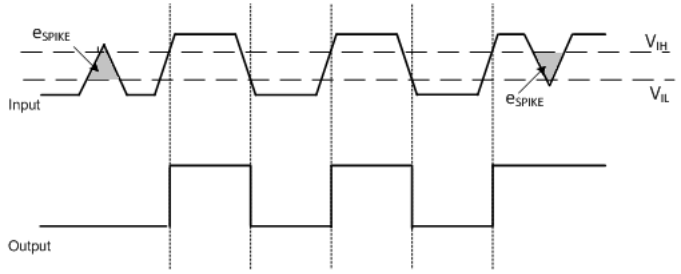


Figure 33 Input Glitch Rejection of Low-Power Receiver

The following figure shows parameters for Disturbances group:

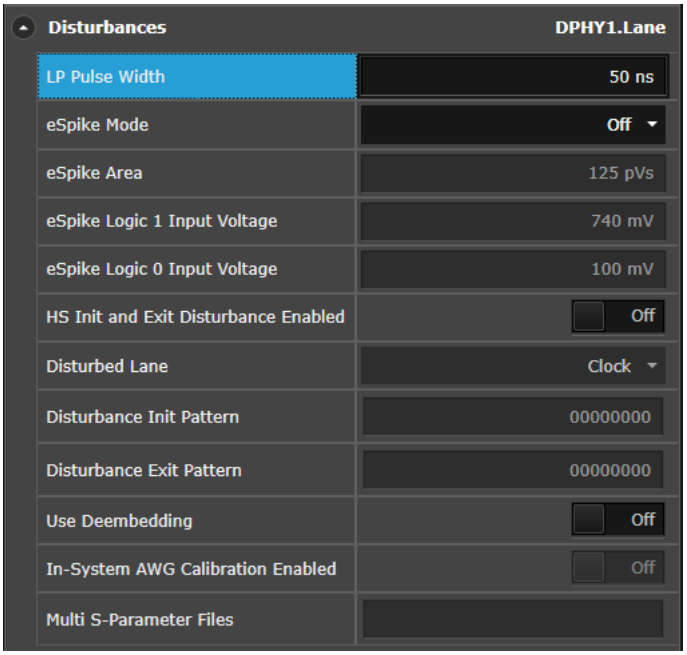


Figure 34 Disturbances Group

Disturbances Group Parameters:

- LP Pulse Width: Allows you to change the pulse width at the Low Power mode within the defined ranges. Changing the value of LP pulse width does not affect the data rate. The data rate remains constant because increasing the pulse width by this option applies to LP High state, which changes the duty cycle and simultaneously, decreases the width of LP low state.
- eSpike Mode: Allows you to enable or disable eSPIKE generation on the high levels or on the low levels of the pulse. The eSPIKE mode is used to test the ability of LP receiver to reject any input signal smaller than the eSPIKE.

- eSpike Area: Defines the area covered by the e_{SPIKE}. It can be calculated as following:
  - Area calculation: With eSPIKE in High Levels, the glitch goes from LP high level to LP low level, the area calculation begins below the selected “Logic 1 Input Voltage to the peak of the glitch”. The width of the e_{SPIKE} increases when you increase the value of Logic 1 Input Voltage, which simultaneously increases the area. “Logic 0 Input Voltage” has no effect on high level e_{SPIKE}. Similarly, you can calculate the area of e_{SPIKE} in Low Levels.
- eSpike Logic 1 Input Voltage: Specifies the lower receiver detection threshold level for a logic 1. It is the voltage level just below the LP high level voltage and is represented by V_{IH} (refer to Figure 33).
- eSpike Logical 0 Input Voltage: Specifies the upper receiver detection threshold level for a logic 0. It is the voltage level just above the LP low level voltage and is represented by V_{IL} (refer to Figure 33).
  - Some examples are (refer to Figure 35):
    - If you select eSpike Logic 1 Input Voltage = 740 mV and eSpike Logic 0 Input Voltage = 300 mV, eSpike Area = 350 pVs
    - If you select eSpike Logic 1 Input Voltage = 740 mV and eSpike Logic 0 Input Voltage = 100 mV, eSpike Area = 500 pVs

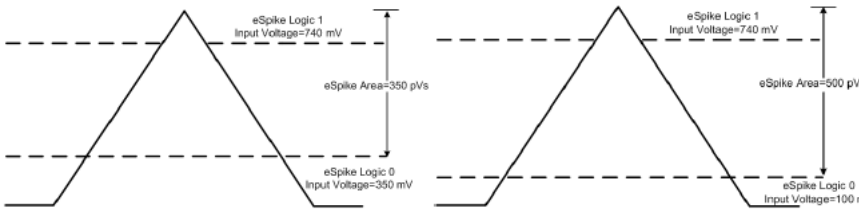


Figure 35 Examples of eSPIKE

- HS Init and Exit Disturbance Enabled: Allows you to enable the disturbance on the High speed Burst by adding one byte of data at the beginning (Init Pattern) and at the end (Exit Pattern). If set to On, the following options are enabled:
  - Disturbed Lane: Allows you to select the lane where you want to add the disturbance. Select one of the following options to add disturbance:
    - Clock: Add disturbance to the clock lanes. This option is available only when the clock is also in burst mode (and not in continuous mode)

- All Data: Add disturbance to all data lanes simultaneously.
- Data<n>: Add disturbance to a specific data lane. <n> indicates the specific number of the data lane, such as Data0.

- Disturbance Init Pattern: Define the disturbance pattern that must be added to the initialization of the High Speed burst.
- Disturbance Exit Pattern: Define the disturbance pattern that must be added to the exit of the High Speed burst.

- Use Deembedding: If set to 'On', this parameter applies a software filter to compensate impairments of the AWG output amplifier, thereby, improving the MIPI D-PHY signal quality. It also enables the parameter to enable/disable the In-System AWG Calibration.
- In-System AWG Calibration Enabled: If set to 'On', this parameter enables the In-System AWG Calibration data for de-embedding. While the process of a normal de-embedding achieves de-embedding of the output amplifier only, using In-System Calibration achieves de-embedding of cables and probes as well (that is, the entire signal path), if applied. For more information regarding how to perform In-System AWG Calibration, see Performing In-System AWG Calibration with Keysight IQ Tools on page 99.
- Multi S-Parameter Files: Define the s-parameter files to be de-embedded from the calculated waveform.

  To select one or more S-Parameter files, click this field and navigate to *C:\ProgramData\BitifEye\ValiFrame\SParameter\DPhy*. Select the required files and click Open. The selected file locations appear on the field and the de-embedding attributes are loaded.

Signal Interference Group

To change the signal interference settings, click the "Signal Interference" button in the "Signal Modifications & Impairments" area such that the corresponding settings are visible on the "Parameters" panel. Other than setting values for "Signal Interference" in the "Parameters" panel of the user interface, you may also run the SCPI Commands for Signal Interference Group on page 170.

The Signal Interference settings indicate the lanes, lines, mode of the interference that is to be introduced along with amplitude and frequency, of the interfering signal.

The different values of interferences in lane pair, mode, amplitude and frequency for different lanes can be applied in the same manner as described in section Setting different signal levels for lanes on page 78.

By default, the Signal Interference settings is disabled and you must enable it only when the waveform generation must be performed along with an available interfering signal.

The Signal Interference group allows to generate signal interference that is super-imposed on the MIPI D-PHY waveform. This is particular useful to emulate external interferences on the MIPI D-PHY devices.

The following figure shows parameters for signal interference group:

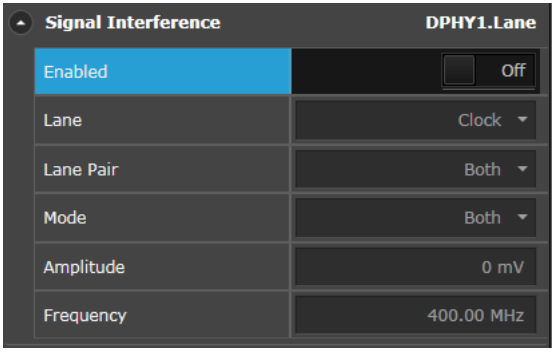


Figure 36 Signal Interference Group

Signal Interference Group Parameters:

- Enabled: Enables or Disables the sinusoidal Signal Interference generation.
- Lane: Represents the target lane where the Signal Interference are added. It allows the interference on the all data lanes simultaneously or on a specific data lane.
- Lane pair: Represents the target lane's pair where the Signal Interference are added. It allows the interference on the Normal, Complement or Both.
- Mode: Allows to add interference on the High Speed signal, Low Speed signal or on both.
- Amplitude: Amplitude of the generated interference. Interference Amplitude is always measured 0Vpk terminated, regardless of whether it is inserted in HS mode or LP mode.
- Frequency: Frequency of the generated interference.

Skew Group

To change the skew settings, click the “Skew” button in the “Signal Modifications & Impairments” area such that the corresponding settings are visible on the “Parameters” panel. Other than setting values for “Skew” in the “Parameters” panel of the user interface, you may also run the SCPI Commands for Skew Group on page 173.

The Skew Group parameter gives the ability to change the skew in seconds (usually pico seconds) of a specific channel on the AWGs. This feature is useful for de-skewing the influences of cables. Depending on the configured setup, some of the parameters do not apply and they are disabled.

The following figure shows parameters for skew group:

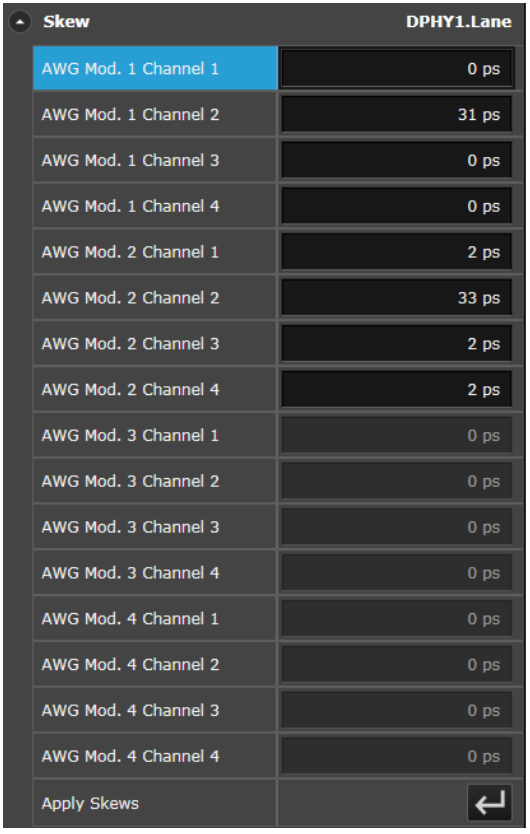


Figure 37 Skew Group

NOTE The "Apply Skews" buttons is bounded with the "Apply Delays". So when the "Apply Skews" is pressed the delays parameters are also applied. Similarly, when the "Apply Delays" is pressed the skews parameters are also applied.

Delay Group

To change the delay settings, click the "Delay" button in the "Signal Modifications & Impairments" area such that the corresponding settings are visible on the "Parameters" panel. Other than setting values for "Delay" in the "Parameters" panel of the user interface, you may also run the SCPI Commands for Delay Group on page 175.

The Delay Group allows you to change the delays among the lanes. If the Enable Intra-Line Delay is activated it is also possible to apply delays among the lines. In order to apply the delays, click the "Apply Delays" button or alternatively click the "Apply" button of the toolbar. When the "Apply Delays" button is clicked the MIPI D-PHY Editor tries to use the hardware resources to generate the delays. If the hardware resources are not enough, the delays need to be embedded in the pattern and therefore the waveform needs to be recalculated.

NOTE The "Apply Delays" buttons is bounded with the "Apply Skews". So when the "Apply Delays" is pressed the skews parameters are also applied. Similarly, when the "Apply Skews" is pressed the delays parameters are also applied.

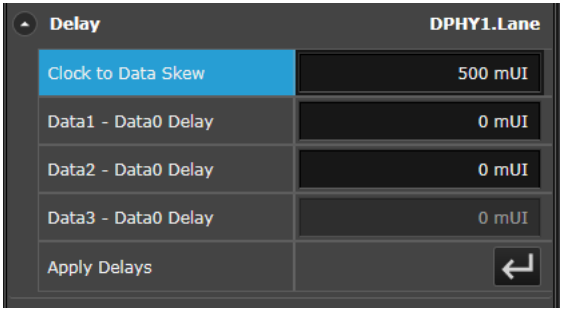


Figure 38 Delay Group

Performing In-System AWG Calibration with Keysight IQ Tools

To improve the quality of the MIPI D-PHY signal, you may enable the calibrated values obtained via the In-System Calibration of the connected AWGs. The In-System Calibration can be performed using the Keysight IQ Tools software, which you may download from www.keysight.com.

To perform In-System Calibration:

1 Launch the *Keysight IQ Tools* software.

2 In the **Configuration** area of the main window, click **Configure instrument connection**.

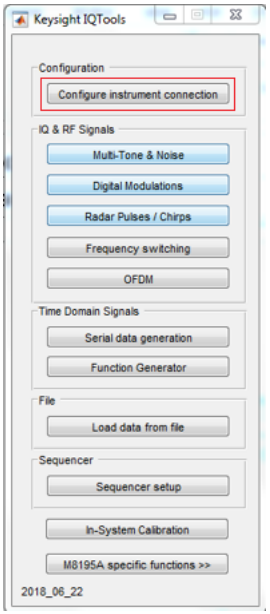


Figure 39 Main window of the IQ Tools software

3 In the **Arbitrary Waveform Generator Configuration** section of the **Instrument Configuration** window,

- For 4-Channel mode:
  - *a* Select **Instrument model** as *M8195A*
  - *b* Select **Mode** as *4 ch, deep mem, 16 GSa/s*
  - *c* Select **Connection** type as *visa*
  - *d* In the **VISA Address** field, type the address of the M8195A module

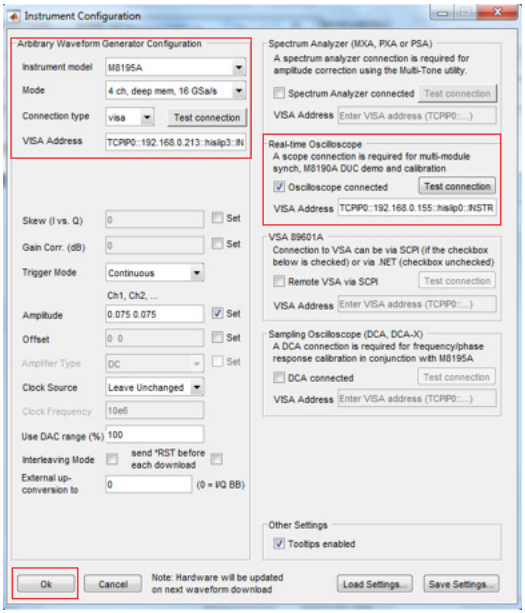


Figure 40 Settings on the Instrument Configuration window for 4-Channel

- For Dual Channel mode:
  a Select **Instrument model** as *M8195A*
  b Select **Mode** as *2 ch, deep mem, 32 GSa/s*
  c Select **Connection** type as *visa*
  d In the **VISA Address** field, type the address of the M8195A module

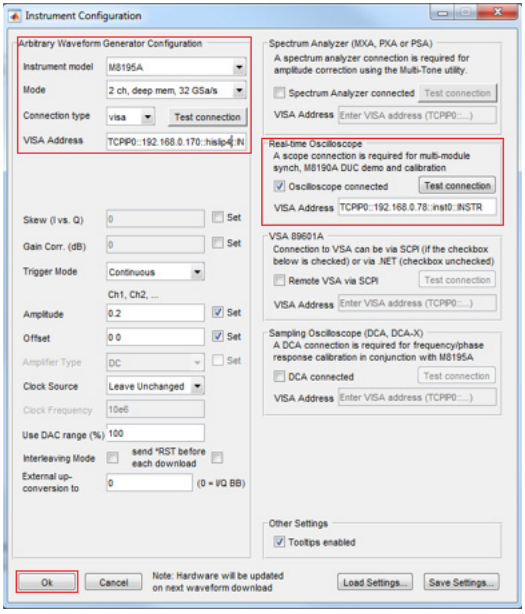


Figure 41 Settings on the Instrument Configuration window for Dual Channel

4 In the **Real-time Oscilloscope** section of the **Instrument Configuration** window,
  a Select the **Oscilloscope connected** check box, if not checked already
  b In the **VISA Address** field, type the address of the Oscilloscope.
5 Click **OK** on the Instrument Configuration window.

