Keysight MIPI C-PHY Editor for M819xA AWG

User Guide

KEYSIGHT TECHNOLOGIES

Notices

Trademarks

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

Manual Part Number

M8085-91010

Edition

Edition 4.2, December 2015

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 SCPI Programming

Keysight MIPI C-PHY Editor for M819xA AWG

User Guide

1 Introduction

This chapter describes a high-speed serial interface called C-PHY, which provides high throughput performance over bandwidth limited channels for connecting to peripherals.

KEYSIGHT TECHNOLOGIES

Overview

The M8070A software has an add-on **MIPI C-PHY Editor** that generates C-PHY signals so as to test the DUTs that are compatible to this new standard.

The C-PHY describes a high-speed, rate-efficient PHY where channel rate limitations are a factor. The needs of rate limited channels are accomplished through the use of 3-Phase symbol encoding technology delivering approximately 2.28 bits per symbol over a three-wire group of conductors. This C-PHY specification (http://mipi.org/) has been written primarily for the connection of cameras and displays to a host processor. Nevertheless, it can be applied to many other applications.

Key characteristics of C-PHY (High-Speed Mode) are:

- Uses a group of three conductors rather than conventional pairs. The group of three wires is called a lane, and the individual lines of the lane are called: A, B and C. C-PHY does not have a separate clock lane.
- Within a three-wire lane, two of the three wires are driven to opposite levels. All wires are terminated at the end to the same star termination point. At one single time one line needs to have high, one line mid and the third line low level. No two lines may have the same level at one point in time. The voltages at which the wires are driven changes at every symbol.
- Multiple bits are encoded into each symbol epoch, the data rate is ~2.28x the symbol rate. There is a transition coding.
- Clock timing is encoded into each symbol. This is accomplished by requiring that the combination of voltages driven onto the wires must change at every symbol boundary on at least one wire. This simplifies clock recovery.

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

- System software license ("M8070A-0TP" or "M8070A-0NP")
- C-PHY Editor license ("M8085A-CT1 | M8085A-CN1 | M8085A-CD1")
- M8190A software version 5.0 or later for both AWGs and with the following options:
  - 002
  - 12G
  - AMP
  - SEQ
  - FSW

For more details on how to install these licenses, refer to *M8020A User Guide*.

NOTE **Currently the MIPI C-PHY Editor does not support very large patterns. However, if the user requires to test large patterns, the AWG large memory option (0G2) might be needed.**

Overview of C-PHY Functionality

C-PHY provides a synchronous connection between master and slave. A practical C-PHY configuration consists of one or more three-wire lanes. The link includes a high-speed signaling mode for fast-data traffic and a low-power signaling mode for control purposes. Optionally, a low-power escape mode can be used for low speed asynchronous data communication. High-speed data communication appears in bursts with an arbitrary number of payload data bytes.

The C-PHY uses three wires per lane, so three wires are required for the minimum C-PHY configuration. In high-speed mode each lane is typically terminated into a star point at mid level in case of the same high low levels on all three lines with a symmetric offset to the mid level driven by a low-swing, 3-Phase signal. In low-power mode all wires are operated single-ended and non-terminated. To minimize EMI, the drivers for this mode shall be slew-rate controlled and current limited.

The maximum achievable bit rate in high-speed mode is determined by the performance of transmitter, receiver and interconnect implementations. This specification is primarily intended to define a solution for a symbol rate range of 80 to 2500 Msps per lane, which is the equivalent of about 182.8 to 5714 Mbps per lane. Although C-PHY configurations are not limited to this range, practical constraints make it the most suitable range for the intended applications. For a fixed clock frequency, the available data capacity of a C-PHY configuration can be increased by using more lanes. Effective data throughput can be reduced by employing burst mode communication. The maximum data rate in low-power mode is 10 Mbps.

Overview of Lane Signaling States

A C-PHY configuration consists of one or more lanes. All lanes supports high-speed transmission and escape mode in the forward direction. The current flow through the lane for all six wire states.The positive-polarity wire states on the left and negative-polarity wire states on the right. The three rotation states (x, y and z) are shown from top to bottom. The six

driven states (called wire states) on a C-PHY lane are called: +x, -x, +y, -y, +z, and -z. The positive polarity wire states have the same wires driven as the corresponding negative polarity states, but the polarity is opposite on the driven pair of wires. For example: the +x wire state is defined as A being driven high and B driven low, while the -x wire state is B driven high and A driven low. The “undriven” conductor can be undriven when operating at lower symbol rates, or is actually driven by a termination at a voltage half way between the highest and lowest driven levels if operating at higher symbol rates.

Representation of Symbols in High-Speed Mode

One of the symbol to wire state encoding rules is that a state-transition exists at every symbol boundary. The reason for this rule is that it encodes the clock timing within the symbol, which has a number of advantages.

With six possible wire states there are always 5 possible transitions to the next wire state from any present wire state. The possible state transitions are illustrated in the state diagram in Figure 1 on page -9. The symbol value is defined by the change in wire state values from one unit interval to the next. Note that more than two bits of information (actually log2(5) = 2.3219 bits) can be encoded into each symbol. Seven consecutive symbols are used to transmit 16 bits of information. (Note that 57 = 78,125 permutations in seven consecutive symbols, with five possible wire state transitions that define each symbol. The information encoded in seven symbols is more than sufficient to represent a 16-bit binary value, 216 = 65,536.)

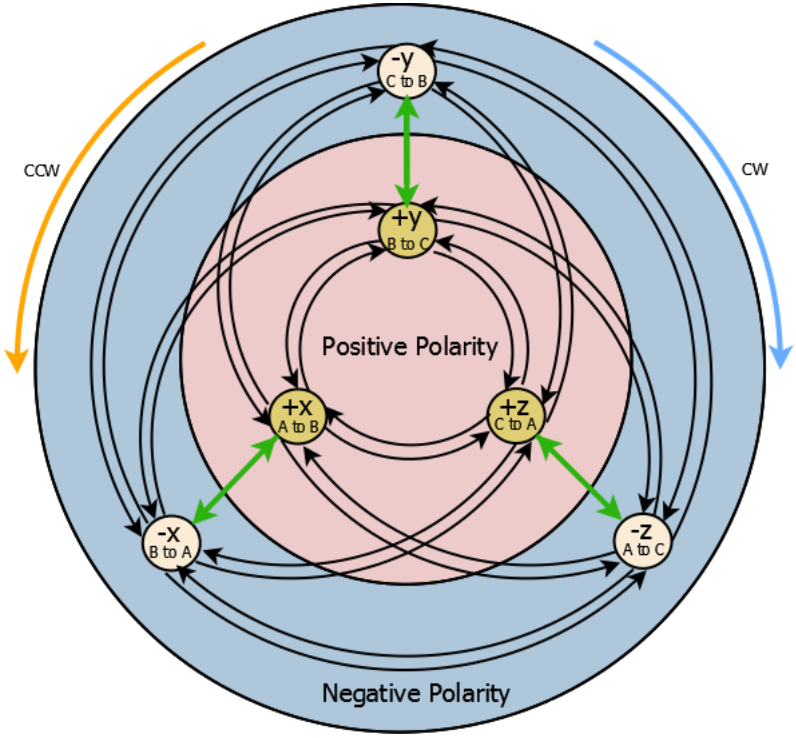


Figure 1 State Diagram showing all six wire states, and all possible transitions

High-Speed Data Transmission Burst

The sequence of events during the transmission of a Data Burst is shown in Figure 2 on page -11. For any lane, transmission can be started and ended independently by the protocol layer. However, for most applications the lanes will start synchronously but may end at different times due to an unequal amount of transmitted bytes per lane.

Beginning from the TS-HS-Exit state, LP-111, the signals transition to LP-001 and then LP-000 to signal that high-speed data transmission will begin soon. At the end of TX-HS-Prepare Duration, the low-power drivers are disabled and the high-speed drivers are enabled simultaneously. The first high-speed wire state transmitted at the beginning of TX-HS-Preample Pattern shall be the “+x” state. This does not correspond to any particular symbol value because there is no previous HS wire state

before it. It is recommended that the receive circuitry be initialized so the non-existent prior state value corresponds to the “-z” state so the decoded symbol resulting from this first wire state is “3” (Flip = 0, Rotation = CW, Polarity = opposite). Although this first wire state shall be “+x”, it is likely that the first few wire states of the TX-HS-Preample Pattern interval will not be seen at the high-speed receiver. This is because there will be some delay for the high-speed drivers to reach their required signal levels at the beginning of TX-HS-Preample Pattern, and also the high-speed receivers will be enabled and at some point start producing outputs toward the end of TX-HS-Preample. The receive circuitry shall be enabled toward the end of TX-HS-Preample when it is safe for it to reliably decode the “3” symbols during TX-HS-Preample. It is not guaranteed at exactly which symbol clock generation and symbol decoding will begin at the end of TX-HS-Preample. The TX-HS-Preample field may often consist of multiple groups of seven “3” symbols to provide a sufficient number of clocks to the upper layer protocol to initialize any pipeline stages prior to receiving data. The length of TX-HS-Preample is a programmable value set in the master.

The master may output a programmable sequence during TX-HS-Prog-Seq Pattern of the preamble, if it is enabled using a programmable sequence enable bit such as the MSB of the control register. The symbol values transmitted in the programmable sequence, or whether the programmable sequence is used at all, is a choice of the system designer.

Figure 2 on page -11 shows example of the preamble with the programmable sequence. Seven symbols of value “3” are sent during TX-HS-Pre-End Pattern just prior to sending the Sync Word.

The Sync Word precisely identifies the beginning of the Packet Data and also identifies the timing alignment of word boundaries in the Packet Data. The Sync Word contains a sequence of five “4” symbols which does not occur in any sequence of symbols generated by the Mapper. The Sync Word may also be transmitted later in the burst to mark the beginning of redundant Packet Headers transmitted by the upper layer protocol.

The end of Packet Data is identified by a unique sequence of “4” symbols in TX-HS-Post Pattern. The receiver identifies the end of Packet Data when it detects a sequence of seven consecutive “4” symbols. The Post field may often consist of multiple groups of seven “4” symbols to provide a sufficient number of clocks to the upper layer protocol to clear out any pipeline stages that may contain received data.

The high-speed drivers are disabled and the low-power drivers are enabled simultaneously, at the end of TX-HS-Post Pattern. All three signals of the lane are driven high together to LP-111, the TX-HS-Exit state.