6 In the **Time Domain Signals** area of the main window, click **Serial data generation**.

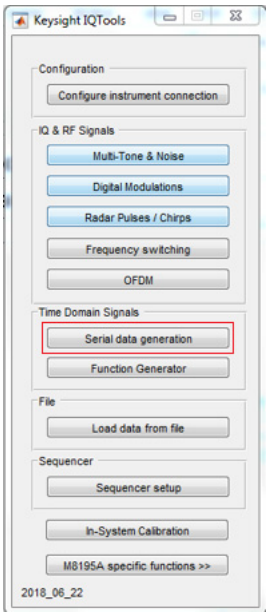


Figure 42 Accessing Serial data generation option

7 In the **Serial Data** window that appears, click **Show Correction**.

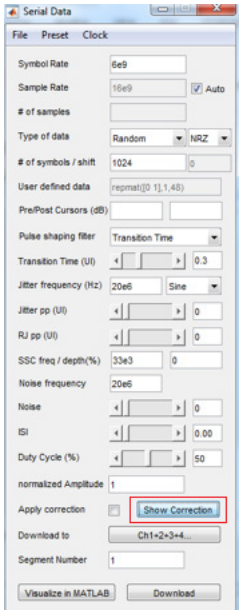


Figure 43 Accessing the Correction Management window

8 From the **Correction Management** window that appears, click **In-System Calibration**.

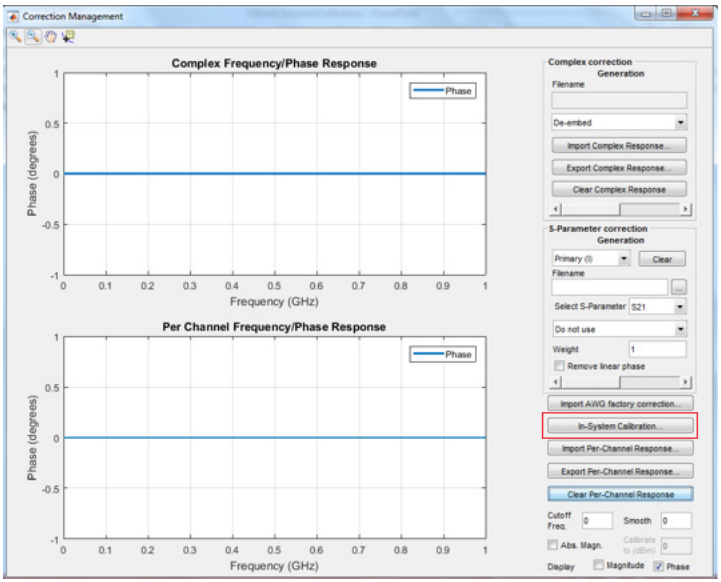


Figure 44 Selecting In-System Calibration option

9 On the **Frequency/Phase response calibration** window that appears,

- For 4-Channel mode

  a Ensure that the physical channels of the AWG and the Oscilloscope are connected as per the configuration shown in the **Channel Mapping** area (AWG Ch1 to Oscilloscope Ch1 and so on).

NOTE On the AWG complement outputs, ensure that terminations are connected.

b In the **Settings** area,

- **Sample Rate** value remains as-is
- If a Low-Pass Filter (LPF) is connected, the **Max. tone frequency** value should not exceed *8e9*

c Click **Run** to begin calibration measurements.

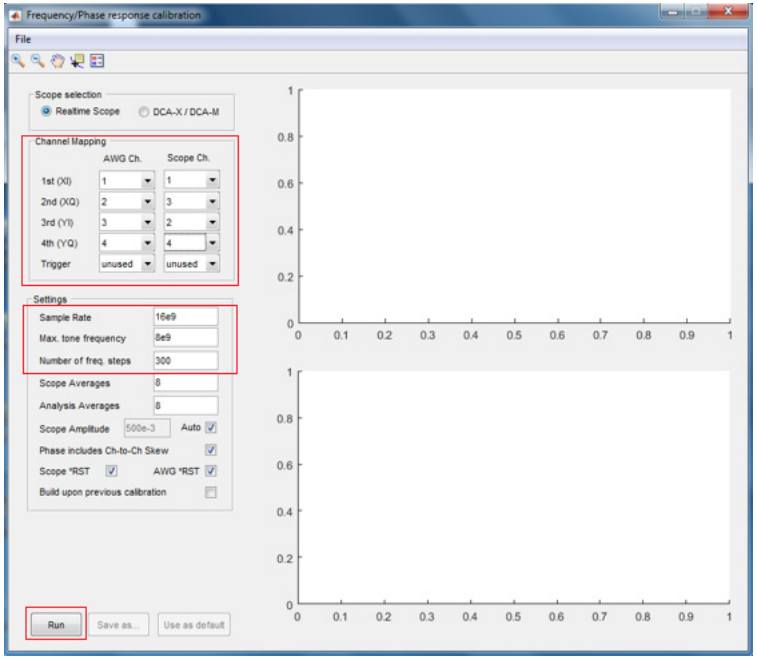


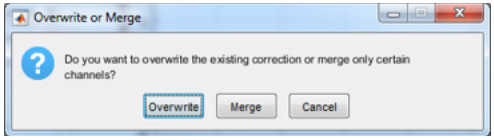
Figure 45 Validating settings to perform In-System Calibration

d If error messages are displayed,

- Validate the AWG-Oscilloscope Channel Mapping
- Reduce the Max. tone frequency value and try again

e If one or more outliers are found in the measurement, repeat the measurement a few more times.

f Once measurements are performed, overwrite the results.



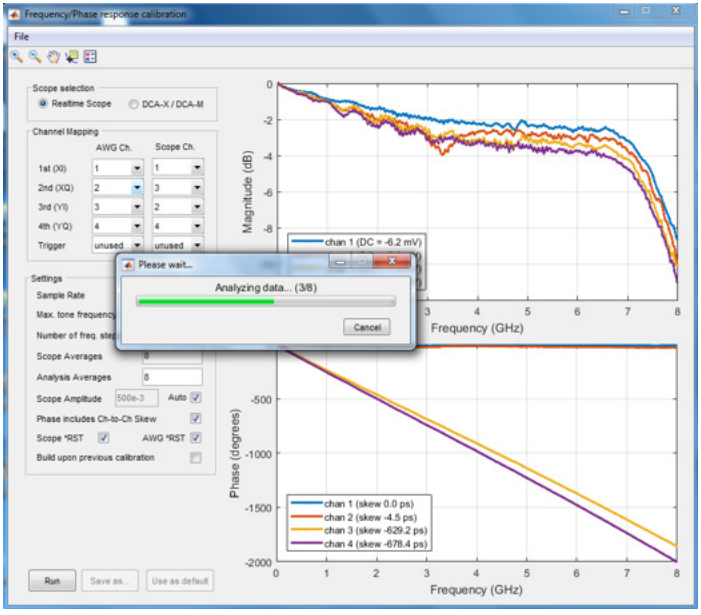


Figure 46 Viewing In-System Calibration results

- For Dual-Channel mode
  a Ensure that the physical channels of the AWG and the Oscilloscope are connected as per the configuration shown in the **Channel Mapping** area (AWG Ch1 to Oscilloscope Ch1 and so on).

NOTE On the AWG complement outputs, ensure that terminations are connected.

  b In the **Settings** area,
    - **Sample Rate** value remains as-is
    - No Low-Pass Filter (LPF) is connected, therefore, the **Max. tone frequency** value should be *16e9*
  c Click **Run** to begin calibration measurements.

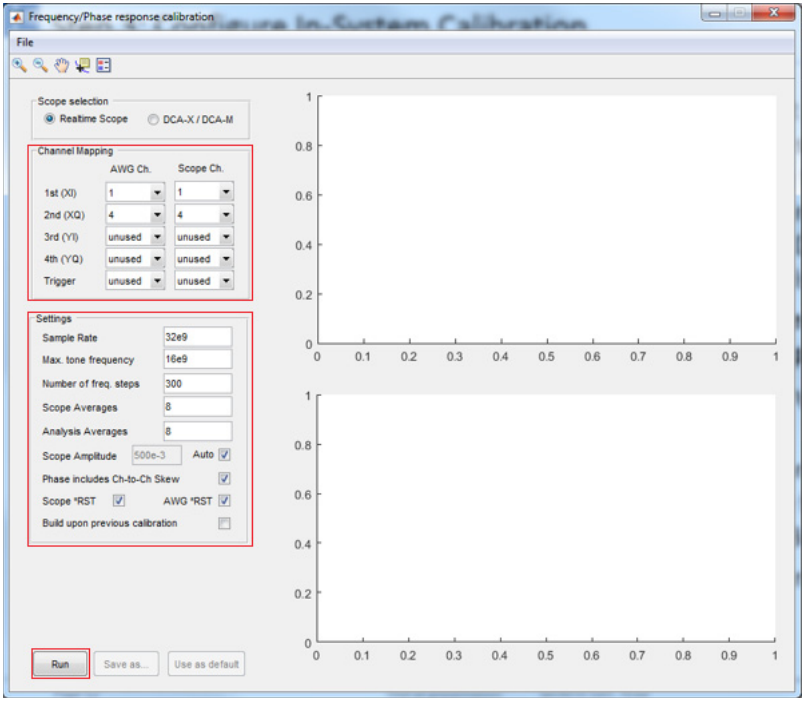


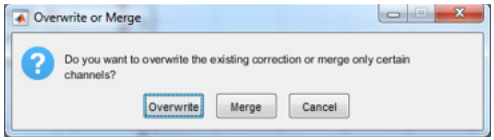
Figure 47 Validating settings to perform In-System Calibration

d If error messages are displayed,

- Validate the AWG-Oscilloscope Channel Mapping
- Reduce the Max. tone frequency value and try again

e If one or more outliers are found in the measurement, repeat the measurement a few more times.

f Once measurements are performed, overwrite the results.



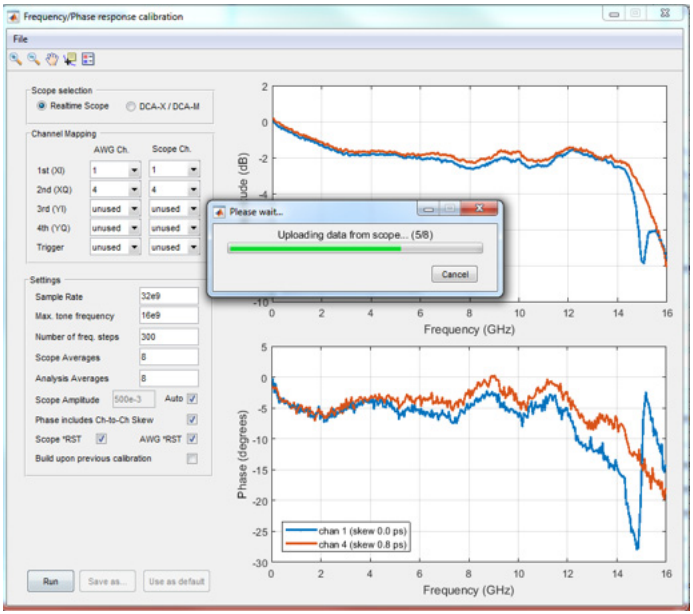


Figure 48 Viewing In-System Calibration results

10 To view the updated results, return to the **Correction Management** window.

- For 4-Channel mode

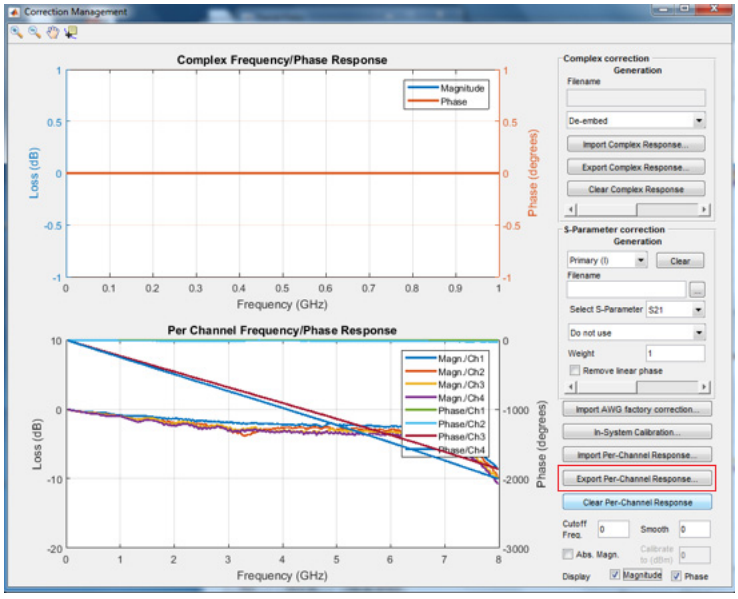


Figure 49 Exporting the Per-Channel Response

- For Dual-Channel mode

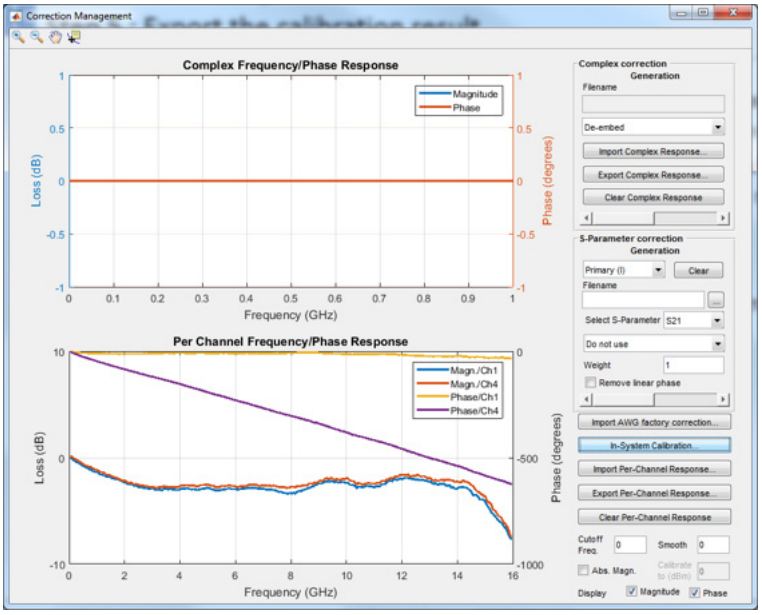


Figure 50 Exporting the Per-Channel Response

11 On the same window, click **Export Per-Channel Response...**.

12 On the **Save Frequency Response As...** window,

- For 4-Channel mode

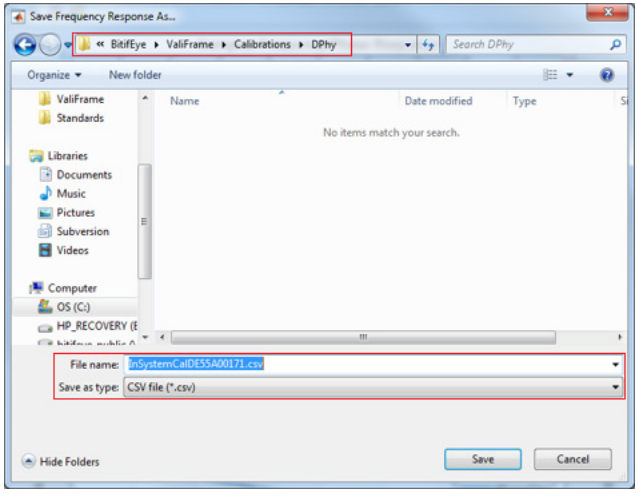


Figure 51 Saving the In-System Calibration values

a In the **Save as type:** drop-down field, select *CSV file (*.csv)*

b Navigate to the location *C:\ProgramData\BitifEye\ValiFrame\Calibrations\Dphy*

c Save the file with the following naming convention:

InSystemCal<*AWG-Serial-Number*>.csv

- For Dual-Channel mode

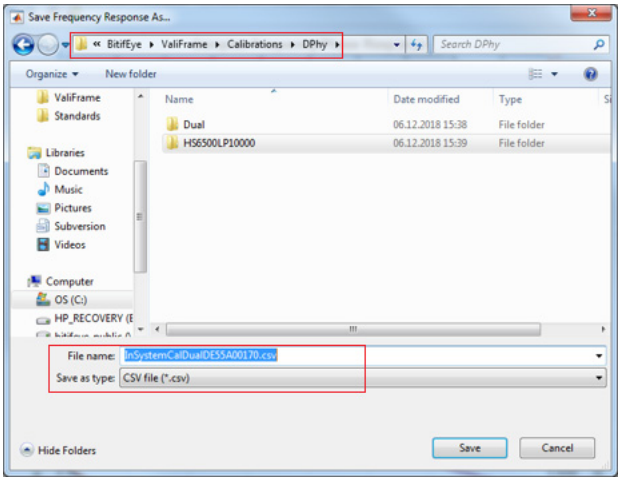


Figure 52 Saving the In-System Calibration values

a In the **Save as type:** drop-down field, select *CSV file (*.csv)*

b Navigate to the location *C:\ProgramData\BitifEye\ValiFrame\Calibrations\Dphy*

c Save the file with the following naming convention:

InSystemCalDual<*AWG-Serial-Number*>.csv

d To obtain the AWG Serial Number, go to **About Keysight M8195A**.

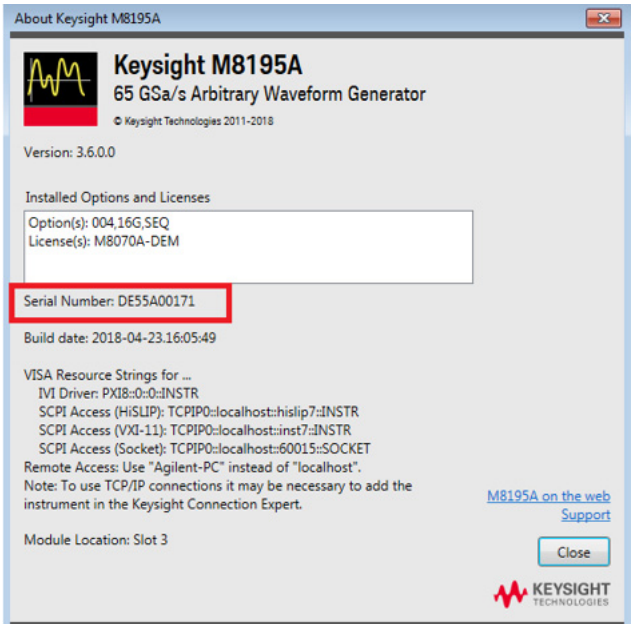


Figure 53 Identifying the AWG Serial Number

13 Repeat the entire process for each AWG that is physically connected.

14 Ensure to save the generated calibration output in the correct folder location with the CSV file format and the correct naming convention, as described earlier.

New Sequences in MIPI D-PHY 2.1

There are three sequences that are introduced in MIPI D-PHY specification 2.1 and the corresponding features in version 2.7 of the M8085A MIPI D-PHY Editor plug-in.

- Alternate Calibration Sequence
- Preamble Sequence
- HS-Idle Sequence

Understanding Alternate Calibration Sequence

When data rate of a transmitter signal is set to or modified to 2.5 Gbps or above, an alternate calibration sequence is applied. The purpose of this calibration is to achieve Link power-up, Link reinitialization or both.

On signals operating at the aforementioned data rates, the Alternate Calibration Sequence follows an Initial Skew Calibration. The Alternate Calibration Sequence consists of a leading HS0 pattern, a Calibration Sync and a Calibration pattern (a PRBS9 sequence).

Figure 54 depicts the application of the Alternate Calibration Sequence after the Initial Skew Calibration Sequence in a signal with data rate of 2.5 Gbps and above.

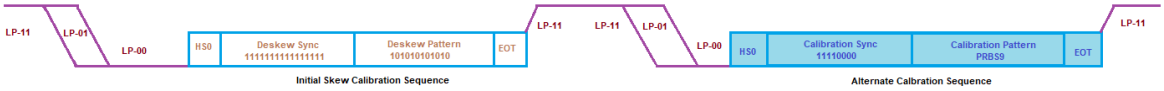


Figure 54 Signal with Alternate Calibration Sequence applied

To apply the Alternate Calibration Sequence in the signal generated using the MIPI D-PHY Editor, select **Frames** in the **Data Pattern** area followed by opening the *HsAlternateCal.seq* file from the location *C:\ProgramData\BitifEye\ValiFrame\Pattern\DPhy* in the **Sequence** field. Ensure that the **HS Data Rate** parameter is set to 2.5 Gb/s or higher. Then, click **Protocol Timings** in the **Signal Modifications & Impairments** area. Here, the **TX-InitialSkewCal Duration** parameter controls the timing for the Deskew Pattern in the Initial Skew Calibration Sequence applied on the signal, as shown in Figure 54.

To set the duration for the Calibration Pattern (PRBS9) in the Alternate Calibration Sequence, configure the value for **TX-AlternateCal Duration** parameter, which corresponds to the T_{ALTCAL} duration, as shown in Figure 55. As per *Table 19: Alternate Calibration Timing Parameters* of the *MIPI Alliance Specification for D-PHY*^{SM} *version 2.1*, the typical value of T_{ALTCAL_SYNC} is 8 UI whereas the maximum value that can be configured for T_{ALTCAL} duration, which corresponds to the **TX-AlternateCal Duration** parameter in the MIPI D-PHY plug-in, is 100 µs.

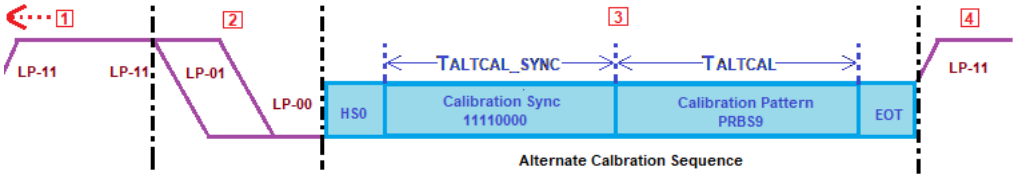


Figure 55 Understanding the Alternate Calibration Mode

Let's consider Figure 55 again to correlate the elements in the diagram with the Alternate Calibration Sequence file definition, which is applied in the MIPI D-PHY Editor plug-in during signal generation.

Following are the contents of the Alternate Calibration Sequence file *HsAlternateCal.seq*:

HSFreq: 1 GBit/s;

Blocks:

Init: LP11N128;

PreCal: LP11N16,LPHS;

AlternateCal: AlternateCal;

PostCal: LP11N128;

HS: LP11N1,B"HsCompliancePattern.dat",LP11N1;

LP: LP11N1,LPB"LpCompliancePattern.dat",LP11N1;

Sequence:

1. Init,1;

2. PreCal,1;

3. AlternateCal,1;

4. PostCal,1;

The *Init* block defines the LP-11 state along with the Initial Skew Calibration Sequence, which is depicted by the region marked as 1 in red text.

The next block *PreCal* defines the region marked as 2 in red text, where the signal transitions from LP-11 to LP-10 state.

The Alternate Calibration mode is defined by the *AlternateCal* block and is marked as 3 in red text. This block consists of a small HS-0 state followed by Calibration Sync pattern (11110000) of 8 UI duration. Then, a payload of PRBS9 sequence is sent in the Alternate Calibration Pattern, wherein a PRBS sequence generator is initialized prior to the Alternate Calibration to achieve the required payload, where the LSB is transmitted first. For more information regarding the PRBS sequence generated for the Alternate Calibration mode, refer to section *6.13 Alternate Calibration Sequence* of *MIPI Alliance Specification for D-PHY*^{SM} *version 2.1*.

The *PostCal* block, which defines the transition to LP-11 state is marked as 4 in red text.

The HS and LP blocks are not shown in the figure but they indicate that after the 128 iterations of LP-11 state, the next LP-11 state marks the beginning of the HS data payload (defined in the *HsCompliancePattern.dat* file) and another LP-11 state marks the end of the HS block. Similarly, the LP data payload (defined in the *LpCompliancePattern.dat* file) has one LP-11 state at the beginning and end of the LP block.

Therefore, for signals transmitted at 2.5 Gbps or higher, the Alternate Calibration Sequence of **TX-AlternateCal Duration** is inserted; whereas for signals with data rate less than 2.5 Gbps, the Alternate Calibration Sequence is optional.

Understanding Preamble Sequence

On signals that are transmitted at data rates above 2.5 Gbps, the Preamble Sequence of short length can be optionally inserted at the start of every high speed (HS burst) payload to achieve a better clock-to-data skew, which occurs due to variations in temperature and voltage.

A normal High Speed Burst consists of an HS-0 state followed by a leader sequence with a pattern 00011101, which marks the beginning of the High-Speed Packet Data. The Preamble Sequence can be inserted just before the Leader Sequence.

The Preamble Sequence consists of a preamble pattern 101010 with a duration of $T_{PREAMBLE}$, which can be configured from a minimum of 32 UI to a maximum of 512 UI, as per *Table 20: Preamble Timing Parameters* of

the *MIPI Alliance Specification for D-PHY^{SM} version 2.1*. The Preamble Sequence also contains an Extended Sync pattern HS-1 of fixed length and with a duration of $T_{EXTSYNC}$, having a typical duration of 8 UI. The Extended Sync pattern is required to separate the Preamble Sequence from the Leader Sequence, to prevent the former from being detected by receivers as the Leader Sequence.

To enable Preamble Sequence on a High Speed Burst in the M8085A MIPI D-PHY Editor plug-in, the option **Preamble** in the **Data Pattern** area must be checked. Also, you can set the multiplier value for the required preamble steps to be inserted into the HS Burst by configuring the **Preamble Steps** parameter under **Protocol Timings**. By default, this value is set to 1, which corresponds to a typical $T_{PREAMBLE}$ duration of 32 UI. The plug-in multiplies the configured value of the Preamble Steps parameter with 32 UI, which in turn, sets the $T_{PREAMBLE}$ duration on the signal. Only prerequisite for the Preamble Sequence to be inserted by the MIPI D-PHY Editor is that the **HS Data Rate** must be configured to a value more than 2.5 Gb/s, otherwise the plug-in ignores the Preamble flag even if enabled.

Figure 56 shows the difference in the appearance of a normal High Speed Burst and a High Speed Burst that has Preamble Sequence enabled.

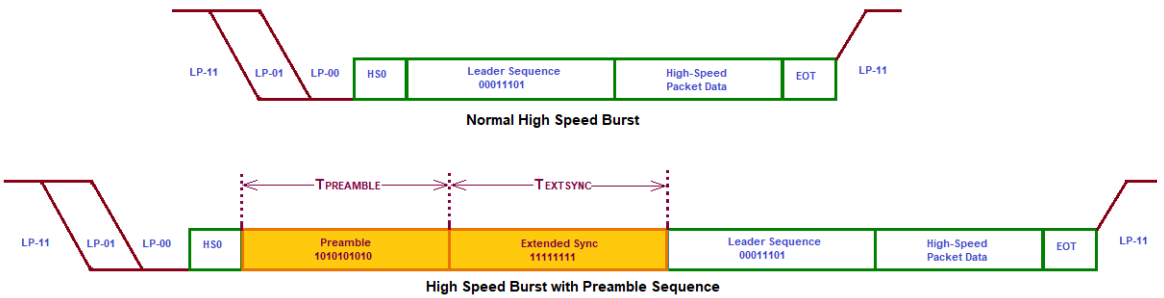


Figure 56 High Speed Burst - without and with Preamble sequence applied

Understanding HS-Idle Sequence

The HS-Idle Sequence is used to apply an HS-Idle state between two High-Speed Data Bursts during HS signaling, that is, without the need for transitioning to LP signaling. The purpose for using the HS-Idle state is to reduce the latency between the two HS Data Bursts, based on the data rate and HS-Idle timings.

As shown in Figure 57, the HS-Idle state comprises of HS-Idle-Post, HS-Idle-ClkHS0 and HS-Idle-Pre sub-states, with durations $T_{HS-IDLE-POST}$, $T_{HS-IDLE-CLKHS0}$ and $T_{HS-IDLE-PRE}$, respectively. After the last bit of the HS Data Bursts on all lanes, the Clock lane continues signaling for a duration of $T_{HS-IDLE-POST}$. Then, the Clock lane enters into HS-0 state for a duration of $T_{HS-IDLE-CLKHS0}$ and eventually, the Clock lane is active for a duration $T_{HS-IDLE-PRE}$ prior to exiting from HS-Idle state and entering into the first bit of the next HS Data Burst. For more information regarding the state transitions for the HS-Idle state, refer to section *6.15 HS-Idle State* of *MIPI Alliance Specification for D-PHY^{SM} version 2.1*.

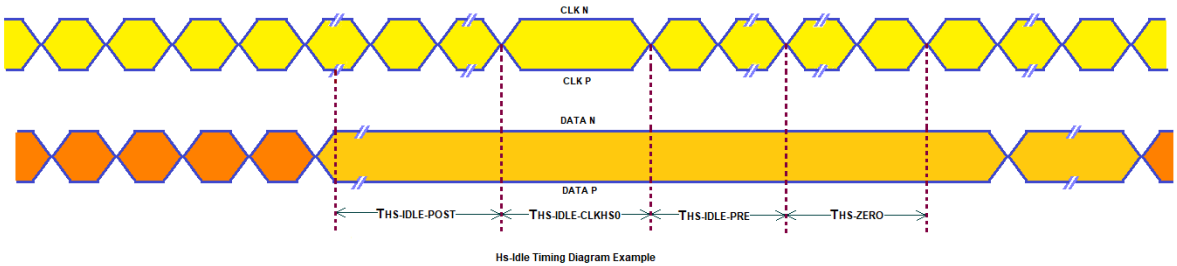


Figure 57 Signal with HS-Idle state applied

To apply the HS-Idle states in the signal generated using the MIPI D-PHY Editor, select **Frames** in the **Data Pattern** area followed by opening the *HsIdle.seq* file from the location *C:\ProgramData\BitifEye\ValiFrame\Pattern\DPhy* in the **Sequence** field. Then, click **Protocol Timings** in the **Signal Modifications & Impairments** area. Here, the **TX-HsIdlePost Duration** parameter controls the $T_{HS-IDLE-POST}$ duration, the **TX-HsIdleClkHs0 Duration** parameter controls the $T_{HS-IDLE-CLKHS0}$ duration and the **TX-HsIdlePre Duration** parameter controls the $T_{HS-IDLE-PRE}$ duration of the HS-Idle state. By default, the M8085A MIPI D-PHY Editor plug-in sets the “Default” values of these three timing parameters. The minimum, default and maximum values for each of these timing parameters are defined in *Table 21: HS-Idle State Timing Parameters* of *MIPI Alliance Specification for D-PHY^{SM} version 2.1*.