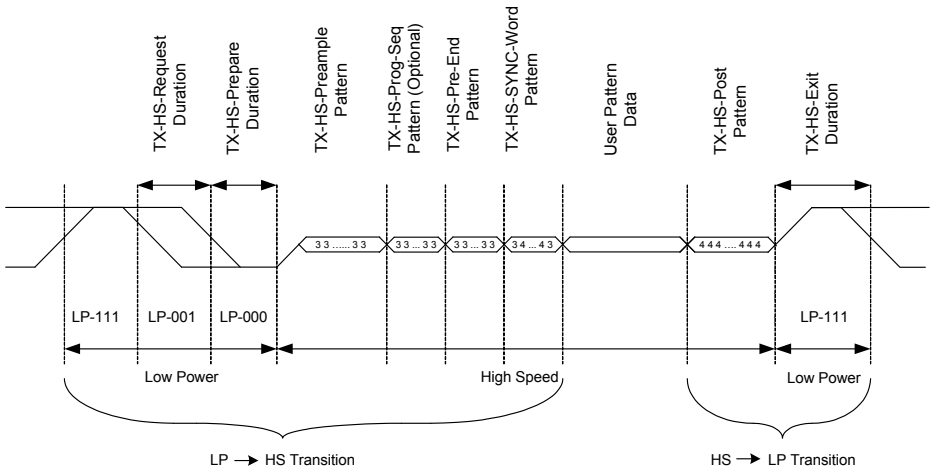


Figure 2 High-Speed Data Transmission in Burst

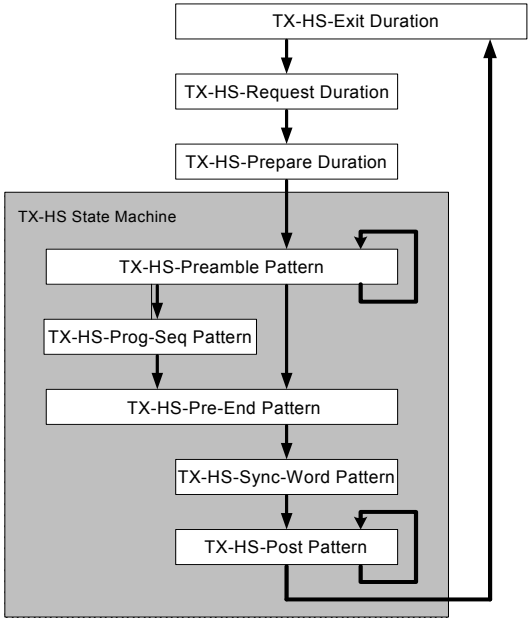


Figure 3 TX State for High-Speed Data Transmission

The following table describes the high-speed data transmission.

Table 1 High-Speed Data Transmission Description

State	Line State	Exit State	Exit Condition
TX-HS-Exit Duration	Transmit LP-111	TX-HS-Request Duration	On request of Protocol for High-Speed Transmission
TX-HS-Request Duration	Transmit LP-001	TX-HS-Prepare Duration	End of timed interval t_{LPX}
TX-HS-Prepare Duration	Transmit LP-000	TX-HS-Preamble Pattern	End of interval $t_{3\text{-}PREPARE}$
TX-HS-Preamble Pattern	Preamble 3,3,3,3...	TX-HS-Prog-Seq Pattern	End of Preamble & Prog-Seq selected
		TX-HS-Pre-End Pattern	End of Preamble & Prog-Seq not selected
		TX-HS-Preamble Pattern	Preamble words remaining count > 0
TX-HS-Prog-Seq Pattern	Prog-Seq	TX-HS-Pre-End Pattern	End of Prog-Seq
TX-HS-Pre-End Pattern	Pre-End	TX-HS-Sync-Word Pattern	End of Pre-End
TX-HS-Sync-Word Pattern	Sync Word	TX-HS-Post Pattern	End of Sync-Word
TX-HS-Post Pattern	Post 4,4,4,4...	TX-HS-Exit Duration	Last word of Post sent

16-Bit to-7-Symbol Mapping

The Mapper is the outer-most function in the C-PHY digital coding system that occurs on the transmit side. It converts a 16-bit word into a group of seven symbols at the transmitting end. The
16-bit to-7-symbol Mapper perform a mapping function between 16-bit input/output values and a group of 7 symbols which is comprised of seven 3-bit symbol values. Each symbol is comprised of a flip, rotate and polarity bit, so for any particular symbol, n, sn = [Flip[n], Rotation[n], Polarity[n]]. A seven-symbol Mapper output value is defined for every possible 16-bit Mapper input value. Since the mapping is completely feed-forward function, pipeline registers can be inserted between intermediate stages if necessary to lessen timing constraints in systems that operate at a high symbol rate.

The following table shows 16-bit to 7-symbol mapper:

Table 2 16-Bit to 7-Symbol Mapper

Pattern File Entry	Description
% (Percentage)	Starts or stops binary to symbol mapping. Number of enclosed bits must a multiple of 16. Once the first % is found the subsequent symbols may only contain the binary states 0 and 1 interpreted MSB (Most Significant Bit) first. The % may only occur in pairs. It can be placed freely inside the pattern definition and its output will always be regarded as high speed mode symbols.
0	Binary 0 (Low)
1	Binary 1 (High)

Pattern File Symbol Legends

The following table shows the line states available in **High-Speed Mode**:

Table 3 Line States

Pattern File Entry	Line State	Line State {Line A, Line B, Line C}
X	+x	{1, 0, ½ }
x	-x	{0, 1, ½ }
Y	+y	{½ , 1, 0}
y	-y	{½, 0, 1}
Z	+z	{0, ½, 1}
z	-z	{1, ½, 0}

The following table shows the symbols (transitions) available in **High-Speed Mode**:

Table 4 Symbol (Transitions

Pattern File Entry	Symbol Input Value	Activity
0	000	Rotate CCW, polarity stays same
1	001	Rotate CCW, polarity is inverted
2	010	Rotate CW, polarity stays same
3	011	Rotate CW, polarity is inverted
4	1xx	Same phase, polarity is inverted

The following table shows the line states provided by **Low Power Mode**:

Table 5 Lines states provided by Low Power Mode

Pattern File Entry	Line A	Line B	Line C	Activity
L	0	0	0	One “L” sets all three wires to low state
C	0	0	1	One pattern file entry defines the state of all 3 wires (1 Symbol). An upper case letter means only the selected wire is high, the 2 other wires are low. A lower case letter means only the selected wire is low, the 2 other wires are high.
B	0	1	0	
a	0	1	1	
A	1	0	0	
b	1	0	1	
c	1	1	0	
H	1	1	1	One “H” sets all three wires to high state

Pattern Coding Examples

The following table shows the example of different types of patten coding used:

Table 6 Pattern Coding Examples

Pattern Coding	Line States	Description
X0123 (XZyzX)	210432113 Resulting Line States: X -> YxzZxyXzX	High speed init pattern and high speed loop pattern. The high speed signal can be coded with transition symbols too.
X0123 (XZyzX)	%0101000011101101% Resulting Transitions: X -> 1432300 Resulting Line States: zZxyZYX	High speed init pattern and coded binary loop pattern. Between the percentage symbols (%) binary coding can be used. The last wire state of the init pattern needs to be equal to the last wire state of the loop pattern (X). If this is not the case coding of the looped pattern will be erroneus.
xXyx	X0241%1011011101001010%x Resulting Line States: X ZXxZ yYzxYZY x	High speed init pattern and high speed loop pattern mixed with binary coding. You can mix the use of wire states, transition symbols and binary coding. This is mainly useful for introducing coding errors.

Pattern Coding	Line States	Description
HLB	xYzxZyyZy	Low power init pattern and high speed loop pattern. There will be a low power to high speed transition at the end of the init pattern.
HLBZ	xYzxZyyZ	Mixed low power/high speed init pattern and high speed loop pattern. The transition from low power to high speed will happen within the init pattern.
acL	y -> 0120412CLAbc	Low power init pattern and high speed and low power loop pattern. There will be a high speed low power transition within the loop pattern and a low power high speed transition at the end of the loop pattern. Limitation: The "LP->HS Start Wire State" cannot be modified.

	Expanded Pattern	Wire States		Expanded Pattern	Wire States
Init Pattern	acL	acL	Loop Pattern 1	0120412	xZXZzYZ
LP->HS Start	H	H	HS->LP Post	4444444	zZzZzZz
LP->HS Request	C	C	HS->LP Exit	C	C
LP->HS Prepare	L	L	Loop Pattern 2	CLAbc	CLAbc
LP->HS Start Wire State	X	X	LP->HS Start	H	H
LP->HS Preamble	3333333	yZxYzXy	LP->HS Request	C	C
LP->HS Pre-End	3333333	ZxYzXyZ	LP->HS Prepare	L	L
LP->HS Sync	3444443	xXxXxXy	LP->HS Start Wire State	X	X
			LP->HS Preamble	3333333	yZxYzXy
			LP->HS Pre-End	3333333	ZxYzXyZ
			LP->HS Sync	3444443	xXxXxXy

Transmit Lane PRBS Register Operation

The PRBS generator generates bits. These bits are then put into the 16-Bit to 7-Symbol Mapper resulting in transitions which will be played back indefinitely after having been converted to wire states.

The Transmit Lane PRBS Register Q[16:1] is the source of data input to TxD[15:0] of the Mapper when the lane master is transmitting one of the three PRBS patterns as defined by the TLRn_Test_Patterns_Select register. The Transmit Lane PRBS register is initialized using the seed values TLRn_PRBS_Seed_0, TLRn_PRBS_Seed_1 and TLRn_PRBS_Seed_2. The first word transmitted from the PRBS generator is equal to the seed value: [TLRn_PRBS_Seed_1[7:0], TLRn_PRBS_Seed_0[7:0]]. This initial 16-bit value from the PRBS register is transmitted immediately following transmission of the first Sync Word after the low-power to high-speed mode transition. The Transmit Lane PRBS Register is shifted 16 bit positions after each 16-bit word is output to minimize correlation from one data value to the next. This way no bits are re-used in successive samples.

The important part for PRBS generation is that for degree 9 and 11 the seed value register is 16 bits wide. The seed defines the whole 16 bits and the first value shifted out of the PRBS is the value at bit 16. This means that the first 7 bits for degree 9 will only occur once and will not be looped. For PRBS 11 this will be true for the first 5 bits. PRBS for degree 18 is different. The Seed register is 18 bits wide and all bits will be looped. But Bit position 16 is still where the values from the PRBS will be captured.

Example Seed values and data sequences for the chosen PRBS mode are as follows:

- PRBS9 – Seed = 0x789a; TLRn_PRBS_Seed_0[7:0] = 0x9a; TLRn_PRBS_Seed_1[7:0] = 0x78; TLRn_PRBS_Seed_2[7:0] value does not matter; Transmit data sequence: 0x789a, 0x9980, 0xc651, 0xa5fd, 0x163a, 0xcb3c, 0x7dd0...
- PRBS11 – Seed = 0x789a; TLRn_PRBS_Seed_0[7:0] = 0x9a; TLRn_PRBS_Seed_1[7:0] = 0x78; TLRn_PRBS_Seed_2[7:0] value does not matter; Transmit data sequence: 0x789a, 0x5e64, 0xfee0, 0xac43, 0xa9a1, 0xe4ce, 0xfea0...
- PRBS18 – Seed = 0x2789a; TLRn_PRBS_Seed_0[7:0] = 0x9a; TLRn_PRBS_Seed_1[7:0] = 0x78; TLRn_PRBS_Seed_2[7:0] = 0x02; Transmit data sequence: 0x789a, 0x8d77, 0x0dbc, 0x74e1, 0x8108, 0x414a, 0x3915...

For details, visit http://mipi.org/.

eSpike

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.