Let's correlate the elements in the diagram with the HS-Idle Sequence file definition, which is applied in the MIPI D-PHY Editor plug-in during signal generation.

Following are the contents of the HS-Idle Sequence file *HsIdle.seq*:

```
HSFreq: 1 GBit/s;

Blocks:
Init: LP11N128;
PreHs:LP11N1,B"HsCompliancePattern.dat";
Idle: HsIdleState;
PostHs:B"HsCompliancePattern.dat",LP11N1;

Sequence:
1. Init,1;
2. PreHs,1;
3. Idle,1;
4. PostHs,1;
LoopTo 1;
```

The *Init* block defines the LP-11 state.

The *PreHs* block defines the HS data payload (defined in the *HsCompliancePattern.dat* file) prior to entering Hs-Idle state.

Unlike other sequences, an LP-11 state does not mark the end of the HS data payload. Instead, the *Idle* block, which defines the HSIdleState, indicates the transition of the last bit of the HS data burst into the HS-Idle state.

Similarly, the next block *PostHs* defines the HS data payload (defined in the *HsCompliancePattern.dat* file) after exiting the Hs-Idle state.

4 SCPI Programming

KEYSIGHT TECHNOLOGIES

SCPI Command Language

The M8085A 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

Data Types

The M8085A has the capability of receiving and returning data in the following formats:

STRING

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

NUMERIC

The M8020A 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 as either ON | OFF or 0 | 1. The M8020A 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

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

- Instrument Model
- Command Syntax
- Query Response
- Command Separators
- SCPI Command Structure

Instrument Model

SCPI guidelines require that the M8020A 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.

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 VOLTage has the short-form VOLT. 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.

Query Responses

It is possible to cross-examine 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.
- A colon after identifier is used to select specific location. For example, 'MIPI D-PHY Editor 1:DPHY1.Clock' shows that the SCPI is applied to the clock lane location of MIPI D-PHY Editor.

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 command provides example of this.

:PLUGin:DPHYplugin:LINK:OSCilloscope:OFFline 'MIPI D-PHY Editor 1', 1

The same command can also be sent as follows:

:PLUG:DPHY:LINK:OSC:OFF 'MIPI D-PHY Editor 1', 1

SCPI Command Structure Example

The SCPI command structure can be best examined by means of an example. For example, the command enables the plugin to work without connecting to the oscilloscope:

:PLUGin:DPHYplugin:LINK:OSCilloscope:OFFline 'MIPI D-PHY Editor 1', 1

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. A colon after identifier is used to select specific location.

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 possible forms of the command line; all are acceptable.

In long form:

:PLUGin:DPHYplugin:LINK:OSCilloscope:OFFline 'MIPI D-PHY Editor 1', 1

In short form:

:PLUG:DPHY:LINK:OSC:OFF 'MIPI D-PHY Editor 1', 1

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

SCPI Editor

The **SCPI Editor** lists all SCPI that can be used to program M8020A and also provides a platform to execute them.

The following figure depicts the M8020A **SCPI Editor** user interface:

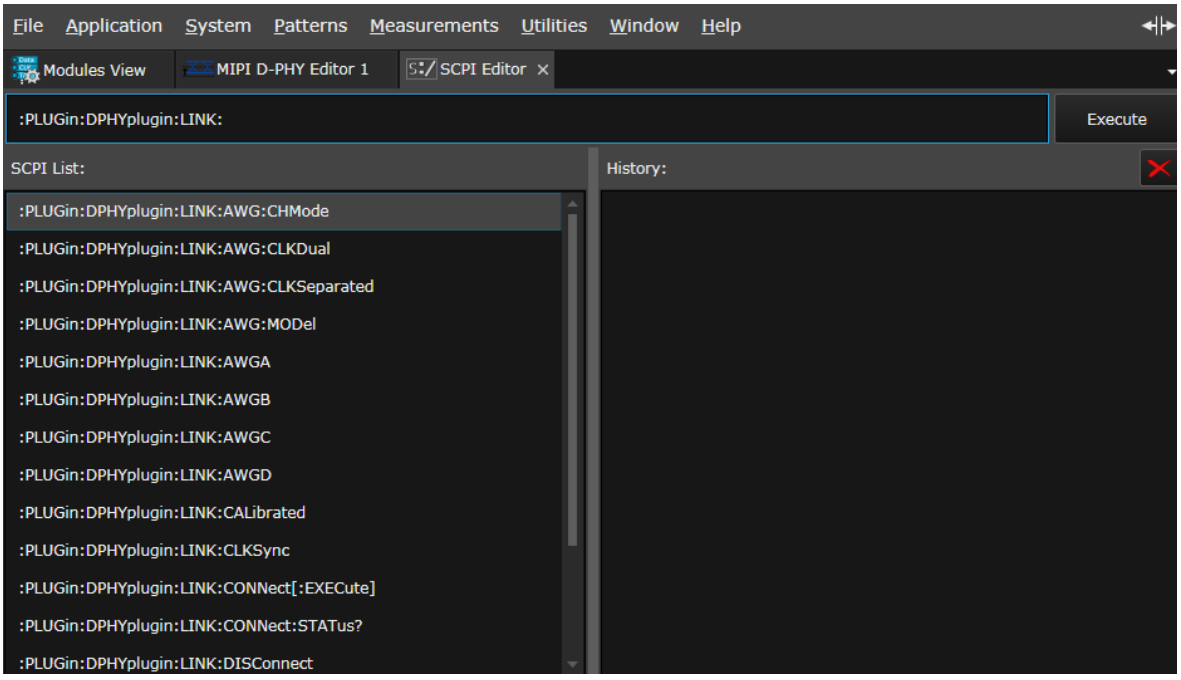


Figure 58 M8020A SCPI Editor

For complete details, refer to section “SCPI Editor” in the *M8020A User Guide*.

Executing SCPI Command

To execute a SCPI command, follow the given steps:

- Select the SCPI from the given list. You can also type the SCPI in the provided text box to expedite the command search.
- Use the proper SCPI command syntax along with the command separators.

  The following example shows a SCPI command enables the plugin to work without connecting to the oscilloscope:

:PLUG:DPHY:LINK:OSC:OFF 'MIPI D-PHY Editor 1', 1

- Click **Execute**. The output of the SCPI command will be displayed in the **History** pane.

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 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)

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)

SCPI Command Reference

The MIPI D-PHY Editor has the following commands:

SCPI Commands for Connection Group

Table 6 SCPI Commands for Connection Group

Command	Description under
:PLUGin:DPHYplugin:DELete	For details, see **:PLUGin:DPHYplugin:DELete** on page 129.
:PLUGin:DPHYplugin:LINK:CALibrated[?]	For details, see **:PLUGin:DPHYplugin:LINK:CALibrated[?]** on page 129.
:PLUGin:DPHYplugin:NEW	For details, see **:PLUGin:DPHYplugin:NEW** on page 129.
:PLUGin:DPHYplugin:CATalog?	For details, see **:PLUGin:DPHYplugin:CATalog?** on page 129.
:PLUGin:DPHYplugin:PARameters:RESet	For details, see **:PLUGin:DPHYplugin:PARameters:RESet** on page 130.
:PLUGin:DPHYplugin:LINK:DISConnect	For details, see **:PLUGin:DPHYplugin:LINK:DISConnect** on page 130.
:PLUGin:DPHYplugin:LINK:OFFLine[?]	For details, see **:PLUGin:DPHYplugin:LINK:OFFLine[?]** on page 130.
:PLUGin:DPHYplugin:LINK:AWG:CHMode[?]	For details, see **:PLUGin:DPHYplugin:LINK:AWG:CHMode[?]** on page 130.
:PLUGin:DPHYplugin:LINK:AWG:CLKDual[?]	For details, see **:PLUGin:DPHYplugin:LINK:AWG:CLKDual[?]** on page 131.
:PLUGin:DPHYplugin:LINK:NLANes[?]	For details, see **:PLUGin:DPHYplugin:LINK:NLANes[?]** on page 131.
:PLUGin:DPHYplugin:LINK:AWG:CLKSeparated[?]	For details, see **:PLUGin:DPHYplugin:LINK:AWG:CLKSeparated[?]** on page 131.
:PLUGin:DPHYplugin:LINK:CLKSync[?]	For details, see **:PLUGin:DPHYplugin:LINK:CLKSync[?]** on page 132.
:PLUGin:DPHYplugin:LINK:AWGA[?]	For details, see **:PLUGin:DPHYplugin:LINK:AWGA[?]** on page 132.
:PLUGin:DPHYplugin:LINK:AWGB[?]	For details, see **:PLUGin:DPHYplugin:LINK:AWGB[?]** on page 133.
:PLUGin:DPHYplugin:LINK:AWGC[?]	For details, see **:PLUGin:DPHYplugin:LINK:AWGC[?]** on page 133.
:PLUGin:DPHYplugin:LINK:OSCilloscope:ADDRess[?]	For details, see **:PLUGin:DPHYplugin:LINK:OSCilloscope:ADDRess[?]** on page 134.
:PLUGin:DPHYplugin:LINK:OSCilloscope:OFFline[?]	For details, see **:PLUGin:DPHYplugin:LINK:OSCilloscope:OFFline[?]** on page 134.
:PLUGin:DPHYplugin:LINK:CONNect[:EXECute] [?]	For details, see **:PLUGin:DPHYplugin:LINK:CONNect[:EXECute] [?]** on page 134.

:PLUGin:DPHYplugin:DELete

Syntax :PLUGin:DPHYplugin:DELete 'MIPI D-PHY Editor 1'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command is used to delete the MIPI D-PHY Editor.

Example Command

:PLUGin:DPHYplugin:DELete 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:CALibrated[?]

Syntax :PLUGin:DPHYplugin:LINK:CALibrated 'identifier', <Boolean>

:PLUGin:DPHYplugin:LINK:CALibrated? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0 | 1

Description This command enables the plugin to work with calibrated values, if available.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:CALibrated 'MIPI D-PHY Editor 1',1

Query

:PLUGin:DPHYplugin:LINK:CALibrated? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:NEW

Syntax :PLUGin:DPHYplugin:NEW 'MIPI D-PHY Editor 1'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command enables the plugin to add new values, if available.

Example Command

:PLUGin:DPHYplugin:NEW 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:CATalog?

Syntax :PLUGin:DPHYplugin:CATalog? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This query returns the list of all open instance of MIPI D-PHY plugin

Example Query

:PLUGin:DPHYplugin:CATalog? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PARameters:RESet

Syntax :PLUGin:DPHYplugin:PARameters:RESet 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command enables the plugin to reset the values, if available.

Example Command

:PLUGin:DPHYplugin:PARameters:RESet 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:DISConnect

Syntax :PLUGin:DPHYplugin:LINK:DISConnect 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command enables the plugin to disconnect, if available.

Example Command

:PLUGin:DPHYplugin:LINK:DISConnect 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:OFFLine[?]

Syntax :PLUGin:DPHYplugin:LINK:OFFLine 'identifier', <Boolean>

:PLUGin:DPHYplugin:LINK:OFFLine? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0 | 1

Description This command enables the plugin to work without connecting to any instrument.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:OFFLine 'MIPI D-PHY Editor 1',1

Query

:PLUGin:DPHYplugin:LINK:OFFLine? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:AWG:CHMode[?]

Syntax :PLUGin:DPHYplugin:LINK:AWG:CHMode 'identifier', <String>

:PLUGin:DPHYplugin:LINK:AWG:CHMode? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', DUAL | FOUR

Description This command configures the M8195A AWG to either Dual Channel or Four Channel mode.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:AWG:CHMode 'MIPI D-PHY Editor 1', DUAL

Query

:PLUGin:DPHYplugin:LINK:AWG:CHMode? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:NLANes[?]

Syntax **:PLUGin:DPHYplugin:LINK:NLANes 'identifier', <Enum>**

:PLUGin:DPHYplugin:LINK:NLANes? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', LAN1 | LAN2 | LAN3 | LAN4

Description This command configures the number of lanes required to test.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:NLANes 'MIPI D-PHY Editor 1', LAN1

Query

:PLUGin:DPHYplugin:LINK:NLANes? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:AWG:CLKSeparated[?]

Syntax **:PLUGin:DPHYplugin:LINK:AWG:CLKSeparated 'identifier', <Boolean>**

:PLUGin:DPHYplugin:LINK:AWG:CLKSeparated? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command specifies the clock lane generated on a separated AWG module.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:AWG:CLKSeparated 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:LINK:AWG:CLKSeparated? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:AWG:CLKDual[?]

Syntax **:PLUGin:DPHYplugin:LINK:AWG:CLKDual 'identifier', <Boolean>**

:PLUGin:DPHYplugin:LINK:AWG:CLKDual? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command specifies the clock lane generated on a separated AWG module.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:AWG:CLKDual 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:LINK:AWG:CLKDual? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:CLKSync[?]

Syntax **:PLUGin:DPHYplugin:LINK:CLKSync 'identifier', <String>**

:PLUGin:DPHYplugin:LINK:CLKSync? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 'TCPIP0::193.162.1.1::hslip0::INSTR'

Description This command defines the VISA address of the Clock Sync module. (Only for M8195A).

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:CLKSync 'MIPI D-PHY Editor 1', 'TCPIP0::193.162.1.1::hslip0::INSTR'

Query

:PLUGin:DPHYplugin:LINK:CLKSync? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:AWGA[?]

Syntax **:PLUGin:DPHYplugin:LINK:AWGA 'identifier', <String>**

:PLUGin:DPHYplugin:LINK:AWGA? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 'TCPIP0::193.162.1.1::hslip1::INSTR'

Description This command defines the AWG VISA address of the first AWG module.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:AWGA 'MIPI D-PHY Editor 1', 'TCPIP0::193.162.1.1::hslip1::INSTR'

Query

:PLUGin:DPHYplugin:LINK:AWGA? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:AWGB[?]

Syntax **:PLUGin:DPHYplugin:LINK:AWGB 'identifier', <String>**

:PLUGin:DPHYplugin:LINK:AWGB? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 'TCPIP0::193.162.1.1::hslip1::INSTR'

Description This command defines the AWG VISA address of the second AWG module.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:AWGB 'MIPI D-PHY Editor 1', 'TCPIP0::193.162.1.1::hslip1::INSTR'

Query

:PLUGin:DPHYplugin:LINK:AWGB? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:AWGC[?]

Syntax **:PLUGin:DPHYplugin:LINK:AWGC 'identifier', <String>**

:PLUGin:DPHYplugin:LINK:AWGC? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 'TCPIP0::193.162.1.1::hslip1::INSTR'

Description This command defines the AWG VISA address of the third AWG module.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:AWGC 'MIPI D-PHY Editor 1', 'TCPIP0::193.162.1.1::hslip1::INSTR'

Query

:PLUGin:DPHYplugin:LINK:AWGC? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:OSCilloscope:ADDRess[?]

Syntax :PLUGin:DPHYplugin:LINK:OSCilloscope:ADDRess 'identifier', <String>

:PLUGin:DPHYplugin:LINK:OSCilloscope:ADDRess? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 'TCPIP0::192.162.1.3::inst0::INSTR'

Description This command defines the VISA address of the oscilloscope.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:OSCilloscope:ADDRess 'MIPI D-PHY Editor 1', 'TCPIP0::192.162.1.3::inst0::INSTR'

Query

:PLUGin:DPHYplugin:LINK:OSCilloscope:ADDRess? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:OSCilloscope:OFFline[?]

Syntax :PLUGin:DPHYplugin:LINK:OSCilloscope:OFFline 'identifier', < Boolean >

:PLUGin:DPHYplugin:LINK:OSCilloscope:OFFline? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command enables the plugin to work without connecting to the oscilloscope.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:OSCilloscope:OFFline 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:LINK:OSCilloscope:OFFline? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LINK:CONNect[:EXECute] [?]

Syntax :PLUGin:DPHYplugin:LINK:CONNect[:EXECute] 'identifier', < Boolean >

:PLUGin:DPHYplugin:LINK:CONNect:STATus? 'identifier',

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command shows if the MIPI D-PHY editor is connected to the instruments or not.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LINK:CONNect:EXECute ‘MIPI D-PHY Editor 1’, 0

Query

:PLUGin:DPHYplugin:LINK:CONNect? ‘MIPI D-PHY Editor 1’

SCPI Commands for Data Pattern Group

Table 7 SCPI Commands for Data Pattern Group

Command	Description under
:PLUGin:DPHYplugin:RUN:MODe[?]	For details, see **:PLUGin:DPHYplugin:RUN:MODe[?]** on page 136.
:PLUGin:DPHYplugin:PATTern:IDLe:VOLTage[?]	For details, see **:PLUGin:DPHYplugin:PATTern:IDLe:VOLTage[?]** on page 137.
:PLUGin:DPHYplugin:PATTern:AAmp:VOLTage[?]	For details, see **:PLUGin:DPHYplugin:PATTern:AAmp:VOLTage[?]** on page 137.
:PLUGin:DPHYplugin:PATTern:MODe[?]	For details, see **:PLUGin:DPHYplugin:PATTern:MODe[?]** on page 137.
:PLUGin:DPHYplugin:PATTern:SYNCword[:PATTern][?]	For details, see **:PLUGin:DPHYplugin:PATTern:SYNCword[:PATTern][?]** on page 138.
:PLUGin:DPHYplugin:PATTern:LSBFirst[?]	For details, see **:PLUGin:DPHYplugin:PATTern:LSBFirst[?]** on page 138.
:PLUGin:DPHYplugin:PATTern:LPFRaming[?]	For details, see **:PLUGin:DPHYplugin:PATTern:LPFRaming[?]** on page 138.
:PLUGin:DPHYplugin:PATTern:TRIGstart[:ENABle][?]	For details, see **:PLUGin:DPHYplugin:PATTern:TRIGstart[:ENABle][?]** on page 139.
:PLUGin:DPHYplugin:PATTern:TRIGstart:PATTern[?]	For details, see **:PLUGin:DPHYplugin:PATTern:TRIGstart:PATTern[?]** on page 139.
:PLUGin:DPHYplugin:PATTern:PREAmble[?]	For details, see **:PLUGin:DPHYplugin:PATTern:PREAmble[?]** on page 139.
:PLUGin:DPHYplugin:PATTern:HSData[:PATTern][?]	For details, see **:PLUGin:DPHYplugin:PATTern:HSData[:PATTern][?]** on page 140.
:PLUGin:DPHYplugin:PATTern:HSData:FILename[?]	For details, see **:PLUGin:DPHYplugin:PATTern:HSData:FILename[?]** on page 140.
:PLUGin:DPHYplugin:PATTern:LPData[:PATTern][?]	For details, see **:PLUGin:DPHYplugin:PATTern:LPData[:PATTern][?]** on page 141.
:PLUGin:DPHYplugin:PATTern:LPData:FILename[?]	For details, see **:PLUGin:DPHYplugin:PATTern:LPData:FILename[?]** on page 141.
:PLUGin:DPHYplugin:PATTern:SEQuence:FILename[?]	For details, see **:PLUGin:DPHYplugin:PATTern:SEQuence:FILename[?]** on page 141.

:PLUGin:DPHYplugin:RUN:MODe[?]

Syntax :PLUGin:DPHYplugin:RUN:MODe ‘pluginidentifier’, <mode>

:PLUGin:DPHYplugin:RUN:MODe? ‘pluginidentifier’

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’, INTerrupted | CONTinuous

Description This command changes the run mode of the plugin.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:RUN:MODe ‘MIPI D-PHY Editor 1’, INT

Query

:PLUGin:DPHYplugin:RUN:MODe? ‘MIPI D-PHY Editor 1’

:PLUGin:DPHYplugin:PATTern:IDLe:VOLTage[?]

Syntax :PLUGin:DPHYplugin:PATTern:IDLe:VOLTage 'pluginidentifier', <Enum>

:PLUGin:DPHYplugin:PATTern:IDLe:VOLTage? 'pluginidentifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command sets the offset voltage on the AWG when it is in Stop state.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:IDLe:VOLTage 'MIPI D-PHY Editor 1', 3.000e-001

Query

:PLUGin:DPHYplugin:PATTern:IDLe:VOLTage? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:AAmp:VOLTage[?]

Syntax :PLUGin:DPHYplugin:PATTern:AAmp:VOLTage 'pluginidentifier', <Enum>

:PLUGin:DPHYplugin:PATTern:AAmp:VOLTage? 'pluginidentifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command sets the signal amplitude on the AWG.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:AAmp:VOLTage 'MIPI D-PHY Editor 1', 4.500e-001

Query

:PLUGin:DPHYplugin:PATTern:AAmp:VOLTage? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:MODe[?]

Syntax :PLUGin:DPHYplugin:PATTern 'identifier', <Enum>

:PLUGin:DPHYplugin:PATTern:MODe? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', Burst | BCHClock | PUREhs | Frames | FCHClock

Description This command defines the pattern mode.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:MODe 'MIPI D-PHY Editor 1', Burst

Query

:PLUGin:DPHYplugin:PATTern:MODe? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:SYNCword[:PATTern][?]

Syntax **:PLUGin:DPHYplugin:PATTern:SYNCword[:PATTern] 'identifier', <String>**

:PLUGin:DPHYplugin:PATTern:SYNCword[:PATTern]? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0xB8

Description This command defines the synchronized word.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:SYNCword:PATTern 'MIPI D-PHY Editor 1', '0xB8'

Query

:PLUGin:DPHYplugin:PATTern:SYNCword? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:LSBFirst[?]

Syntax **:PLUGin:DPHYplugin:PATTern:LSBFirst 'identifier', <Boolean>**

:PLUGin:DPHYplugin:PATTern:LSBFirst? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command defines the byte transmission order. When LSB is selected, the least significant bit is sent first.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:LSBFirst 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:PATTern:LSBFirst? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:LPFRaming[?]

Syntax **:PLUGin:DPHYplugin:PATTern:LPFRaming 'identifier', <Boolean>**

:PLUGin:DPHYplugin:PATTern:LPFRaming? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command configures the number of lanes required to test.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:LPFRaming 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:PATTern:LPFRaming? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:TRIGstart[:ENABle][?]

Syntax **:PLUGin:DPHYplugin:PATTern:TRIGstart[:ENABle] 'identifier', <Boolean>**

:PLUGin:DPHYplugin:PATTern:TRIGstart[:ENABle]? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command activates the Triggered start.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:TRIGstart 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:PATTern:TRIGstart:ENABle? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:TRIGstart:PATTern[?]

Syntax **:PLUGin:DPHYplugin:PATTern:TRIGstart:PATTern 'identifier', <Enum>**

:PLUGin:DPHYplugin:PATTern:TRIGstart:PATTern? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', LP00 | LP11 | HS10

Description This command activates the Triggered start.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:TRIGstart:PATTern 'MIPI D-PHY Editor 1', LP00

Query

:PLUGin:DPHYplugin:PATTern:TRIGstart:PATTern? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:PREAmble[?]

Syntax **:PLUGin:DPHYplugin:PATTern:PREAmble 'identifier', <Boolean>**

:PLUGin:DPHYplugin:PATTern:PREAmble? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', ON | OFF | 1 | 0

Description This command enables the Preamble feature, which in turn, inserts Preamble Sequence in every HS Burst.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:PREAmble 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:PATTern:PREAmble? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:HSData[:PATTern][?]

Syntax **:PLUGin:DPHYplugin:PATTern:HSData[:PATTern] 'identifier', <String>**

:PLUGin:DPHYplugin:PATTern:HSData[:PATTern]? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', '0xE7, 0xE7, 0xE7, 0xE7'

Description This command defines the data for the Burst High Speed.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:HSData 'MIPI D-PHY Editor 1', '0xE7'

Query

:PLUGin:DPHYplugin:PATTern:HSData:PATTern? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:HSData:FILename[?]

Syntax **:PLUGin:DPHYplugin:PATTern:HSData:FILename 'identifier', <String>**

PLUGin:DPHYplugin:PATTern:HSData:FILename? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 'C:\hs.dat'

Description This command defines data file for the Burst High Speed.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:HSData:FILename 'MIPI D-PHY Editor 1', 'C:\hs.dat'

Query

:PLUGin:DPHYplugin:PATTern:HSData:FILename? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:LPData[:PATTern][?]

Syntax :PLUGin:DPHYplugin:PATTern:LPData[:PATTern 'identifier', <String>

:PLUGin:DPHYplugin:PATTern:LPData[:PATTern]? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', '0xE7, 0xE7, 0xE7, 0xE7'

Description This command defines data for the Burst Low Power.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:LPData:PATTern 'MIPI D-PHY Editor 1', '0xE7'

Query

:PLUGin:DPHYplugin:PATTern:LPData? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:LPData:FILename[?]

Syntax :PLUGin:DPHYplugin:PATTern:LPData:FILename 'identifier', <String>

:PLUGin:DPHYplugin:PATTern:LPData:FILename? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 'C:\lp.dat'

Description This command defines data file for the Burst Low Power.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:LPData:FILename 'MIPI D-PHY Editor 1', C:\lp.dat

Query

:PLUGin:DPHYplugin:PATTern:LPData:FILename? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PATTern:SEQuence:FILename[?]

Syntax :PLUGin:DPHYplugin:PATTern:SEQuence:FILename 'identifier', <String>

:PLUGin:DPHYplugin:PATTern:SEQuence:FILename? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 'C:\compliance.seq'

Description This command sequence file for the Frame.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PATTern:SEQuence:FILename ‘MIPI D-PHY Editor 1’, ‘C:\compliance.seq’

Query

:PLUGin:DPHYplugin:PATTern:SEQuence:FILename? ‘MIPI D-PHY Editor 1’

SCPI Commands for Data Rate and Transition Time Group

Table 8 SCPI Commands for Data Rate and Transition Time Group

Command	Description under
:PLUGin:DPHYplugin:HSMode:DATarate[?]	For details, see **:PLUGin:DPHYplugin:HSMode:DATarate[?]** on page 143.
:PLUGin:DPHYplugin:HSMode:TTCValue[?]	For details, see **:PLUGin:DPHYplugin:HSMode:TTCValue[?]** on page 143.
:PLUGin:DPHYplugin:LPMode:DATarate[?]	For details, see **:PLUGin:DPHYplugin:LPMode:DATarate[?]** on page 144.
:PLUGin:DPHYplugin:LPMode:DATarate[?]	For details, see:**PLUGin:DPHYplugin:LPMode:TTCValue[?]** on page 144.

:PLUGin:DPHYplugin:HSMode:DATarate[?]

Syntax :PLUGin:DPHYplugin:HSMode:DATarate 'identifier', <Double>

:PLUGin:DPHYplugin:HSMode:DATarate? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 800e6

Description This command defines High Speed data rate.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:HSMode:DATarate 'MIPI D-PHY Editor 1', 800e6

Query

:PLUGin:DPHYplugin:HSMode:DATarate? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:HSMode:TTCValue[?]

Syntax :PLUGin:DPHYplugin:HSMode:TTCValue 'identifier', <Double>

:PLUGin:DPHYplugin:HSMode:TTCValue? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 82e-12

Description This command defines High Speed Transition Time Converter.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:HSMode:TTCValue 'MIPI D-PHY Editor 1', 82e-12

Query

:PLUGin:DPHYplugin:HSMode:TTCValue? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LPMode:DATarate[?]

Syntax :PLUGin:DPHYplugin:LPMode:DATarate 'identifier', <Double>

:PLUGin:DPHYplugin:LPMode:DATarate? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 10e6

Description This command defines Low Power Data Rate.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LPMode:DATarate 'MIPI D-PHY Editor 1', 10e6

Query

:PLUGin:DPHYplugin:LPMode:DATarate? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LPMode:TTCValue[?]

Syntax :PLUGin:DPHYplugin:LPMode:TTCValue 'identifier', <Double>

:PLUGin:DPHYplugin:LPMode:TTCValue? 'identifier', <Double>

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 2e-9

Description This command defines Low Power Transition Time Converter.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LPMode:TTCValue 'MIPI D-PHY Editor 1', 2e-9

Query

:PLUGin:DPHYplugin:LPMode:TTCValue? 'MIPI D-PHY Editor 1'

SCPI Command for Signal Levels Group

Table 9 SCPI Commands for Signal Levels Group

Command	Description under
:PLUGin:DPHYplugin:LEVELs:CALibrated[?]	For details, see **:PLUGin:DPHYplugin:LEVELs:CALibrated[?]** on page 145.
:PLUGin:DPHYplugin:LEVELs:DEEMphasis[:VALue][?]	For details, see **:PLUGin:DPHYplugin:LEVELs:DEEMphasis[:VALue][?]** on page 145.
:PLUGin:DPHYplugin:LEVELs:LANe[?]	For details, see **:PLUGin:DPHYplugin:LEVELs:LANe[?]** on page 146.
:PLUGin:DPHYplugin:LEVELs:SIGNal[?]	For details, see**:PLUGin:DPHYplugin:LEVELs:SIGNal[?]** on page 146
:PLUGin:DPHYplugin:LEVELs:HSMode:AMPLitude[?]	For details, see **:PLUGin:DPHYplugin:LEVELs:HSMode:AMPLitude[?]** on page 146.
:PLUGin:DPHYplugin:LEVELs:HSMode:OFFSet[?]	For details, see **:PLUGin:DPHYplugin:LEVELs:HSMode:OFFSet[?]** on page 147.
:PLUGin:DPHYplugin:LEVELs:LPMode:LOW[?]	For details, see **:PLUGin:DPHYplugin:LEVELs:LPMode:LOW[?]** on page 147.
:PLUGin:DPHYplugin:LEVELs:LPMode:HIGH[?]	For details, see **:PLUGin:DPHYplugin:LEVELs:LPMode:HIGH[?]** on page 147.

:PLUGin:DPHYplugin:LEVELs:CALibrated[?]

Syntax :PLUGin:DPHYplugin:LEVELs:CALibrated ‘identifier’, <Boolean>

:PLUGin:DPHYplugin:LEVELs:CALibrated? ‘identifier’

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’, 1 | 0

Description This command activates the calibrated signal level values.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LEVELs:CALibrated ‘MIPI D-PHY Editor 1’, 1

Query

:PLUGin:DPHYplugin:LEVELs:CALibrated? ‘MIPI D-PHY Editor 1’

:PLUGin:DPHYplugin:LEVELs:DEEMphasis[:VALue][?]

Syntax :PLUGin:DPHYplugin:LEVELs:DEEMphasis[:VALue] 'pluginidentifier', <parameter>

:PLUGin:DPHYplugin:LEVELs:DEEMphasis[:VALue]? 'pluginidentifier'

Parameters De-Emphasis value in dB (double)

Description This command is used to set the De-Emphasis value.

This query returns the present setting.

Example :PLUGin:DPHYplugin:LEVELs:DEEMphasis:VALue 'MIPI D-PHY Editor 1', -5.00000E+00

:PLUGin:DPHYplugin:LEVELs:DEEMphasis:VALue? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LEVELs:LANe[?]