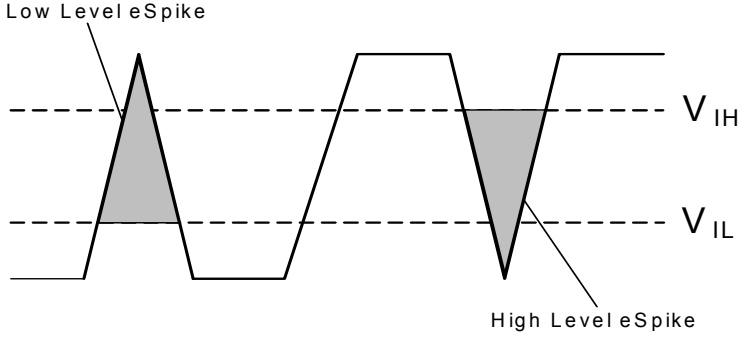


Figure 4 Input Glitch Rejection of Low-Power Receivers

The following table shows the LP Receiver DC specifications:

Table 7 LP Receiver DC Specifications.

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

The following table shows the LP Receiver AC specifications:

Table 8 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

ISI Generation - S6P Support

Scattering Parameter are used to describe the electrical characteristics of a linear system.

So when a signal is transmitted at some given frequency through a port (in circuit), some part of it is reflected and rest is transmitted. Transmitted signal reaches the receiver at different time, with different magnitudes and phase and mingles with existing symbols, which means part or all of a given signals will be spread into smaller signals, thereby interfering with the correct detection of the signal, which is one of the causes of ISI.

In order to define characteristics of such system Scattering matrix or S-Parameter is used in the form of SnP files. Where n is the number of ports. 2 ports (S2P) file contains four pairs of magnitude and amplitude values at given frequency.

By applying these set of magnitude and phase values at given frequency, ISI is emulated on respective lines.

In addition to S2P support, now S6P files are also supported to emulate ISI on all 3 lines.

S6P file are transformed to 2 port parameters (magnitude and phase) which are applied to line A, B and C.

Start Pattern and Triggered Start

The purpose of the triggered startup is to enable the C-PHY plugin to provide a static LP STOP signal.

C-PHY is represented as an LP-111 state on the data lane(s). Triggered start mode can be also selected as LP-000 state from Trigger Start property.

This triggered startup is only active when setting the parameter "AWG Setup. Startup Mode" to "Triggered".

The parameter can only be changed while the plugin is stopped. When this feature is selected and signal generation is started the plugin will calculate the complete waveform and downloads it into the AWG as it would occur for the "Immediate" startup mode. The difference is then that once calculation and download is finished the AWG outputs will drive a static voltage at the LP level with is selected for the respective wire/lane.

To advance from this idle state either the GUI trigger button can be pressed, or the respective SCPI command can be called, or the TRIG IN of the AWG module can be called. After doing this the run mode is active and parameters can be reprogrammed on the fly.

Separate LP Amplitudes per Line

In low power mode same levels settings are applied on all lines by AWG. So it's not possible to configure levels on individual line. In order to configure levels per line, separate low power levels field per line is provided. This has a direct implication on eSpike calculation and thus eSpike relative to line considers appropriate levels into account for calculations.

De-skew without Re-cabling

Default Calibration is done on normal ports of AWG and via same ports DUT is connected with same cable as used during calibration. Since calibration is sample rate dependent, every time sample rate is changed, DUT needs to be disconnected and re-calibration is performed again.

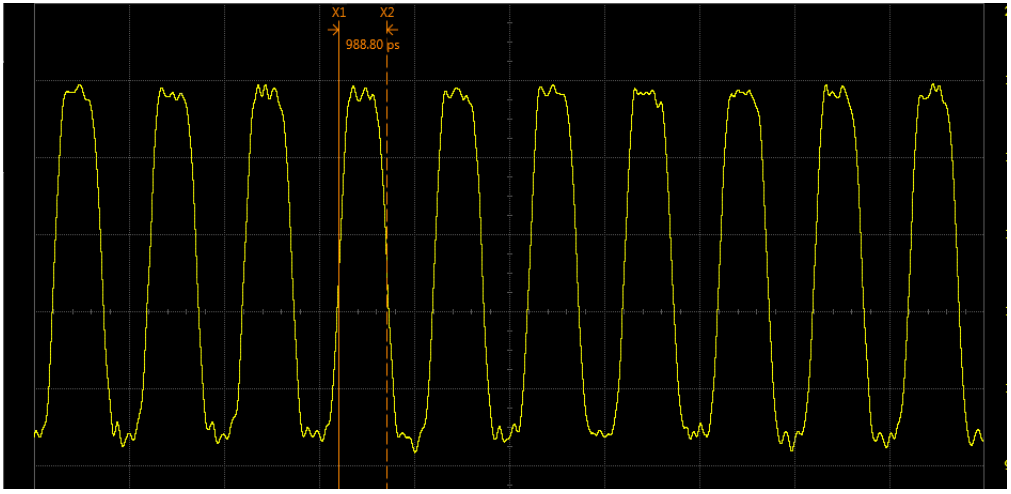
To avoid this situation, now both normal and complement ports of AWG are used. So that calibration can be done on complement port, keeping the normal outputs connected to DUT. Any change in sample rate will not require any rewiring of DUT.

Only prerequisite is to have match cable pair length for normal and complement connection. If this is not the case then signals are not de-skewed correctly.

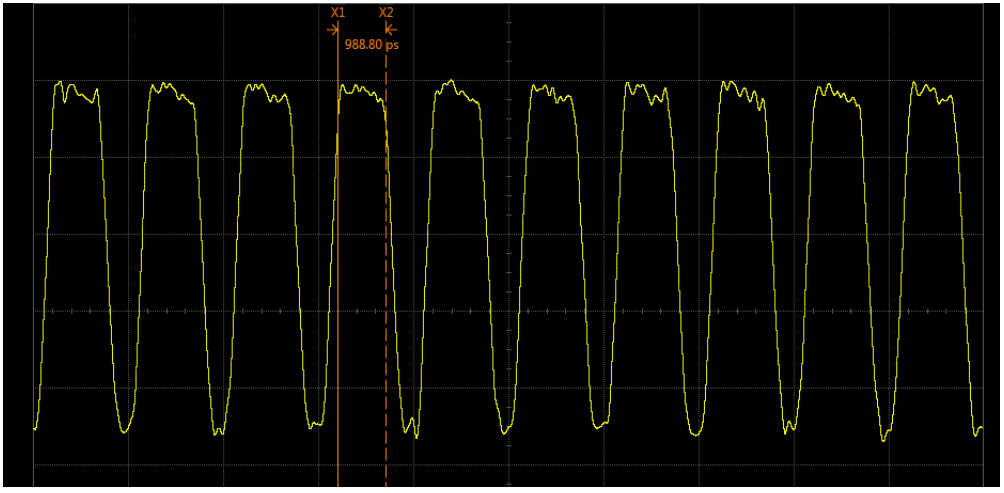
Duty Cycle Distortion

Duty cycle distortion is a timing impairment that falls under deterministic jitter category. It is used to achieve eye closure by varying the transition time of rising and falling edge. The time domain behavior of DCD is that all rising edges are delayed or advanced by the same amount. The same behavior holds good for falling edges, which are delayed or advanced by the same amount. There are two scenarios, one in which rising edge is advanced and falling edge is delayed and second in which rising edge is delayed and falling edge is advanced. By delaying or advancing transition time of edges in this manner, duration of levels are either increased or decreased, which results in eye closure. Both edges cannot be advanced or delayed at the same time, otherwise DCD will not have desired behavior.

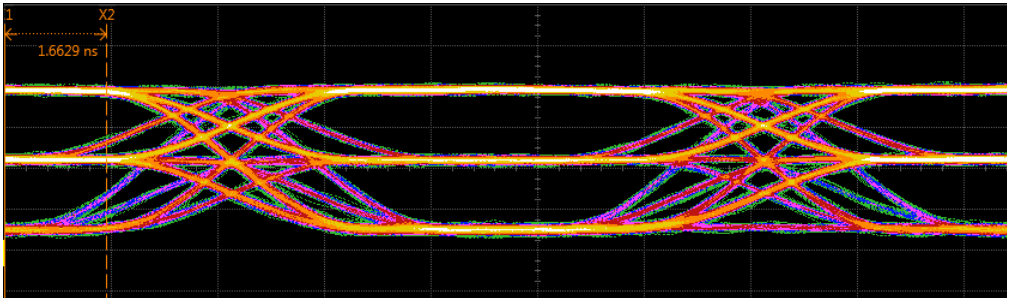
The following figure shows an example when no DCD is applied on high speed symbols:



The following figure shows an example when DCD of 300 mUI applied on high speed symbols:

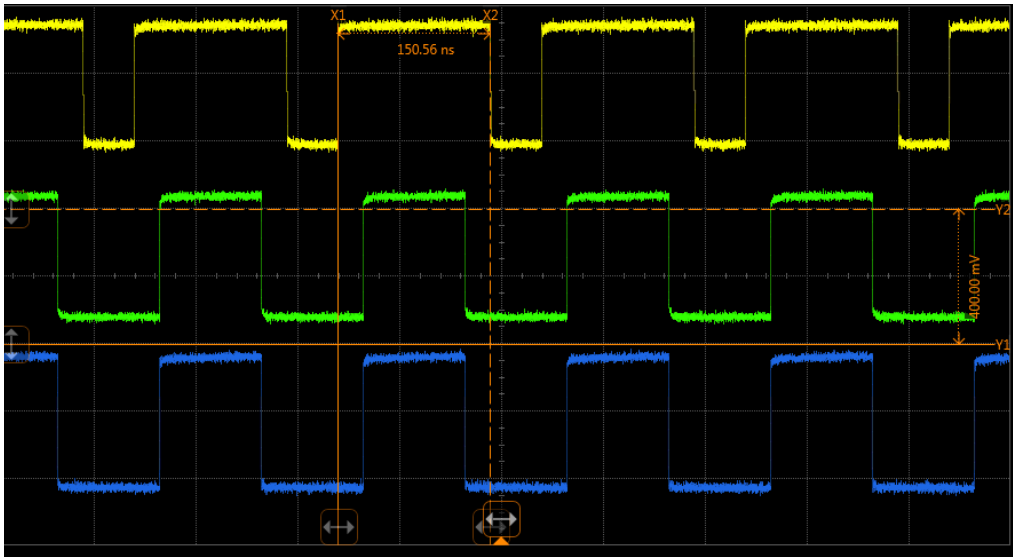


The following figure shows an example of eye when DCD of 300 mUI applied on high speed symbols:



Same holds good for LP Pulse width, except it works for Low Power symbols only. But in case of LP pulse width instead of saying by how much amount edges are shifted or delayed, we can control the width of pulse with respect to high level.

The following figure shows an example of LP pulse of 150 ns width on line A.



Different values of DCD and Pulse width applied to channels, results in skew among channels due to shift in transition times. At the boundary of LP and HS symbols DCD or pulse width doesn't apply. As both DCD and pulse width requires two consecutive high speed and low power rising and falling edges.

Skew Calibration Values

To access skew calibration values, read only fields in GUI under calibration functional block is provided. This read-only value is the sum of measured delay (skew) between AWG channels and delay at which marker bit is set in delay segment to trigger second AWG. Skew calibrated values are provided per channel and can be accessed via SCPI too.

These properties will indicate how much skew between the 3 different AWG channels was measured during calibration. It shifts the pattern in way that all three outputs will be synchronous at the receiver.

A pre-requisite to use calibration on the complement and normal pins is to have matched cable pairs, which connects scope and DUT.

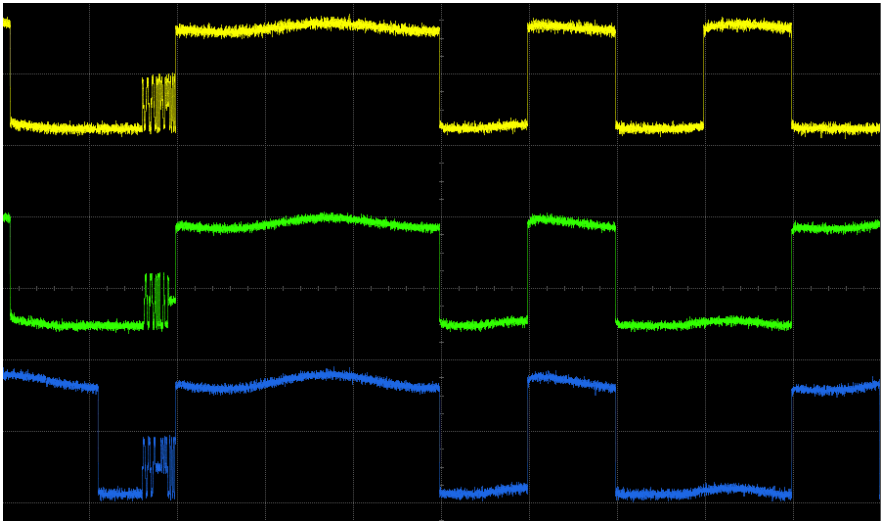
Hysteresis

The Hysteresis is incorporated into Low Power receivers to reduce sensitivity to noise, and prevents Logic state changes due to short-term, low amplitude excursions below the VIH Logic-1 threshold (or above VIL Logic-0 threshold), after the instantaneous voltage has initially crossed the threshold for any given bit interval.

In order to test, if DUT can successfully receive Low Power test sequence after applying a sin wave signal of desired peak voltage (Hysteresis voltage which is half of peak to peak amplitude) on signal, which results in additive noise signal on line A, B or C. Period of applied sin wave signal is 2.3x the TLPX value of the applied test sequence. Applied signal will modulate the phase of the additive noise with respect to the LP data test sequence, for added impact. The amplitude of the additive noise should be calibrated to produce an approximately 25 mVpk (50 mVpp) deviation from the nominal LP-0/1 levels of the Test System (which will be set to the measured VIH and VIL values for the DUT). When the additive noise is enabled, the DUT should still be able to successfully receive the LP test sequence without error, if it employs sufficient hysteresis on its LP-RX.

Sinusoidal Noise is applicable to Low Power symbols only. High Speed symbols are unaffected by this noise.

The following figure shows hysteresis of 15 mV, 20 mV and 25 mV induced on line A, line B and line C, respectively.



2 Basic Setup

Hardware Setup

The hardware setup requires:

- Two AWGs modules (M8190A)

  Please make sure to use the M8190A software version 5.0 or later for both AWGs and with the following options:

  - 002
  - 12G
  - AMP
  - SEQ
  - FSW
- An Embedded Controller (M9536A)
- A 5 slot AXIe chassis
- An Inifniium or SCPI compatible oscilloscope
- LAN or GPIB/USB adapter

For hardware setup, follow the given steps:

1 Put two AWGs (M8190A) and an Embedded Controller in the 5 slot frame AXIe chassis as shown in the Figure 5 on page 26.

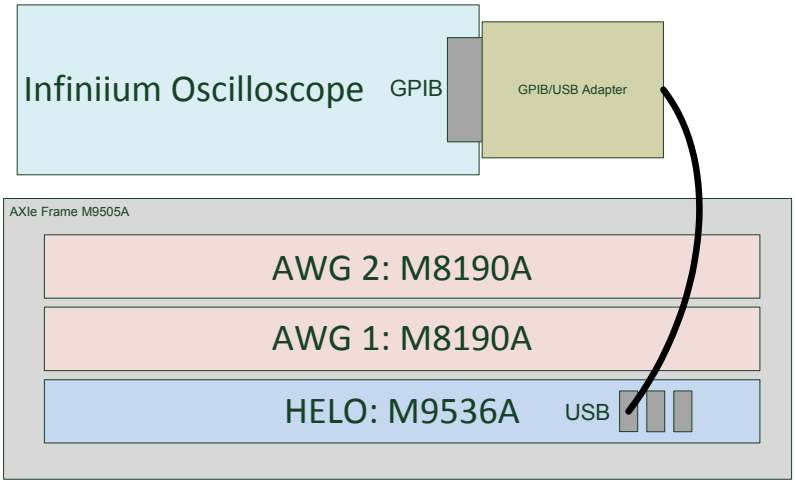


Figure 5 Hardware setup

2 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 AWG 1 must be installed in the slot 2 and the AWG 2 must be installed in the slot 4 of the 5 slot AXIe chassis.

4 Connect an Inifniium or SCPI compatible Oscilloscope for deskewing the third data channel. This scope can be either connected via LAN or via GPIB/USB adapter with the Embedded Controller host PC. For stability it is advised to use the GPIB/USB adapter instead of a LAN connection.

Connections

Once the hardware setup is in place, connect the AWG 1, 2 and the oscilloscope as shown in Figure 6 on page 28.

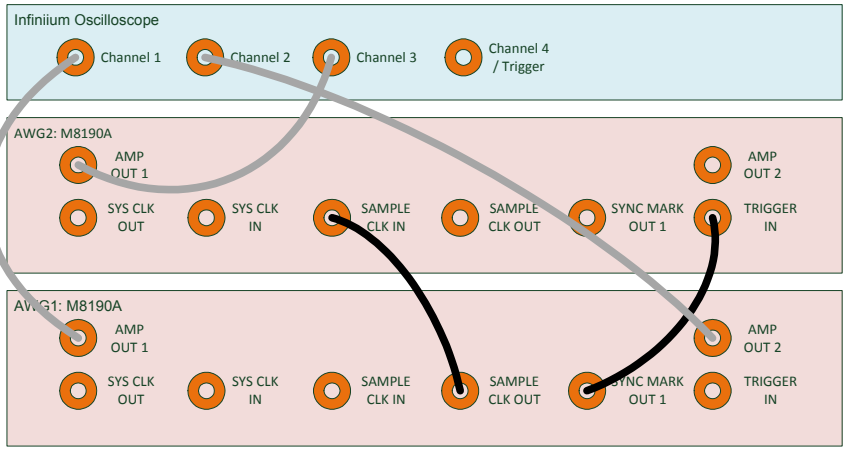


Figure 6 Connections

Refer to Table 9 on page 28 to know how to establish connections among the AWG 1, 2 and the oscilloscope:

Table 9 Connections among AWG 1, 2 and the oscilloscope

Instrument / Port	Connected to Instrument / Port
AWG1 / AMP OUT 1	Oscilloscope / Channel 1
AWG1 / Sample Clock Out	AWG 2 / Sample Clock In
AWG1 / Sync Mark Out 1	AWG2 / Trigger In
AWG1 / Amp Out 2	Oscilloscope / Channel 2
AWG2 / AMP OUT 1	Oscilloscope / Channel 3

Make sure to use cables of similar length for all three channels otherwise an unwanted skew will be present when using a sync module instead of deskew via oscilloscope. To get the best signal performance and to reduce AWG sampling artifacts, Transition Time Converters must be used on each data channel. (e.g. 4 Ghz Low Pass Filter).

For low AWG sample rates the AWGs internal delay capabilities may not suffice to deskew module 1 and 2 successfully. If this is the case, it is necessary to add external delay (e.g. via delay lines or cabling of different length) to module 1 data channels to reduce the delay needed for synchronizing both modules.

Software Installation

The MIPI C-PHY Editor plug-in has to be installed separately along with the M8070A system software. Please make sure that the system should already have M8070A software install on it.

NOTE **The M8070A software (version 2.5.x.x) is required to install the MIPI C-PHY Editor plug-in.**

Plug-in file which can be downloaded from Keysight webpage: www.keysight.com/find/m8000

The **MIPI C-PHY Editor** plug-in is a licensed feature. To enable it, the "M8085A-CT1 | M8085A-CN1 | M8085A-CD1 licenses are required.

The M8070A system software comes with a **Plug-in Manager** to simplify all the tasks related to plug-in management. For details on how to use the **Plug-in Manager** to install, uninstall and update the **MIPI C-PHY Editor** plug-in, refer to *Getting Started with M8070A Plug-ins*.

For M8070A plug-ins related documents, click **Start** > **All Programs** > **Agilent** > **M8070A** > **Plug-ins**.

You can also visit www.keysight.com/find/m8000 to find the latest version of related manuals.

Startup

1 Start two instances of the AWGs (M8190A firmware). The following figure shows one instance of the AWG.

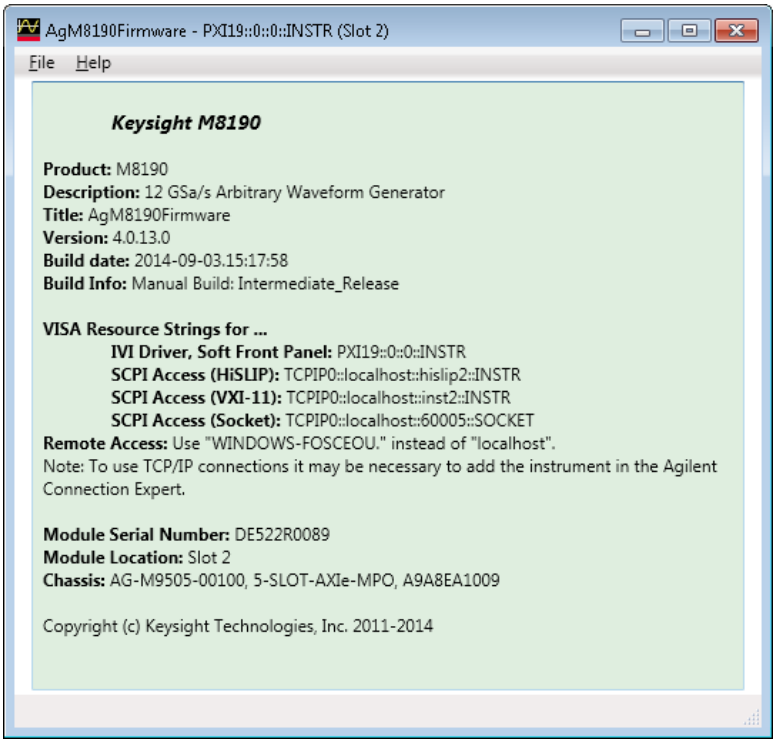


Figure 7 M8090A Firmware

2 Configure both AWGs and the USB connected Infiniium oscilloscope into Keysight IO Libraries. For easier MIPI C-PHY Editor setup it is advantageous to define an alias for the three devices. The AWG with slot 2 should be called “awg1” and the AWG with slot 4 “awg2”. The oscilloscope could be called “GPIBOscilloscope”.