Syntax **:PLUGin:DPHYplugin:LEVELs:LANe 'identifier', <Enum>**

:PLUGin:DPHYplugin:LEVELs:LANe? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', ALL | CLOCk | ALLData | DAT0 | DAT1 | DAT2 | DAT3

Description This command defines the lane where levels are applied.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LEVELs:LANe 'MIPI D-PHY Editor 1', DAT0

Query

:PLUGin:DPHYplugin:LEVELs:LANe? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LEVELs:SIGNal[?]

Syntax **:PLUGin:DPHYplugin:LEVELs:SIGNal 'identifier', <Enum>**

:PLUGin:DPHYplugin:LEVELs:SIGNal? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', BOTH | NORMal | COMPlement

Description This command defines the lane pair where levels are applied.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LEVELs:SIGNal 'MIPI D-PHY Editor 1', NORMal

Query

:PLUGin:DPHYplugin:LEVELs:SIGNal? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LEVELs:HSMode:AMPLitude[?]

Syntax **:PLUGin:DPHYplugin:LEVELs:HSMode:AMPLitude 'identifier', <Double>**

:PLUGin:DPHYplugin:LEVELs:HSMode:AMPLitude? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0.2

Description This command defines the differential amplitude of the High Speed signal.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LEVELs:HSMode:AMPLitude 'MIPI D-PHY Editor 1', 0.2

Query

:PLUGin:DPHYplugin:LEVELs:HSMode:AMPLitude? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LEVELs:HSMode:OFFSet[?]

Syntax **:PLUGin:DPHYplugin:LEVELs:HSMode:OFFSet 'identifier', <Double>**

:PLUGin:DPHYplugin:LEVELs:HSMode:OFFSet? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0.1

Description This command defines the differential offset of the High Speed signal.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LEVELs:HSMode:OFFSet 'MIPI D-PHY Editor 1', 0.1

Query

:PLUGin:DPHYplugin:LEVELs:HSMode:OFFSet? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LEVELs:LPMode:LOW[?]

Syntax **:PLUGin:DPHYplugin:LEVELs:LPMode:LOW 'identifier', <Double>**

:PLUGin:DPHYplugin:LEVELs:LPMode:LOW? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0.1

Description This command defines the low voltage level for the Lower Power Signal.

This query returns the present setting

Example Command

:PLUGin:DPHYplugin:LEVELs:LPMode:LOW 'MIPI D-PHY Editor 1', 0.1

Query

:PLUGin:DPHYplugin:LEVELs:LPMode:LOW? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:LEVELs:LPMode:HIGH[?]

Syntax **:PLUGin:DPHYplugin:LEVELs:LPMode:HIGH 'identifier', <Double>**

:PLUGin:DPHYplugin:LEVELs:LPMode:HIGH? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1.0

Description This command defines the high voltage level for the Lower Power Signal.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LEVELs:LPMode:HIGH ‘MIPI D-PHY Editor 1’, 1.0

Query

:PLUGin:DPHYplugin:LEVELs:LPMode:HIGH? ‘MIPI D-PHY Editor 1’

SCPI Commands for Protocol Timings Group

Table 10 SCPI Commands for Protocol Timings Group

Command	Description under
:PLUGin:DPHYplugin:PROTocol:CMISs:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:CMISs:DURation[?]** on page 149.
:PLUGin:DPHYplugin:PROTocol:CPOSt:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:CPOSt:DURation[?]** on page 150.
:PLUGin:DPHYplugin:PROTocol:CPREpare:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:CPREpare:DURation[?]** on page 150.
:PLUGin:DPHYplugin:PROTocol:CPRLength:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:CPRLength:DURation[?]** on page 150.
:PLUGin:DPHYplugin:PROTocol:CTRail:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:CTRail:DURation[?]** on page 151.
:PLUGin:DPHYplugin:PROTocol:CZERo:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:CZERo:DURation[?]** on page 152.
:PLUGin:DPHYplugin:PROTocol:EXIT:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:EXIT:DURation[?]** on page 152.
:PLUGin:DPHYplugin:PROTocol:INIT:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:INIT:DURation[?]** on page 152.
:PLUGin:DPHYplugin:PROTocol:ISKew:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:ISKew:DURation[?]** on page 153.
:PLUGin:DPHYplugin:PROTocol:ALTCal:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:ALTCal:DURation[?]** on page 153.
:PLUGin:DPHYplugin:PROTocol:PERSKew:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:PERSKew:DURation[?]** on page 153.
:PLUGin:DPHYplugin:PROTocol:IDLPost:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:IDLPost:DURation[?]** on page 154.
:PLUGin:DPHYplugin:PROTocol:IDLClk:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:IDLClk:DURation[?]** on page 154.
:PLUGin:DPHYplugin:PROTocol:IDLPre:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:IDLPre:DURation[?]** on page 155.
:PLUGin:DPHYplugin:PROTocol:PRSTep:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:PRSTep:DURation[?]** on page 155.
:PLUGin:DPHYplugin:PROTocol:PREPare:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:PREPare:DURation[?]** on page 155.
:PLUGin:DPHYplugin:PROTocol:TRAIl:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:TRAIl:DURation[?]** on page 156.
:PLUGin:DPHYplugin:PROTocol:WAKeup:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:WAKeup:DURation[?]** on page 156.
:PLUGin:DPHYplugin:PROTocol:ZERo:DURation[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:ZERo:DURation[?]** on page 157.
:PLUGin:DPHYplugin:PROTocol:SET:MINimum[?]	For details, see **:PLUGin:DPHYplugin:PROTocol:SET:MINimum** on page 157.

:PLUGin:DPHYplugin:PROTocol:CMISs:DURation[?]

Syntax :PLUGin:DPHYplugin:PROTocol:CMISs:DURation 'identifier', <Double>

:PLUGin:DPHYplugin:PROTocol:CMISs:DURation? 'identifier

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 50e-9

Description This command defines Clock-Miss duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:CMISs:DURation 'MIPI D-PHY Editor 1', 50e-9

Query

:PLUGin:DPHYplugin:PROTocol:CMISs:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:CPOSt:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:CPOSt:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:CPOSt:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 50e-9

Description This command defines Clock-Post duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:CPOSt:DURation 'MIPI D-PHY Editor 1', 50e-9

Query

:PLUGin:DPHYplugin:PROTocol:CPOSt:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:CPREpare:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:CPREpare:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:CPREpare:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1',

Description This command defines Clock-Pre duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:CPREpare:DURation 'MIPI D-PHY Editor 1', 80e-9

Query

:PLUGin:DPHYplugin:PROTocol:CPREpare:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:CPRLength:DURation[?]

Syntax :PLUGin:DPHYplugin:PROTocol:CPRLength:DURation 'identifier', <Double>

:PLUGin:DPHYplugin:PROTocol:CPRLength:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 10e-9

Description This command

Example Command

:PLUGin:DPHYplugin:PROTocol:CPRLength:DURation 'MIPI D-PHY Editor 1', 10e-9

Query

:PLUGin:DPHYplugin:PROTocol:CPRLength:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:CTRail:DURation[?]

Syntax :PLUGin:DPHYplugin:PROTocol:CTRail:DURation 'identifier', <Double>

:PLUGin:DPHYplugin:PROTocol:CTRail:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 50e-9

Description This command defines Clock-Trail duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:CTRail:DURation 'MIPI D-PHY Editor 1', 50e-9

Query

:PLUGin:DPHYplugin:PROTocol:CTRail:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:CZERo:DURation[?]

Syntax :PLUGin:DPHYplugin:PROTocol:CZERo:DURation 'identifier', <Double>

:PLUGin:DPHYplugin:PROTocol:CZERo:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 80e-9

Description This command defines Clock-Zero duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:CZERo:DURation 'MIPI D-PHY Editor 1', 80e-9

Query

:PLUGin:DPHYplugin:PROTocol:CZERo:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:EXIT:DURation[?]

Syntax :PLUGin:DPHYplugin:PROTocol:EXIT:DURation 'identifier', <Double>

:PLUGin:DPHYplugin:PROTocol:EXIT:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 200e-9

Description This command defines TX-HS-Exit duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:EXIT:DURation 'MIPI D-PHY Editor 1', 200e-9

Query

:PLUGin:DPHYplugin:PROTocol:EXIT:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:INIT:DURation[?]

Syntax :PLUGin:DPHYplugin:PROTocol:INIT:DURation 'identifier', <Double>

:PLUGin:DPHYplugin:PROTocol:INIT:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 100e-6

Description This command defines TX-Init duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:INIT:DURation 'MIPI D-PHY Editor 1', 100e-6

Query

:PLUGin:DPHYplugin:PROTocol:INIT:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:ISKew:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:ISKew:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:ISKew:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 100e-6

Description This command defines TX-Init Skew Cal duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:ISKew:DURation 'MIPI D-PHY Editor 1', 100e-6

Query

:PLUGin:DPHYplugin:PROTocol:ISKew:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:ALTCal:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:ALTCal:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:ALTCal:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', Range from 0 ps to 100 µs

Description This command defines the duration when the transmitter drives Alternate Calibration Pattern in the Alternate Calibration mode.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:ALTCal:DURation 'MIPI D-PHY Editor 1', 3.27680E-05

Query

:PLUGin:DPHYplugin:PROTocol:ALTCal:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:PERSKew:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:PERSKew:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:PERSKew:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 100e-6

Description This command defines TX-Periodic Skew Cal duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:PERSKew:DURation 'MIPI D-PHY Editor 1', 100e-6

Query

:PLUGin:DPHYplugin:PROTocol:PERSKew:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:IDLPost:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:IDLPost:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:IDLPost:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', Interval range from 0 UI to 1000 UI

Description This command defines the duration when the transmitting signal transitions from Data Burst to HS Idle state.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:IDLPost:DURation 'MIPI D-PHY Editor 1', 2.560e+002

Query

:PLUGin:DPHYplugin:PROTocol:IDLPost:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:IDLClk:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:IDLClk:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:IDLClk:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', Duration range from 0 ps to 10 ms

Description This command defines the duration when the transmitting signal drives in HS-Zero state on Clock lane.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:IDLClk:DURation 'MIPI D-PHY Editor 1', 6.000e-008

Query

:PLUGin:DPHYplugin:PROTocol:IDLClk:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:IDLPre:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:IDLPre:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:IDLPre:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', Interval range from 0 UI to 1000 UI

Description This command defines the duration when the transmitting signal transitions from HS Idle state to next Data Burst state.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:IDLPre:DURation 'MIPI D-PHY Editor 1', 4.800e+001

Query

:PLUGin:DPHYplugin:PROTocol:IDLPre:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:PRSTep:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:PRSTep:DURation 'identifier', <numeric>**

:PLUGin:DPHYplugin:PROTocol:PRSTep:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', Range of numbers from 1 to 255

Description This command defines the required preamble steps to be inserted into the HS Burst. Each step is 32 UI in length. The parameter sets the multiplier value, which is multiplied by 32 UI, and the preamble sequence length is configured.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:PRSTep:DURation 'MIPI D-PHY Editor 1', 2

Query

:PLUGin:DPHYplugin:PROTocol:PRSTep:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:PREPare:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:PREPare:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:PREPare:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 50e-9

Description This command defines TX-Prepare duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:PREPare:DURation 'MIPI D-PHY Editor 1', 50e-9

Query

:PLUGin:DPHYplugin:PROTocol:PREPare:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:TRAIl:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:TRAIl:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:TRAIl:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 100e-9

Description This command defines TX-Trail duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:TRAIl:DURation 'MIPI D-PHY Editor 1', 100e-9

Query

:PLUGin:DPHYplugin:PROTocol:TRAIl:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:WAKeup:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:WAKeup:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:WAkeup:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 100e-3

Description This command defines TX-Wakeup duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:WAKeup:DURation 'MIPI D-PHY Editor 1', 100e-3

Query

:PLUGin:DPHYplugin:PROTocol:WAKeup:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:ZERo:DURation[?]

Syntax **:PLUGin:DPHYplugin:PROTocol:ZERo:DURation 'identifier', <Double>**

:PLUGin:DPHYplugin:PROTocol:ZERo:DURation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1',

Description This command defines TX-Zero duration.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:PROTocol:ZERo:DURation 'MIPI D-PHY Editor 1', 100e-9

Query

:PLUGin:DPHYplugin:PROTocol:ZERo:DURation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:PROTocol:SET:MINimum

Syntax **:PLUGin:DPHYplugin:PROTocol:SET:MINimum 'identifier'**

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command sets the minimum timing values for each parameter under Protocol Timings. Press the <Enter> key after typing the command.

Example Command

:PLUGin:DPHYplugin:PROTocol:SET:MINimum 'MIPI D-PHY Editor 1'

SCPI Commands for Jitter Group

Table 11 SCPI Commands for Jitter Group

Command	Description under
:PLUGin:DPHYplugin:RJitter:ENABled[?]	For details, see **:PLUGin:DPHYplugin:RJitter:ENABled[?]** on page 158.
:PLUGin:DPHYplugin:RJitter:CALibrated[?]	For details, see **:PLUGin:DPHYplugin:RJitter:CALibrated[?]** on page 158.
:PLUGin:DPHYplugin:RJitter:LANe[?]	For details, see **:PLUGin:DPHYplugin:RJitter:LANe[?]** on page 159.
:PLUGin:DPHYplugin:RJitter:STDDev[?]	For details, see **:PLUGin:DPHYplugin:RJitter:STDDev[?]** on page 159.
:PLUGin:DPHYplugin:SJitter:ENABled[?]	For details, see **:PLUGin:DPHYplugin:SJitter:ENABled[?]** on page 159.
:PLUGin:DPHYplugin:SJitter:CALibrated[?]	For details, see **:PLUGin:DPHYplugin:SJitter:CALibrated[?]** on page 160.
:PLUGin:DPHYplugin:SJitter:LANe[?]	For details, see **:PLUGin:DPHYplugin:SJitter:LANe[?]** on page 160
:PLUGin:DPHYplugin:SJitter:AMPLitude[?]	For details, see **:PLUGin:DPHYplugin:SJitter:AMPLitude[?]** on page 161 .
:PLUGin:DPHYplugin:SJitter:FREQuency[?]	For details, see **:PLUGin:DPHYplugin:SJitter:FREQuency[?]** on page 161.

:PLUGin:DPHYplugin:RJitter:ENABled[?]

Syntax :PLUGin:DPHYplugin:RJitter:ENABled 'identifier', <Boolean>

:PLUGin:DPHYplugin:RJitter:ENABled? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command enables the random jitter.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:RJitter:ENABled 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:RJitter:ENABled? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:RJitter:CALibrated[?]

Syntax :PLUGin:DPHYplugin:RJitter:CALibrated 'identifier', <Boolean>

:PLUGin:DPHYplugin:RJitter:CALibrated? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command activates calibrated values of the random jitter.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:RJitter:CALibrated 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:RJitter:CALibrated? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:RJitter:LANe[?]

Syntax **:PLUGin:DPHYplugin:RJitter:LANe 'identifier', <Enum>**

Parameter 'Identifier': 'MIPI D-PHY Editor 1', CLOCk | ALLData | DAT0 | DAT1 | DAT2 | DAT3

Description This command defines lane where the random jitter is applied.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:RJitter:LANe 'MIPI D-PHY Editor 1', DAT0

Query

:PLUGin:DPHYplugin:RJitter:LANe? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:RJitter:STDDev[?]

Syntax **:PLUGin:DPHYplugin:RJitter:STDDev 'identifier', <Double>**

:PLUGin:DPHYplugin:RJitter:STDDev? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0.01

Description This command defines random jitter amplitude in RMS.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:RJitter:STDDev 'MIPI D-PHY Editor 1', 0.01

Query

:PLUGin:DPHYplugin:RJitter:STDDev? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:SJitter:ENABled[?]