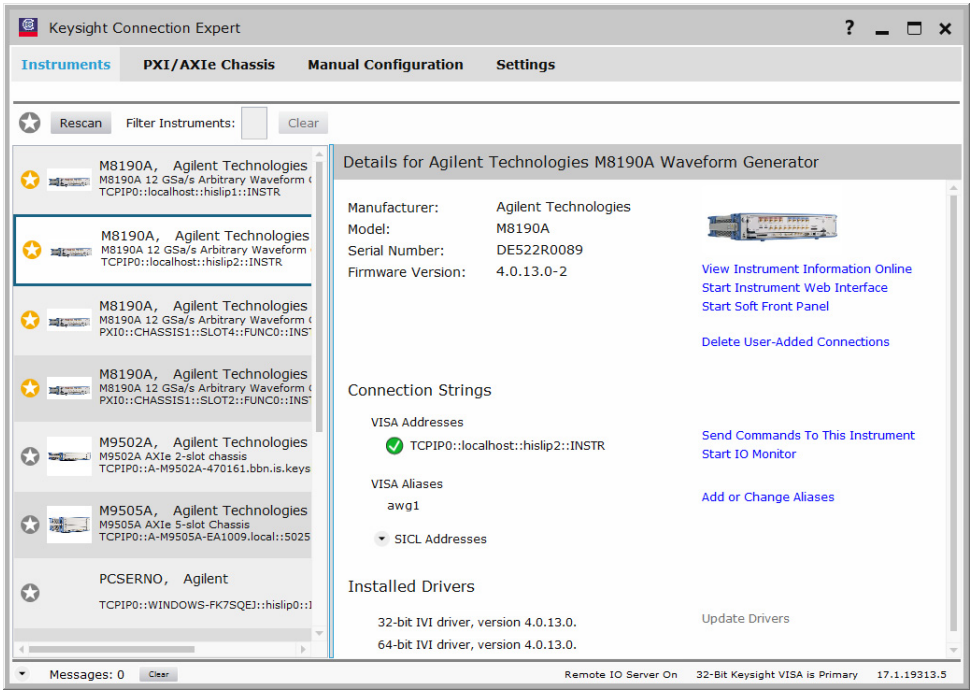


Figure 8 Configuring instrument using Keysight IO Libraries

3 Afterwards the setup of Keysight IO Libraries should look similar to this.

NOTE

Do not use the PXI0 addresses of the AWGs. These these are just for internal usage by the AWG firmware.

4 Start the M8070A GUI.

5 Go to **Menu Bar** > **Application** and the select **MIPI C-PHY Editor**. The **MIPI C-PHY Editor** interface will appear as shown in the Figure 9 on page 33.

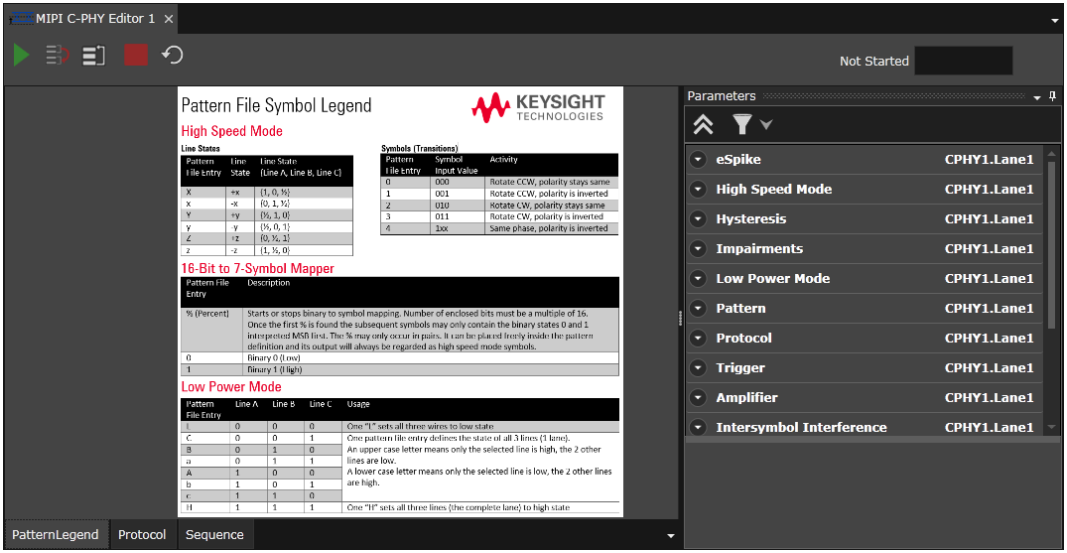


Figure 9 MIPI C-PHY Editor interface

6 However, if you do not see the **MIPI C-PHY Editor** option then you need to install it using **Plugin Manager** and also ensure that you have a valid license. For details, see Software Installation on page 30.

MIPI C-PHY Editor User Interface

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

- **Toolbar**: The toolbar provides the following convenient options:

Table 10 Toolbar

Icons	Description
	Press this button to parse the init and loop pattern/PRBS and start signal generation in the AWG once the pattern have been loaded into the AWG memory. As long as the measurement is running, it is only possible to change certain parameters like amplitude, eSpike and interference properties. Other parameters such as sample rate, symbol rate, protocol parameters and SCPI configuration cannot be changed while data generation is running. If it is desired to change these parameters, the data generation has to be stopped. The following list describes the parameters that are changeable while the data generation is running: ▪ eSpike - All properties ▪ High Speed Mode - All properties except Symbol Rate ▪ Hysteresis - All properties ▪ High Speed Duty Cycle Distortion - All properties ▪ Low Power Pulse - All properties ▪ Impairments - All properties ▪ Low Power Mode - All properties except Symbol Rate ▪ Pattern - Not changeable ▪ Protocol - Not changeable ▪ Trigger - All properties ▪ Amplifier - Not changeable ▪ Intersymbol Interference - All properties, depending on dissimilarity of old and new S-Parameter file there will be a voltage glitch present when applying new file. ▪ AWG Setup - Not changeable ▪ AWG1 Setup - Not changeable ▪ AWG2 Setup - Not changeable ▪ Calibration - Not changeable ▪ Oscilloscope Setup - Not changeable
	Press this trigger button to break from LP Stop states (LP000 and LP 111) into waveform playback. It is only valid for Startup Modes other than immediate. It starts waveform generation after AWGs have been programmed and initialization voltage level at LP High is active.

Icons	Description
	Press this button to restart waveform generation without pattern recalculation. The advantage of this function is instantaneous sequence restart e.g. helpful for doing Schmoo plots. The disadvantage is that as there is no pattern recalculation which is why the init pattern might not be compatible to the loop pattern. This will be the case if there were parameter changes during pattern playback which caused the waveform stored in AWG memory to change.
	Press this button to stop the waveform generation.
	Press this button to reset the parameters to the default values.

NOTE Please note that the reset icon only resets the parameters of MIPI C-PHY Editor to the default values and not the M819XA instrument.

- **Status Indicator**: The status indicator shows the current state of a waveform generation and downloading activity. It shows the following type of status:
  - **Running**: Indicates that the waveform generation and downloading activity is currently running.
  - **Stop**: Indicates that the waveform generation and downloading activity is stopped.
  - **Error**: Indicates that the waveform generation and downloading activity is suspended.

The following figure shows the status indicator while the waveform generation and downloading activity is on progress:

Figure 10 Status Indicator

- **Main Window**: Shows the Pattern File Symbol Legend. For details, refer to Setting up MIPI C-PHY Editor Parameters on page 37.

- **Parameters Window**: Provides settings to configure MIPI C-PHY Editor. For details, refer to Setting up MIPI C-PHY Editor Parameters on page 37.

Context Sensitive Help

To get quick information on the meaning of a particular GUI item, select ? from the title bar of the GUI and click on the item in question.

Setting up MIPI C-PHY Editor Parameters

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

eSpike

The parameters provided by **eSpike** shown in the Figure 11 on page 37.

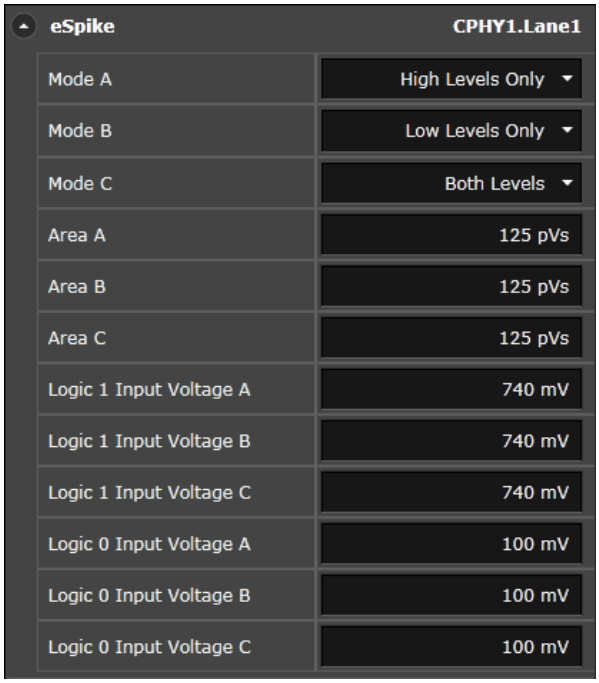


Figure 11 eSpike Parameters

The parameters provided for the **eSpike** are described in Table 11 on page 38.

Table 11 eSpike Parameters

Parameter	Description
Mode A/B/C	Selects whether eSpike insertion is turned on or off on line A, B or C, respectively. In the enabled state it is selectable whether the spikes should occur inside high levels only, low levels only or in high and low levels both. The eSpike will be inserted in the middle of the symbol with the defined area.
Area A/B/C	Defines the area with which the eSpike will be generated on line A, B or C, respectively. Take into consideration that the granularity of the area is limited by the number of samples available in the effective duration of the eSpike.
Logic 1 Input Voltage A/B/C	Specifies the lower receiver detection threshold for a logical 1 on line A, B or C, respectively, which is the voltage level at which the eSpike area inside this logical 1 is to be calculated from.
Logic 0 Input Voltage A/B/C	Specifies the upper receiver detection threshold for a logical 0 on line A, B or C, respectively, which is the voltage level at which the eSpike area inside this logical 0 is to be calculated from.

High-Speed Mode

The parameters provided by **High-Speed Mode** shown in the Figure 12 on page 39.

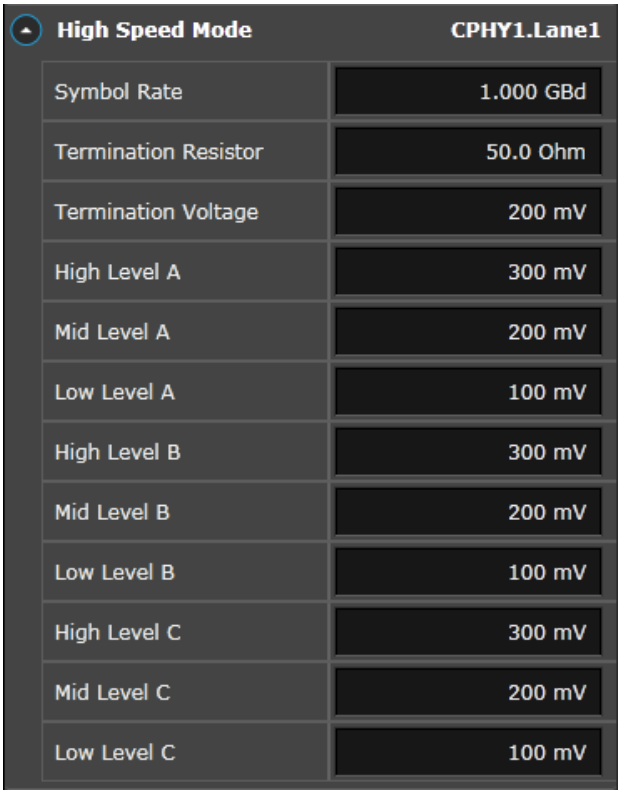


Figure 12 High-Speed Mode Parameters

The parameters provided by the **High-Speed Mode** are described in Table 12 on page 40.

Table 12 High-Speed Mode Parameters

Parameter	Description
Symbol Rate	Sets the symbol rate of C-PHY signal. This is the actual rate of circuit. The effective data rate of the transmission signal will be 2.28 higher than the selected value.
Termination Resistor	Sets resistance of the termination resistors. There is only one value for all three lines.
Termination Voltage	Sets the internal termination voltage of the DUT's receiver circuit. For the symmetrical case where the resistance of the receiver circuit line terminator are 50 Ohms each and the voltage of all three lines are symmetrical around the same mid level, this value needs to be set to the mid level. For the asymmetrical case, the actual termination voltage at the receiver needs to be chosen accordingly. If this is not done the voltages seen at the receiver input will be different from the selected value.
High Level A	Sets the high level of line A.
Mid Level A	Sets the mid level of line A.
Low Level A	Sets the low level of line A.
High Level B	Sets the high level of line B.
Mid Level B	Sets the mid level of line B.
Low Level B	Sets the low level of line B.
High Level C	Sets the high level of line C.
Mid Level C	Sets the mid level of line C.
Low Level C	Sets the low level of line C.

Hysteresis

The parameters provided by **Hysteresis** shown in the Figure 12 on page 39.

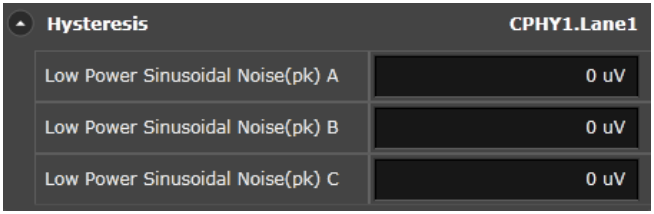


Figure 13 Hysteresis Parameters

The parameters provided by the **Hysteresis** are described in Table 12 on page 40.

Table 13 Hysteresis Parameters

Parameter	Description
Low Power Sinusodial Noise (pk) A/B/C	Sets hysteresis or additive noise in LP signal to check if DUT can still detect the valid LP sequence without any errors. The input hysteresis amplitude is deviation from the nominal LP 0/1 levels, which is half of its peak to peak value.

Impairments

The parameters provided by **Impairments** are shown in the Figure 14 on page 42.

Figure 14 Impairments Parameters

The parameters provided by the Impairment settings are described in Table 14 on page 43.

Table 14 Impairments Settings

Parameter	Description
High Speed Rise Time	Sets the rise time in the high speed mode. For integral dividers of sampling rate to symbol rate it is possible it is possible to set the transition times to 0 for getting a maximally clean signal. For non-integral dividers the transition times are needed to hide sampling artifacts. This is especially true when jitter is added to the signal. As the jitter will shift the phase of the signal in an arbitrary step size it is necessary for the jittered waveform to have transition time containing at least 2 samples. If the AWG restriction is not met the resulting output signal will contain quantized jitter in the granularity of a sample period.
High Speed Fall Time	Sets the fall time in high speed mode. For integral dividers of sampling rate to symbol rate it is possible to set the transition time to 0 for getting a maximally clean signal. For non-integral dividers the transition times are needed to hide sampling artifacts. This is especially true when jitter is added to the signal. As the jitter will shift the phase of the signal in an arbitrary step size it is necessary for the jittered waveform to have transition times containing at least 2 samples. If this AWG restriction is not met the resulting output signal will contain quantized jitter in the granularity of a sample period.
Sinusoidal Jitter Amplitude (p-p)	Sets the amount of sinusoidal jitter peak to peak applied to the lane.
Sinusoidal Jitter Frequency	Sets the frequency of sinusoidal jitter applied to the lane.
Bounded Uncorrelated Jitter (RMS)	Sets the bounded uncorrelated jitter (BUJ, Random Jitter Simulation) applied to the lane.
Analog Skew A/B/C	Sets the skew of line A/B/C via AWG delay line.
Enable Pulse Width	Enables or disables the pulse width change.
High Speed Duty Cycle Distortion A/B/C	Sets the DCD value for line A/B/C. Positive DCD value shifts the rising and falling edges outwards, increasing the duration of levels. Negative DCD value shifts the rising and falling edges inwards, decreasing the duration of levels. DCD changes effects only on the respective the line (A/B/C).
Low Power Pulse Width A/B/C	Sets the duration of high levels by desired pulse value and adjust the difference of 2*UI and pluse width duration in low levels on line A/B/C, respectively.

Low Power Mode

The parameters provided by **Low Power Mode** are shown in the Figure 15 on page 44.

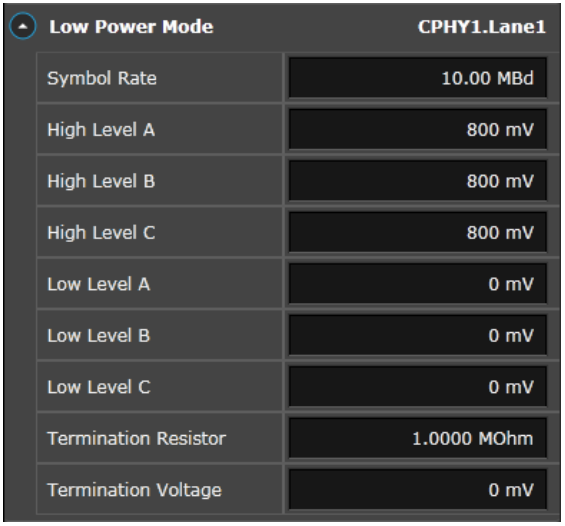


Figure 15 Low Power Mode Parameters

The parameters provided by the **Low Power Mode** are described in Table 15 on page 44.

Table 15 Low Power Mode

Parameter	Description
Symbol Rate	Sets the symbol rate of low power transmission.
High Level A/B/C	Sets the high level of low power mode on line A, B or C, respectively.
Low Level A/B/C	Sets the low level of low power mode on line A, B or C, respectively.
Termination Resistor	Sets the resistance of the termination resistors in low power mode.
Termination Voltage	Sets the termination voltage for low power mode. The receiver input is transmitted against this voltage.

Pattern

The **Pattern** parameters are shown in the Figure 16 on page 45.

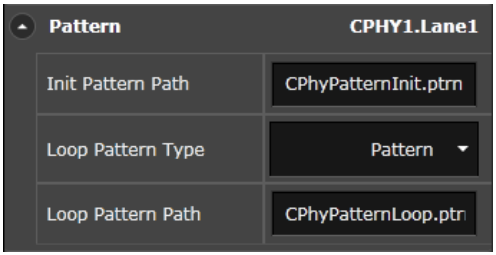


Figure 16 Pattern Parameters

The **Pattern** parameters are described in Table 16 on page 45.

Table 16 Patterns Parameter

Parameter	Description
Init Pattern Path	Sets the file path for a file containing a C-PHY pattern. The path is relative in the user documentation folder. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window. This pattern is played one time after starting the AWG.
Loop Pattern Type	Selects the type of loop pattern. You can either select the user defines pattern file or PRBS generated.
Loop Pattern Path	Sets the file path for a file containing a C-PHY pattern. The path is relative in the user documentation folder. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window. This pattern is looped continuously after the initial pattern.
PRBS Seed	The “**PRBS Seed**” option is available when the **Loop Pattern Type** is selected as PRBS generated. PRBS seed is either prefixed with "0b" in binary format, "0x" in hex format, or no prefix is present in integer format. Depending on the selected PRBS, a specific number of bits is used for filling the seed register. Surplus bits are ignored. For PRBS 9 and 11, the seed register is 16 bit wide for PRBS 18 it is 18 bit wide. For all three PRBS the tap where the actual PRBS data is taken from is at bit position 16. This leads to PRBS 9 having 7 bits definable by the seed which are only being generated once while afterwards the PRBS loops. For PRBS 11 this is true for the first 5 bits while for PRBS 18 the tap is still at position 16 but the PRBS data will repeat immediately.

Protocol

The parameters provided by **Protocol** setting are shown in the Figure 17 on page 46.

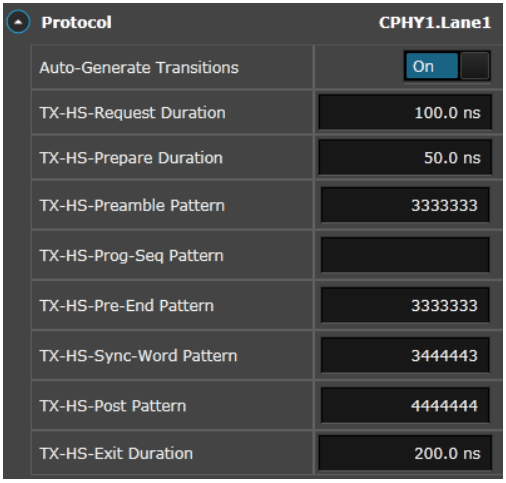


Figure 17 Protocol settings

The parameters provided by the **Protocol** settings are described in Table 17 on page 46.

Table 17 Protocol Settings

Parameter	Description
Auto-Generate Transitions	Turn on this toggle switch to automatically generate LP -> HS and HS -> LP transition. In case there is no start wire state for the HS pattern defined in the pattern file a default wire state will be used. The initial wire state of the preamble pattern is either a start wire state defined in the preamble pattern or in the HS burst pattern definition causing the LP-> HS transition to be generated. In case neither of these two pattern define an initial wire state a default state is used.
TX-HS-Request Duration	Sets the length of TX-HS-Request state.
TX-HS-Prepare Duration	Sets the length TX-HS-Prepare state.
TX-HS-Preamble Pattern	Sets the TX-HS-Preamble pattern in high-speed C-PHY pattern format.

Parameter	Description
TX-HS-Prog-Seq Pattern	Sets the optional TX-HS-Prog-Seq pattern n high-speed C-PHY pattern format.
TX-HS-Pre-End Pattern	Sets the TX-HS-Pre-End pattern in high-speed C-PHY pattern format. By default, this pattern contains a total of 7 symbols of type 3.
TX-HS-Sync-Word Pattern	Sets the TX-HS-Sync-Word pattern which is send immediately before staring high-speed transmission. By default, the pattern is 3444443.
TX-HS-Post Pattern	Sets the TX-HS-Post pattern which is send immediately after staring high-speed transmission.
TX-HS-Exit Duration	Sets the length of LP-111 state following a high-speed burst.

Trigger

To test the minimum initialization period (T_{INIT}) needed by DUT to recognize the valid test sequence, it is done by sending the LP111 sequence and slowly increasing the length of LP111 state. When DUT starts recognizing the valid sequences of LP and HS then record the duration of LP111. In order to achieve this test, DUT must be started in state other than LP111, and to support this DUT can now be started with LP000 state in triggered mode. So if DUT is started in triggered mode then DUT will only see LP000 state i.e. low on all three lines, without actually running any patterns applied by the user. When triggered mode is changed to Immediate mode then user patterns are applied and DUT can start seeing the valid sequence depending on the length of LP111.