Syntax **:PLUGin:DPHYplugin:SJitter:ENABled 'identifier', <Boolean>**

:PLUGin:DPHYplugin:SJitter:ENABled? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command enables the sinusoidal jitter.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:SJitter:ENABled 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:SJitter:ENABled? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:SJitter:CALibrated[?]

Syntax **:PLUGin:DPHYplugin:SJitter:CALibrated 'identifier', <Boolean>**

:PLUGin:DPHYplugin:SJitter:CALibrated? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command activates calibrated values of the sinusoidal jitter.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:SJitter:CALibrated 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:SJitter:CALibrated? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:SJitter:LANe[?]

Syntax **:PLUGin:DPHYplugin:SJitter:LANe 'identifier', <Enum>**

:PLUGin:DPHYplugin:SJitter:LANe? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', CLOCk | ALLData | DAT0 | DAT1 | DAT2 | DAT3

Description This command defines lane where the sinusoidal jitter is applied.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:SJitter:LANe 'MIPI D-PHY Editor 1', ALLData

Query

:PLUGin:DPHYplugin:SJitter:LANe? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:SJitter:AMPLitude[?]

Syntax :PLUGin:DPHYplugin:SJitter:AMPLitude 'identifier', <Double>

:PLUGin:DPHYplugin:SJitter:AMPLitude? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0.1

Description This command defines the amplitude of the sinusoidal jitter.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:SJitter:AMPLitude 'MIPI D-PHY Editor 1', 0.1

Query

:PLUGin:DPHYplugin:SJitter:AMPLitude? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:SJitter:FREQuency[?]

Syntax :PLUGin:DPHYplugin:SJitter:FREQuency 'identifier', <Double>

:PLUGin:DPHYplugin:SJitter:FREQuency? 'identifier', <Double>

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 100e6

Description This command defines the frequency of the sinusoidal jitter.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:SJitter:FREQuency 'MIPI D-PHY Editor 1', 100e6

Query

:PLUGin:DPHYplugin:SJitter:FREQuency? 'MIPI D-PHY Editor 1'

SCPI Commands for Intersymbol Interference Group

Table 12 SCPI Commands for Intersymbol Interference Group

Command	Description under
:PLUGin:DPHYplugin:ISI:ENABled[?]	For details, see **:PLUGin:DPHYplugin:ISI:ENABled[?]** on page 162.
:PLUGin:DPHYplugin:ISI:SPFile[?]	For details, see:**PLUGin:DPHYplugin:ISI:SPFile[?]** on page 162.
::PLUGin:DPHYplugin:ISI:ATTenuation[?]	For details, see **:PLUGin:DPHYplugin:ISI:ATTenuation[?]** on page 163.
:PLUGin:DPHYplugin:ISI:TSCaling[?]	For details, see:**PLUGin:DPHYplugin:ISI:TSCaling[?]** on page 163.

:PLUGin:DPHYplugin:ISI:ENABled[?]

Syntax **:PLUGin:DPHYplugin:ISI:ENABled 'identifier', <Boolean>**

:PLUGin:DPHYplugin:ISI:ENABled? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command enables the ISI.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:ISI:ENABled 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:ISI:ENABled? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:ISI:SPFile[?]

Syntax **:PLUGin:DPHYplugin:ISI:SPFile 'identifier', <String>**

:PLUGin:DPHYplugin:ISI:SPFile? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 'C:\ SParam.s2p'

Description This command sets the ISI S-Parameter (S2P).file.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:ISI:SPFile 'MIPI D-PHY Editor 1', 'C:\ SParam.s2p'

Query

:PLUGin:DPHYplugin:ISI:SPFile? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:ISI:ATTenuation[?]

Syntax :PLUGin:DPHYplugin:ISI:ATTenuation 'identifier', <Double>

:PLUGin:DPHYplugin:ISI:ATTEnuation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1

Description This command changes the attenuation of the S-Parameter values.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:ISI:ATTenuation 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:ISI:ATTenuation? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:ISI:TSCaling[?]

Syntax :PLUGin:DPHYplugin:ISI:TSCaling 'identifier', <Double>

:PLUGin:DPHYplugin:ISI:TSCaling? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1

Description This command changes the time scale of the S-Parameter values.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:ISI:TSCaling 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:ISI:TSCaling? 'MIPI D-PHY Editor 1'

SCPI Commands for Disturbances Group

Table 13 SCPI Commands for Disturbances Group

Command	Description under
:PLUGin:DPHYplugin:LPPulse[?]	For details, see **:PLUGin:DPHYplugin:LPPulse[?]** on page 164.
:PLUGin:DPHYplugin:ESPike:MODe[?]	For details, see **:PLUGin:DPHYplugin:ESPike:MODe[?]** on page 165 .
:PLUGin:DPHYplugin:ESPike:CALibrated[?]	For details, see:**PLUGin:DPHYplugin:ESPike:CALibrated[?]** on page 165 .
:PLUGin:DPHYplugin:ESPike:AREa[?]	For details, see **:PLUGin:DPHYplugin:ESPike:AREa[?]** on page 165.
:PLUGin:DPHYplugin:ESPike:LHIGhinvol[?]	For details, see **:PLUGin:DPHYplugin:ESPike:LHIGhinvol[?]** on page 166.
:PLUGin:DPHYplugin:ESPike:LLOWintvol[?]	For details, see **:PLUGin:DPHYplugin:ESPike:LLOWintvol[?]** on page 166.
:PLUGin:DPHYplugin:DISTurbances:ENABled[?]	For details, see **:PLUGin:DPHYplugin:DISTurbances:ENABled[?]** on page 166.
:PLUGin:DPHYplugin:DISTurbances:LANe[?]	For details, see **:PLUGin:DPHYplugin:DISTurbances:LANe[?]** on page 167.
:PLUGin:DPHYplugin:DISTurbances:IPATtern[?]	For details, see **:PLUGin:DPHYplugin:DISTurbances:IPATtern[?]** on page 167.
:PLUGin:DPHYplugin:DISTurbances:EPATtern[?]	For details, see **:PLUGin:DPHYplugin:DISTurbances:EPATtern[?]** on page 168.
:PLUGin:DPHYplugin:EMBedding:ENABled[?]	For details, see **:PLUGin:DPHYplugin:EMBedding:ENABled[?]** on page 168.
:PLUGin:DPHYplugin:INSCal:ENABled[?]	For details, see **:PLUGin:DPHYplugin:INSCal:ENABled[?]** on page 168.
:PLUGin:DPHYplugin:EMBedding:PATH:SPArameters[?]	For details, see **:PLUGin:DPHYplugin:EMBedding:PATH:SPArameters[?]** on page 169.

:PLUGin:DPHYplugin:LPPulse[?]

Syntax :PLUGin:DPHYplugin:LPPulse ‘identifier’, <Double>

:PLUGin:DPHYplugin:LPPulse? ‘identifier’

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’, 50e-9

Description This command defines Low Power pulse width.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:LPPulse ‘MIPI D-PHY Editor 1’, 50e-9

Query

:PLUGin:DPHYplugin:LPPulse? ‘MIPI D-PHY Editor 1’

:PLUGin:DPHYplugin:ESPike:MODe[?]

Syntax :PLUGin:DPHYplugin:ESPike:MODe 'identifier', <Enum>

:PLUGin:DPHYplugin:ESPike:MODe? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', OFF | LLEVels | HLEVels

Description This command defines eSpike mode.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:ESPike:MODe 'MIPI D-PHY Editor 1', HLEVels

Query

:PLUGin:DPHYplugin:ESPike:MODe? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:ESPike:CALibrated[?]

Syntax :PLUGin:DPHYplugin:ESPike:CALibrated 'identifier', <Boolen>

:PLUGin:DPHYplugin:ESPike:CALibrated? 'identifier',

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command activates calibrated values of the eSpike mode.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:ESPike:CALibrated 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:ESPike:CALibrated? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:ESPike:AREa[?]

Syntax :PLUGin:DPHYplugin:ESPike:AREa 'identifier', <Double>

:PLUGin:DPHYplugin:ESPike:AREa? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 125e-12

Description This command defines area of the eSpike.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:ESPike:AREa 'MIPI D-PHY Editor 1', 125e-12

Query

:PLUGin:DPHYplugin:ESPike:AREa? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:ESPike:LHIGhinvol[?]

Syntax **:PLUGin:DPHYplugin:ESPike:LHIGhinvol 'identifier', <Double>**

:PLUGin:DPHYplugin:ESPike:LHIGhinvol? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0.740

Description This command defines Logic 1 threshold.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:ESPike:LHIGhinvol 'MIPI D-PHY Editor 1', 0.740

Query

:PLUGin:DPHYplugin:ESPike:LHIGhinvol? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:ESPike:LLOWintvol[?]

Syntax **:PLUGin:DPHYplugin:ESPike:LLOWintvol 'identifier', <Double>**

:PLUGin:DPHYplugin:ESPike:LLOWintvol? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0.100

Description This command defines Logic 0 threshold.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:ESPike:LLOWintvol 'MIPI D-PHY Editor 1', 0.100

Query

:PLUGin:DPHYplugin:ESPike:LLOWintvol? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:DISTurbances:ENABled[?]

Syntax **:PLUGin:DPHYplugin:DISTurbances:ENABled 'identifier', <Boolean>**

:PLUGin:DPHYplugin:DISTurbances:ENABled? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command enables High Speed mode pattern disturbance.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:DISTurbances:ENABled 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:DISTurbances:ENABled? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:DISTurbances:LANe[?]

Syntax **:PLUGin:DPHYplugin:DISTurbances:LANe 'identifier', <Enum>**

:PLUGin:DPHYplugin:DISTurbances:LANe? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', CLOCk | ALLData | DAT0 | DAT1 | DAT2 | DAT3

Description This command selects the lane where the High Speed mode disturbances is applied.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:DISTurbances:LANe 'MIPI D-PHY Editor 1', DAT0

Query

:PLUGin:DPHYplugin:DISTurbances:LANe? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:DISTurbances:IPATtern[?]

Syntax **:PLUGin:DPHYplugin:DISTurbances:IPATtern 'identifier', 'String'**

:PLUGin:DPHYplugin:DISTurbances:IPATtern? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', '10010'

Description This command defines the High Speed init pattern disturbance.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:DISTurbances:IPATtern 'MIPI D-PHY Editor 1', '10010'

Query

:PLUGin:DPHYplugin:DISTurbances:IPATtern? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:DISTurbances:EPATtern[?]

Syntax :PLUGin:DPHYplugin:DISTurbances:EPATtern ‘identifier’, ‘String’

:PLUGin:DPHYplugin:DISTurbances:EPATtern? ‘identifier’

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’, ‘10010’

Description This command defines the High Speed exit pattern disturbance.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:DISTurbances:EPATtern ‘MIPI D-PHY Editor 1’, ‘10010’

Query

:PLUGin:DPHYplugin:DISTurbances:EPATtern? ‘MIPI D-PHY Editor 1’

:PLUGin:DPHYplugin:EMBedding:ENABled[?]

Syntax :PLUGin:DPHYplugin:EMBedding:ENABled ‘identifier’, <Boolean>

:PLUGin:DPHYplugin:EMBedding:ENABled? ‘identifier’

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’, ON | OFF | 1 | 0

Description This command enables the Use Deembedding parameter for signal disturbance.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:EMBedding:ENABled 'MIPI D-PHY Editor 1', ON

Query

:PLUGin:DPHYplugin:EMBedding:ENABled? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:INSCal:ENABled[?]

Syntax :PLUGin:DPHYplugin:INSCal:ENABled ‘identifier’, <Boolean>

:PLUGin:DPHYplugin:INSCal:ENABled? ‘identifier’

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’, ON | OFF | 1 | 0

Description This command enables the In-System Calibration Enabled parameter for signal disturbance, with the Use Deembedding parameter active.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:INSCal:ENABled 'MIPI D-PHY Editor 1', ON

Query

:PLUGin:DPHYplugin:INSCal:ENABled? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:EMBedding:PATH:SPArameters[?]

Syntax **:PLUGin:DPHYplugin:EMBedding:PATH:SPArameters 'identifier', "<parameter>"**

:PLUGin:DPHYplugin:EMBedding:PATH:SPArameters? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', "file path"

Description This command helps you to define either the s-parameter (s2p) files or data (dat) files to de-embed the Output Amplifier from the AWG. You may define more than one file path locations, separated by a semi-colon (;)".

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:EMBedding:PATH:SPArameters 'MIPI D-PHY Editor 1', "C:\ProgramData\BitifEye\ValiFrame\SParameter\DPhy\DPHY_LegacyChannel.s2p;C:\ProgramData\BitifEye\ValiFrame\SParameter\DPhy\DPHY_StandardISIChannel.s2p"

Query

:PLUGin:DPHYplugin:INSCal:ENABled? 'MIPI D-PHY Editor 1'

SCPI Commands for Signal Interference Group

Table 14 SCPI Commands for Signal Interference Group

Command	Description under
:PLUGin:DPHYplugin:INTerference:ENABled[?]	For details, see **:PLUGin:DPHYplugin:INTerference:ENABled[?]** on page 170.
:PLUGin:DPHYplugin:INTerference:ENABled[?]	For details, see **:PLUGin:DPHYplugin:INTerference:LANe[?]** on page 170.
:PLUGin:DPHYplugin:INTerference:PAIR[?]	For details, see **:PLUGin:DPHYplugin:INTerference:PAIR[?]** on page 171.
:PLUGin:DPHYplugin:INTerference:MODe[?]	For details, see **:PLUGin:DPHYplugin:INTerference:MODe[?]** on page 171.
:PLUGin:DPHYplugin:INTerference:AMPLitude[?]	For details, see **:PLUGin:DPHYplugin:INTerference:AMPLitude[?]** on page 171.
:PLUGin:DPHYplugin:INTerference:FREQuency[?]	For details, see **:PLUGin:DPHYplugin:INTerference:FREQuency[?]** on page 172.

:PLUGin:DPHYplugin:INTerference:ENABled[?]

Syntax :PLUGin:DPHYplugin:INTerference:ENABled 'identifier', <Boolean>

:PLUGin:DPHYplugin:INTerference:ENABled? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 1 | 0

Description This command enables the signal interference.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:INTerference:ENABled 'MIPI D-PHY Editor 1', 1

Query

:PLUGin:DPHYplugin:INTerference:ENABled? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:INTerference:LANe[?]

Syntax :PLUGin:DPHYplugin:INTerference:LANe 'identifier', <Enum>

:PLUGin:DPHYplugin:INTerference:LANe? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', CLOCk | ALLData | DAT0 | DAT1 | DAT2 | DAT3

Description This command defines the Lane where signal interference is applied.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:INTerference:LANe 'MIPI D-PHY Editor 1', DAT2

Query

:PLUGin:DPHYplugin:INTerference:LANe? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:INTerference:PAIR[?]

Syntax **:PLUGin:DPHYplugin:INTerference:PAIR 'identifier', <Enum>**

:PLUGin:DPHYplugin:INTerference:PAIR? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', BOTH | NORMal | COMPlement

Description This command defines the Lane pair where signal interference is applied.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:INTerference:PAIR 'MIPI D-PHY Editor 1', NORMal |

Query

:PLUGin:DPHYplugin:INTerference:PAIR? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:INTerference:MODe[?]

Syntax **:PLUGin:DPHYplugin:INTerference:MODe 'identifier', <Enum>**

:PLUGin:DPHYplugin:INTerference:MODe? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', BOTH | LPower | HSpeed

Description This command defines that the Signal interference applies on Low Power, High Speed or both.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:INTerference:MODe 'MIPI D-PHY Editor 1', LPower

Query

:PLUGin:DPHYplugin:INTerference:MODe? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:INTerference:AMPLitude[?]