The parameters provided by **Trigger** setup are shown in the Figure 18 on page 47.

Figure 18 Trigger Setup Parameters

The parameters provided by the **Trigger** setup are described in Table 18 on page 48.

Table 18 Trigger Setup

Parameter	Description
Trigger Start	Sets the static LP STOP signal.
High Level Trigger	Sets the high level of trigger.
Low Level Trigger	Sets the low level of trigger.

Amplifier

The parameters provided by **Amplifier** setup are shown in the Figure 19 on page 48.

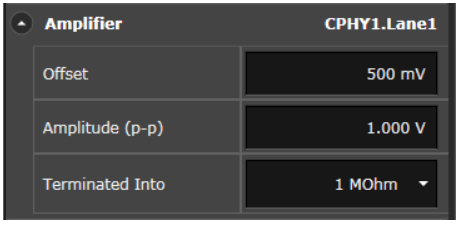


Figure 19 Amplifier Parameters

The parameters provided by the **Amplifier** setup are described in Table 19 on page 49.

Table 19 Amplifier

Parameter	Description
Offset	Sets the amplifier offset for the AWG output amplifiers participating in C-PHY signal generation. This value determines statically how the output amplifier is set during run mode. Subsequent amplitude changes for HS and LP modes will be limited by this value. Its magnitude is defined in respect to a DUT with a selectable input resistance of 50 Ohms or infinite. In case of DUT having 50 Ohms input resistance the actually visible maximum voltage will be halved as the output resistance of the AWG amplifier is also 50 Ohms.
Amplitude (p-p)	Sets the peak-peak amplitude for the AWG output amplifiers participating in C-PHY signal generation. This value determines statically how the output amplifier is set during run mode. Subsequent amplitude changes for HS and LP modes will be limited by this value. Its magnitude is defined in respect to a DUT with a selectable input resistance of 50 Ohms or infinite. In case of DUT having 50 Ohms input resistance the actually visible minimum voltage will be halved as the output resistance of the AWG amplifier is also 50 Ohms.
Terminated Into	Sets the value of termination voltage against 50 Ohms or 1 MOhm.

Intersymbol Interference

The MIPI C-PHY Editor provides Intersymbol Interference (ISI) capability to test next-generation high-speed digital designs. It allows emulating channel loss for C-PHY testing of high-speed digital receivers. The S-Parameter files are treated as single ended per wire. Only the forward transmission term (S21) is used for signal distortion.

In order to support both file formats (S2P and S6P), GUI option is provided to select between 2 port (S2P) and 6 port (S6P). Selecting 2 ports will show 3 S-parameter fields to enter S2P files for individual line. If field(s) are left blank then ISI will be disabled on respective line(s). Selecting 6 ports option will show single S-parameter field to enter S6P file for all lines but trigger. Leaving S6P field blank will disable ISI on all 3 lines.

The parameters provided by **Intersymbol Interference** setup are shown in the Figure 19 on page 48.

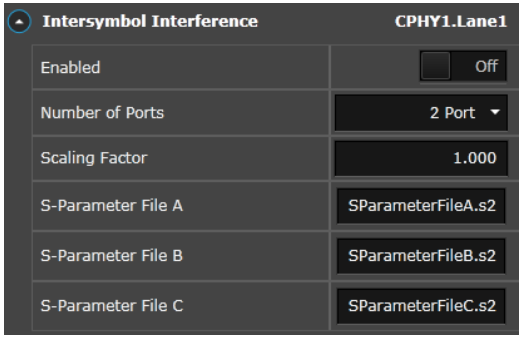


Figure 20 Intersymbol Interference Parameters

The parameters provided by the **Intersymbol Interference** setup are described in Table 19 on page 49.

Table 20 Amplifier

Parameter	Description
Enabled Button	Enables/disables the ISI feature.
Number of Ports	Sets number of input and output ports i.e. for two ports, there will be two input and two output ports.
Scaling Factor	Sets the values that increases or decreases the effect of S-parameter measurement freely. Changing scaling parameter will affect the magnitude and phase of S-parameters.
S-Parameter File A	Sets the file path for a file containing a S-Parameter information for line A. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window.
S-Parameter File B	Sets the file path for a file containing a S-Parameter information for line B. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window.
S-Parameter File C	Sets the file path for a file containing a S-Parameter information for line C. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window.
6 Port S-Parameter File	File path for a file containing a S-Parameter information of 6 ports. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window.

NOTE When changing the ISI properties during run mode a voltage glitch may occur.

AWG Setup

The parameters provided by **AWG Setup** are shown in the Figure 21 on page 51.

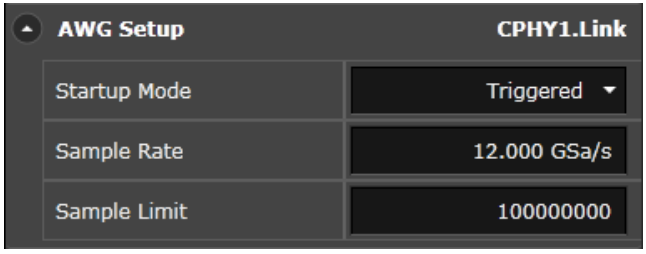


Figure 21 AWG Setup Parameters

The parameters provided by the **AWG Setup** are described in Table 21 on page 51.

Table 21 AWG Setup

Parameter	Description
Start Mode	Sets whether signal generation starts immediately or if it should wait for a trigger event. ▪ **Immediate:** Starts the signal generation immediately. ▪ **Triggered:** Will fully prepare AWGs for immediate startup. After preparation, the LP high level will be active on all wires of the lane. Using this a DUT can be started up receiving the required STOP state.
Sample Rate	Set the AWG sample rate.
Sample Limit	Sets the maximum number of samples the data stream which is downloaded into the AWG can contain. Increasing this number will allow large pattern to be repeated with higher accuracy. But increasing it will also increase download time. Pattern rollout occurs mainly for odd ratios of sample rate and symbol rate and jitter frequency, if applicable.

AWG1 Setup

The parameters provided by **AWG1 Setup** are shown in the Figure 22 on page 52.

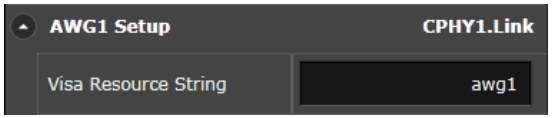


Figure 22 AWG1 Setup Parameters

The parameters provided by the **AWG1 Setup** are described in Table 22 on page 52.

Table 22 AWG1 Setup

Parameter	Description
Visa Resource String	Sets the visa connection string for the AWG1.

AWG2 Setup

The parameters provided by **AWG2 Setup** are shown in the Figure 23 on page 52.

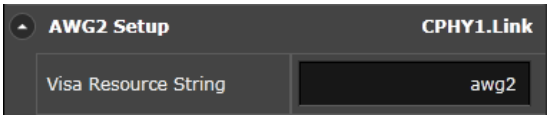


Figure 23 AWG2 Setup Parameters

The parameters provided by the **AWG2 Setup** are described in Table 23 on page 52.

Table 23 AWG2 Setup

Parameter	Description
Visa Resource String	Sets the visa connection string for the AWG2.

Calibration

In order to support de-skew calibration, an option in GUI is provided to select "Normal" or "Complement" port.

Selecting "Normal" port will behave as it was behaving before, i.e. Calibration is done on "Normal" port, keeping both normal and complement output of AWG's ON.

Selecting "Complement" port, calibration will be done on "Complement" port, during calibration complement and normal output of AWG's are switched ON and OFF respectively. After calibration is done, complement and normal outputs are switched OFF and ON respectively.

The parameters provided by **Calibration** is shown in the Figure 24 on page 53.

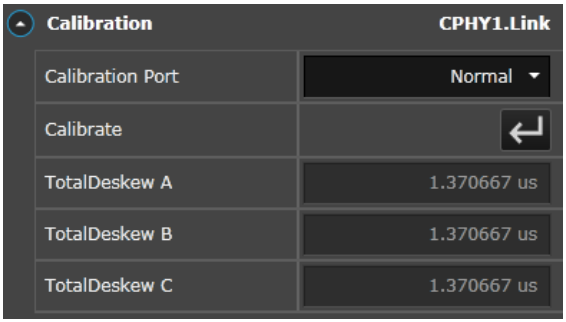


Figure 24 Calibration Parameters

The parameter provided by the **Calibration** is described in Table 24 on page 53.

Table 24 Calibration

Parameter	Description
Calibration Port	Sets the calibration on Normal or Compliment port.
Calibrate	Press ↵ button to calibrate the selected AWG properties like skew and amplitude.
Total Deskew A/B/C	Sets the Deskew of line A/B/C and delay segments.

Oscilloscope Setup

The parameters provided by **Oscilloscope Setup** are shown in the Figure 25 on page 54.

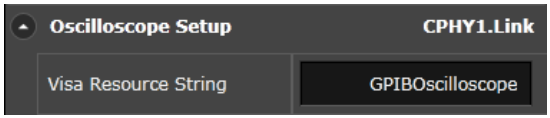


Figure 25 Oscilloscope Setup Parameters

The parameters provided by the **Oscilloscope Setup** are described in Table 25 on page 54.

Table 25 Oscilloscope Setup

Parameter	Description
Visa Resource String	Sets the visa connection string for the oscilloscope.

Sequence

The parameters provided by **Sequence** are shown in the Figure 26 on page 54.

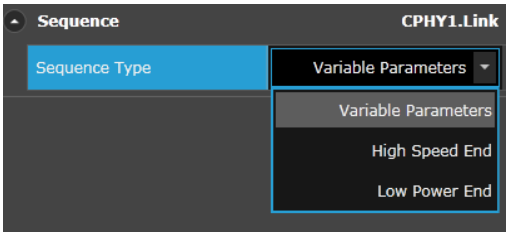


Figure 26 Sequence Parameters

The parameters provided by the **Sequence** are described in Table 26 on page 55.

Table 26 Sequence Type

Parameter	Description
Sequence	Selects the type of sequence statically before downloading any pattern in order to support different kind of sequence scenarios. The available sequence types are Variable Parameters, High Speed End and Low Power End.

High Speed End Sequence

The following figure illustrates the high speed end sequence.

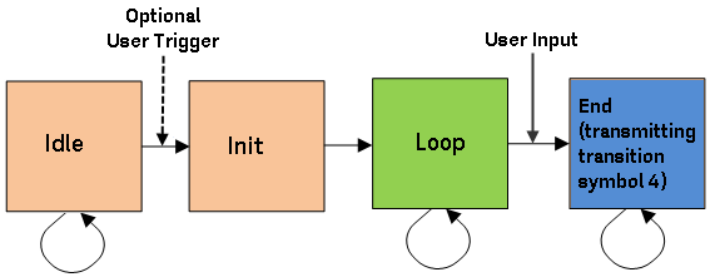


Figure 27 High Speed End Sequence

This sequence begins with playing the idle pattern which can be optionally used with the GUI/SCPI or hardware trigger.