Syntax **:PLUGin:DPHYplugin:INTerference:AMPLitude 'identifier', <Double>**

:PLUGin:DPHYplugin:INTerference:AMPLitude? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0.1

Description This command defines Signal interference amplitude.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:INTerference:AMPLitude ‘MIPI D-PHY Editor 1’, 0.1

Query

:PLUGin:DPHYplugin:INTerference:AMPLitude? ‘MIPI D-PHY Editor 1’

:PLUGin:DPHYplugin:INTerference:FREQuency[?]

Syntax **:PLUGin:DPHYplugin:INTerference:FREQuency ‘identifier’, <Double>**

:PLUGin:DPHYplugin:INTerference:FREQuency? ‘identifier’

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’, 50e6

Description This command defines Signal interference frequency.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:INTerference:FREQuency ‘MIPI D-PHY Editor 1’, 50e6

Query

:PLUGin:DPHYplugin:INTerference:FREQuency? ‘MIPI D-PHY Editor 1’

SCPI Commands for Skew Group

Table 15 SCPI Commands for Skew Group

Command	Description under
:PLUGin:DPHYplugin:SKEW:AWGA:CH{1:4}[?]	For details, see :PLUGin:DPHYplugin:SKEW:AWGA:CH{1:4}[?] on page 173.
:PLUGin:DPHYplugin:SKEW:AWGA:CH{1:4}[?]	For details, see :PLUGin:DPHYplugin:SKEW:AWGB:CH{1:4}[?] on page 173.
:PLUGin:DPHYplugin:SKEW:AWGA:CH{1:4}[?] on page 135	For details, see :PLUGin:DPHYplugin:SKEW:AWGC:CH{1:4}[?] on page 174.

:PLUGin:DPHYplugin:SKEW:AWGA:CH{1:4}[?]

Syntax :PLUGin:DPHYplugin:SKEW:AWGA:CH{1:4} 'identifier', <Double>

:PLUGin:DPHYplugin:SKEW:AWGA:CH{1:4}? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 2e-12

Description This command defines AWG Module 1, Channel {1,4} Skew (Necessary for Intra-Pair Skew).

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:SKEW:AWGA:CH{1:4} 'MIPI D-PHY Editor 1', 2e-12

Query

:PLUGin:DPHYplugin:SKEW:AWGA:CH{1:4}? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:SKEW:AWGB:CH{1:4}[?]

Syntax :PLUGin:DPHYplugin:SKEW:AWGB:CH{1:4} 'identifier', <Double>

:PLUGin:DPHYplugin:SKEW:AWGB:CH{1:4}? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 2e-12

Description This command defines AWG Module 2, Channel {1,4} Skew (Necessary for Intra-Pair Skew).

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:SKEW:AWGB:CH{1:4} 'MIPI D-PHY Editor 1', 2e-12

Query

:PLUGin:DPHYplugin:SKEW:AWGB:CH{1:4}? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:SKEW:AWGC:CH{1:4}[?]

Syntax :PLUGin:DPHYplugin:SKEW:AWGC:CH{1:4} ‘identifier’, <Double>

:PLUGin:DPHYplugin:SKEW:AWGC:CH{1:4}? ‘identifier’

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’, 2e-12

Description This command defines AWG Module 3, Channel {1,4} Skew (Necessary for Intra-Pair Skew).

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:SKEW:AWGC:CH{1:4} ‘MIPI D-PHY Editor 1’, 2e-12

Query

:PLUGin:DPHYplugin:SKEW:AWGC:CH{1:4}? ‘MIPI D-PHY Editor 1’

SCPI Commands for Delay Group

Table 16 SCPI Commands for Delay Group

Command	Description under
:PLUGin:DPHYplugin:DELAy:CLKDataskew[?]	For details, see **:PLUGin:DPHYplugin:DELAy:CLKDataskew[?]** on page 175.
:PLUGin:DPHYplugin:DELAy:DAT{1:3}[?]	For details, see **:PLUGin:DPHYplugin:DELAy:DAT{1:3}[?]** on page 175.
:PLUGin:DPHYplugin:APPLy:DELays	For details, see **:PLUGin:DPHYplugin:APPLy:DELays** on page 176.

:PLUGin:DPHYplugin:DELAy:CLKDataskew[?]

Syntax **:PLUGin:DPHYplugin:DELAy:CLKDataskew 'identifier', <Double>**

:PLUGin:DPHYplugin:DELAy:CLKDataskew? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 0.5

Description This command defines clock to data skew.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:DELAy:CLKDataskew 'MIPI D-PHY Editor 1', 0.5

Query

:PLUGin:DPHYplugin:DELAy:CLKDataskew? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:DELAy:DAT{1:3}[?]

Syntax **:PLUGin:DPHYplugin:DELAy:DAT{1:3} 'identifier', <Double>**

:PLUGin:DPHYplugin:DELAy:DAT{1:3}? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1', 2e-12

Description This command defines Delay between DATA {1:3} and DATA0 Lane.

This query returns the present setting.

Example Command

:PLUGin:DPHYplugin:DELAy:DAT{1:3} 'MIPI D-PHY Editor 1', 2e-12

Query

:PLUGin:DPHYplugin:DELAy:DAT{1:3}? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:APPLy:DELays

Syntax :PLUGin:DPHYplugin:APPLy:DELays ‘identifier’, <Boolean>

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’,

Description This command applies the delays and skew defined. It tries to apply using the M8190A/M8195A hardware resource, if not possible recalculates the patten.

Example Command

:PLUGin:DPHYplugin:APPLy:DELays ‘MIPI D-PHY Editor 1’

SCPI Commands for SSC Group

Table 17 SCPI Commands for SSC Group

Command	Description under
:PLUGin:DPHYplugin:SSC:FREQuency[?]	For details, see **:PLUGin:DPHYplugin:SSC:FREQuency[?]** on page 177.
:PLUGin:DPHYplugin:SSC:DEViation[?]	For details, see **:PLUGin:DPHYplugin:SSC:DEViation[?]** on page 177.

:PLUGin:DPHYplugin:SSC:FREQuency[?]

Syntax :PLUGin:DPHYplugin:SSC:FREQuency 'identifier'

:PLUGin:DPHYplugin:SSC:FREQuency? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Range 1 kHz to 50 kHz

Description This command sets the SSC frequency (in Hertz). During this process, the AWGs automatically stop and restart.

The query returns the present value.

Example Command

:PLUGin:DPHYplugin:SSC:FREQuency 'MIPI D-PHY Editor 1', 40e3

Query

:PLUGin:DPHYplugin:SSC:FREQuency? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:SSC:DEViation[?]

Syntax :PLUGin:DPHYplugin:SSC:DEViation 'identifier'

:PLUGin:DPHYplugin:SSC:DEViation? 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1',

Range Amplitude range is -0.6 % to 0 %.

Description This command sets the minimum SSC deviation (in percentage). During this process, the AWGs automatically stop and restart. With a value of 0 the SSC is switched off. The query returns the present value.

Example Command

:PLUGin:DPHYplugin:SSC:DEViation? 'MIPI D-PHY Editor 1', -0.5

Query

:PLUGin:DPHYplugin:SSC:DEViation? 'MIPI D-PHY Editor 1'

Other SCPI Commands

Table 18 Other SCPI Commands

Command	Description under
:PLUGin:DPHYplugin:CALibrate	For details, see **:PLUGin:DPHYplugin:CALibrate** on page 178.
:PLUGin:DPHYplugin:APPLy[:ALL]	For details, see **:PLUGin:DPHYplugin:APPLy[:ALL]** on page 178.
:PLUGin:DPHYplugin:ABORt	For details, see **:PLUGin:DPHYplugin:ABORt** on page 179.
:PLUGin:DPHYplugin:RESet	For details, see **:PLUGin:DPHYplugin:RESet** on page 179.
:PLUGin:DPHYplugin:RESTart	For details, see **:PLUGin:DPHYplugin:RESTart** on page 179.
:PLUGin:DPHYplugin:TRIGger[:EXECute]	For details, see **:PLUGin:DPHYplugin:TRIGger[:EXECute]** on page 179.
:PLUGin:DPHYplugin:STARt	For details, see **:PLUGin:DPHYplugin:STARt** on page 179.
:PLUGin:DPHYplugin:STOP	For details, see **:PLUGin:DPHYplugin:STOP** on page 181.
:PLUGin:DPHYplugin:RUN:STATus?	For details, see **:PLUGin:DPHYplugin:RUN:STATus?** on page 181.
:PLUGin:DPHYplugin:RUN:PROGress?	For details, see **:PLUGin:DPHYplugin:RUN:PROGress?** on page 181.
:PLUGin:DPHYplugin:RUN:MESSage?	For details, see **:PLUGin:DPHYplugin:RUN:MESSage?** on page 181.
:PLUGin:DPHYplugin:RUN:LOG?	For details, see **:PLUGin:DPHYplugin:RUN:LOG?** on page 182.

:PLUGin:DPHYplugin:CALibrate

Syntax **:PLUGin:DPHYplugin:CALibrate‘identifier’**

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’

Description This command calibrates the skew of the AWGs.

Example Command

:PLUGin:DPHYplugin:CALibrate ‘MIPI D-PHY Editor 1’

:PLUGin:DPHYplugin:APPLy[:ALL]

Syntax **:PLUGin:DPHYplugin:APPLy[:ALL] ‘identifier’**

Parameter ‘Identifier’: ‘MIPI D-PHY Editor 1’,

Description This command calculates the waveform based on the required settings, loads it to the AWGs and starts the AWGs.

Example Command

:PLUGin:DPHYplugin:APPLy:ALL ‘MIPI D-PHY Editor 1’

:PLUGin:DPHYplugin:ABORt

Syntax :PLUGin:DPHYplugin:ABORt 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1',

Description This command sends an abort (stop) request after the waveform generation procedure is applied.

Example Command

:PLUGin:DPHYplugin:ABORt 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:RESet

Syntax :PLUGin:DPHYplugin:RESet 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command resets the settings to the default values and automatically applies them.

Example Command

:PLUGin:DPHYplugin:RESet 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:RESTart

Syntax :PLUGin:DPHYplugin:RESTart 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command restarts the AWGs.

Example Command

:PLUGin:DPHYplugin:RESTart 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:TRIGger[:EXECute]

Syntax :PLUGin:DPHYplugin:TRIGger[:EXECute] 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command makes the AWGs jump to the next block, if the pattern with triggers applied on the AWGs.

Example Command

:PLUGin:DPHYplugin:TRIGger:EXECute 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:STARt

Syntax :PLUGin:DPHYplugin:STARt 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1'

Description This command starts the AWGs and enables the outputs.

Example Command

:PLUGin:DPHYplugin:STARt 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:STOP

Syntax :PLUGin:DPHYplugin:STOP 'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1',

Description This command stops the AWGs and disables the outputs.

Example Command

:PLUGin:DPHYplugin:STOP 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:RUN:STATus?

Syntax :PLUGin:DPHYplugin:RUN:STATus?'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1',

Description This query returns if some waveform calculation or calibration is currently occurring. If "1" is returned, it means that some operation is currently begin executed by the plugin. If "0" is returned, it means no operations is currently begin executed.

Example Query

:PLUGin:DPHYplugin:RUN:STATus? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:RUN:PROGress?

Syntax :PLUGin:DPHYplugin:RUN:PROGress?'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1',

Description The query returns the progress of the current active operation begin executed by the plugin (waveform calculation or calibration). The progress return is contained between 0 and 1.

Example Query

:PLUGin:DPHYplugin:RUN:PROGress? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:RUN:MESSage?

Syntax :PLUGin:DPHYplugin:RUN:MESSage?'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1',

Description The query returns the current status message of the plugin. The possible status message can be "Running", "Finished", "NotStarted", "Error" and "Stopped".

Example Query

:PLUGin:DPHYplugin:RUN:MESSage? 'MIPI D-PHY Editor 1'

:PLUGin:DPHYplugin:RUN:LOG?

Syntax :PLUGin:DPHYplugin:RUN:LOG?'identifier'

Parameter 'Identifier': 'MIPI D-PHY Editor 1',

Description This query returns the logs for the identifier 'MIPI D-PHY Editor 1'.

Example Query

:PLUGin:DPHYplugin:RUN:LOG? 'MIPI D-PHY Editor 1'

Remote Queries to find Parameter ranges (maximum and minimum limits)

The D-PHY Editor plug-in consists of several parameters, where you may define a value only within a permissible range. The D-PHY Editor plug-in automatically configures the ranges for such parameters based on certain options you may select on the user interface or remotely.

On the D-PHY Editor plug-in GUI, you may simply hover the mouse cursor over a specific parameter's value field to find its maximum and minimum permissible values. However, when working remotely, you may have to check the permissible ranges before configuring the value of a parameter.

While running a query for a parameter returns its current value, you can add the **MAX** and **MIN** identifiers in your query (separated by a comma after the plug-in identifier) to find the maximum and minimum value, respectively, for that parameter.

Consider the following examples to understand how to use these identifiers:

1 Checking **De-Emphasis** parameter range under **Signal Levels**:

Running the **:PLUGin:DPHYplugin:LEVELs:DEEMphasis[:VALue]?** query returns the current value configured for the **De-Emphasis** parameter under **Signal Levels**.

-> :PLUGin:DPHYplugin:LEVELs:DEEMphasis:VALue? 'MIPI D-PHY Editor 1'

<- 0.00000000000000E+00

To find the maximum value allowed for the **De-Emphasis** parameter, add the **MAX** identifier to the query separated by a comma.

-> :PLUGin:DPHYplugin:LEVELs:DEEMphasis:VALue? 'MIPI D-PHY Editor 1', MAX

<- 0.00000000000000E+00

Similarly, to find the minimum value allowed for the **De-Emphasis** parameter, add the **MIN** identifier to the query separated by a comma.

-> :PLUGin:DPHYplugin:LEVELs:DEEMphasis:VALue? 'MIPI D-PHY Editor 1', MIN

<- -1.00000000000000E+01

When converted from the scientific notation to numeric values, the maximum and minimum limit for the **De-Emphasis** parameter are 0dB and -10dB.

You may now configure the value for the **De-Emphasis** parameter remotely using **:PLUGin:DPHYplugin:LEVELs:DEEMphasis[:VALue]** command within the permissible values.

2 Checking **LP Data Rate** parameter range under **Data Rates & Transition Times**:

Running the **:PLUGin:DPHYplugin:LPMode:DATarate?** query returns the current value configured for the **LP Data Rate** parameter under **Data Rates & Transition Times**.

-> :PLUGin:DPHYplugin:LPMode:DATarate? 'MIPI D-PHY Editor 1'

<- 1.00000000000000E+07

To find the maximum value allowed for the **LP Data Rate** parameter, add the **MAX** identifier to the query separated by a comma.

-> :PLUGin:DPHYplugin:LPMode:DATarate? 'MIPI D-PHY Editor 1', MAX

<- 1.50000000000000E+08

Similarly, to find the minimum value allowed for the **LP Data Rate** parameter, add the **MIN** identifier to the query separated by a comma.

-> :PLUGin:DPHYplugin:LPMode:DATarate? 'MIPI D-PHY Editor 1', MIN

<- 5.00000000000000E+06

When converted from the scientific notation to numeric values, the maximum and minimum limit for the **LP Data Rate** parameter are 5 Mb/s and 150 Mb/s.

You may now configure the value for the **LP Data Rate** parameter remotely using **:PLUGin:DPHYplugin:LPMode:DATarate** command within the permissible values.

Therefore, to find out the permissible ranges for parameters with numeric values, add the **MAX** and **MIN** identifiers in your query (separated by a comma after the plug-in identifier) to find the maximum and minimum value, respectively, for that parameter.

Edition 6.3, July 2019

www.keysight.com

KEYSIGHT
TECHNOLOGIES