Next is the user definable "Init" pattern which is played back once and then the looping "A" pattern. Both of these patterns are user definable via pattern file. When the A pattern is reached it is possible to break out of the loop via GUI/SCPI trigger. Following is the "B" pattern which loops the transition symbol 4. This means that depending on the last wire state active in the "A" pattern the "B" pattern will continue with periodically negating this last wire state. As an example when the last wire state in "A" is '-z' the "B" segment starts with '+z' and continues with '-z +z -z +z...' until the stop command is executed.

The user definable patterns "Init" and "A" are only allowed to contain high speed symbols. As "over-programming" it allow to also use LP symbols and transitions but doing this may result in a corrupted waveform.

As a prerequisite for a valid waveform it is required that the "A" pattern ends in high speed mode. Otherwise it is not possible to use a high speed wire state to have a valid transition into the "B" segment.

Impairments are not available in this operating mode.

Low Power End Sequence

The following figure illustrates the low power end sequence.

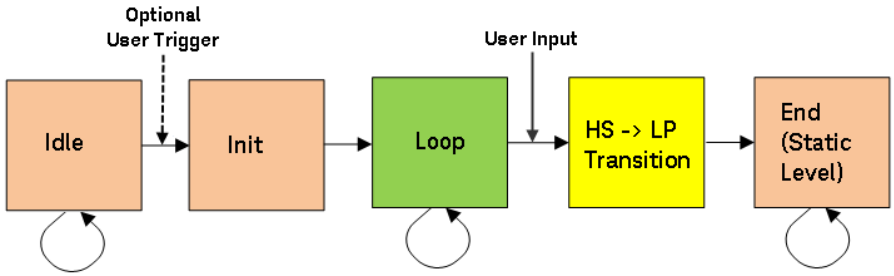


Figure 28 Low Power End Sequence

Sequence begins by playing the idle pattern which can be optionally used with the GUI/SCPI or hardware trigger.

Next is the user definable "Init" pattern which is played back once and then pattern "A" is looped. Both of these patterns are user definable via pattern file. When the A pattern is reached it is possible to break out of the loop via GUI/SCPI trigger. Following is the "End" pattern which contains a single transition from HS to LP resulting in a static LP111 at its end. Which is followed by Static Level pattern which transmits a static LP111 symbol. For End pattern to execute correctly the last symbol in pattern "A" needs to be a high speed symbol.

The user definable patterns "Init" and "A" are only allowed to contain high speed symbols. As "over-programming" it is allowed to also use LP symbols and transitions but doing this may result in a corrupted waveform.

As a prerequisite for a valid waveform it is required that the "A" pattern ends in high speed mode. Otherwise it is not possible to use a high speed wire state to have a valid transition into the "End" segment.

Impairments are not available in this operating mode.

Output

The following figure shows the **MIPI C-PHY Editor** output when viewed on the oscilloscopes. It shows the overlays a lot of states to give an eye diagram, the mid level of wire a is shifted up slightly:

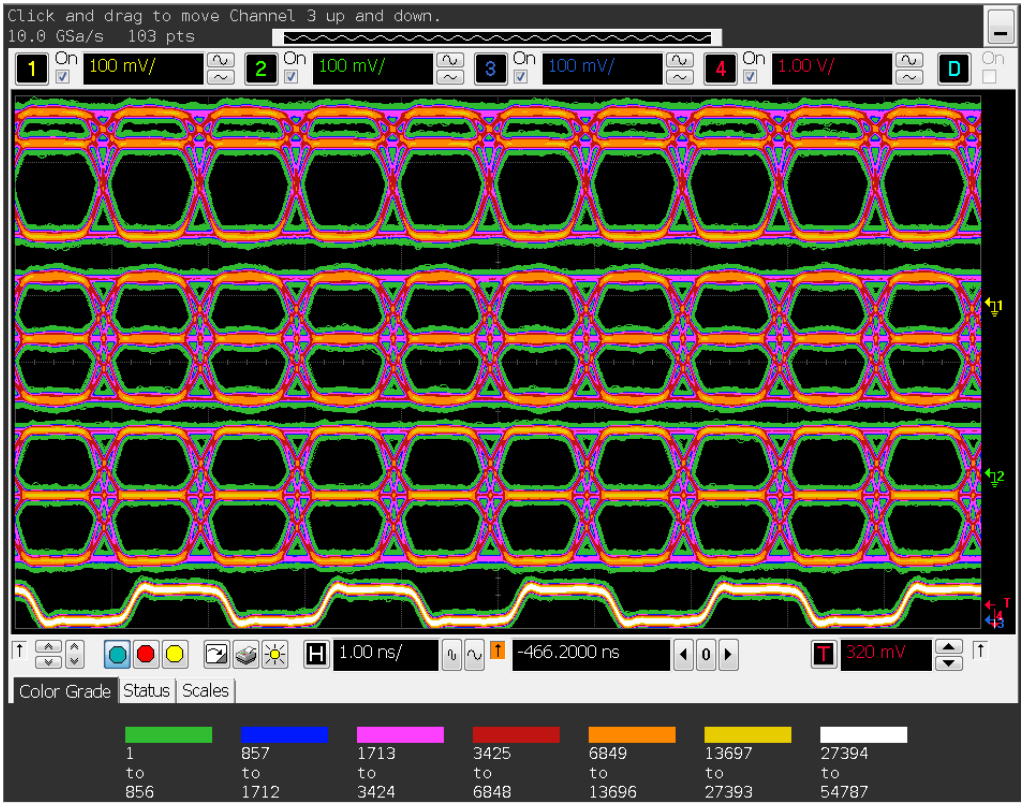


Figure 29 MIPI C-PHY Editor output-1 on oscilloscope

The following figure shows the **MIPI C-PHY Editor** output for the actual state of all three wires at a specific time:

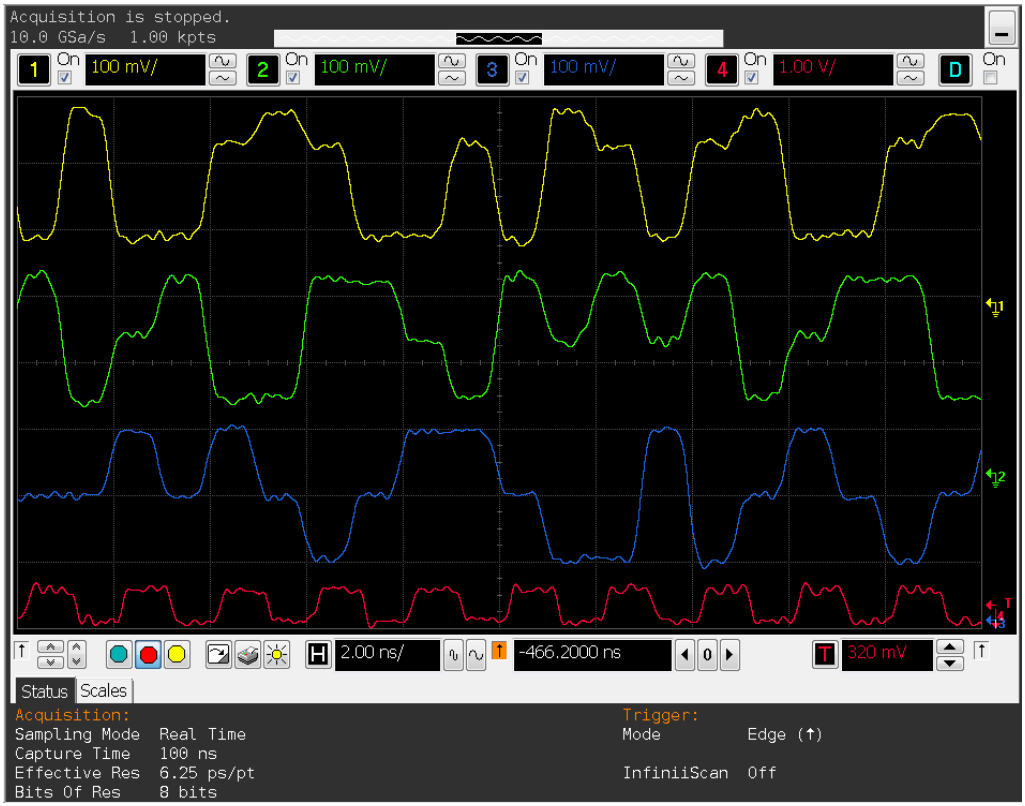


Figure 30 MIPI C-PHY Editor output-2 on oscilloscope

3 SCPI Programming

SCPI Command Language

The M8020A 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

For more details on command sets, refer to *M8020A Programming Guide*.

Data Types

The M8020A 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.

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

The following SCPI commands provide examples of this.

:PLUGin:CPHY:PATTern:LOOP:TYPE 'MIPI C-PHY Editor 1',P9

These commands can also be sent as follows:

:PLUG:CPHY:PATT:LOOP:TYPE 'MIPI C-PHY Editor 1',P9

SCPI Command Structure Example

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

:PLUGin:CPHY:PATTern:LOOP:TYPE 'MIPI C-PHY Editor 1',P9

NOTE

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

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

The bar symbol (|) indicates mutually exclusive commands.

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

In long form:

:PLUGin:CPHY:PATTern:LOOP:TYPE 'MIPI C-PHY Editor 1',P9

In short form:

:PLUG:CPHY:PATT:LOOP:TYPE 'MIPI C-PHY Editor 1',P9

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:

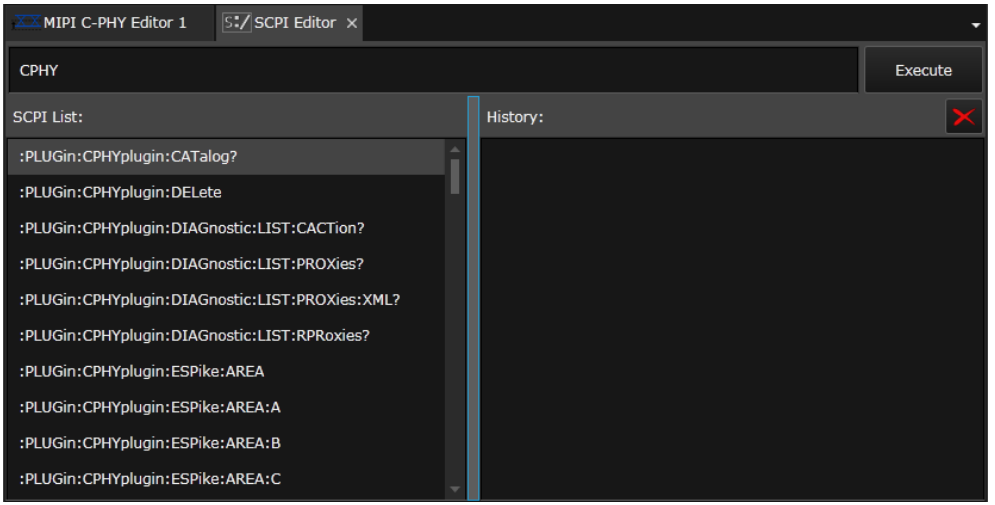


Figure 31 M8020A SCPI Editor

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 to set the loop type for a pattern as PRBS 2^9-1:

  :PLUGin:CPHY:PATTern:LOOP:TYPE 'MIPI C-PHY Editor 1',P9
- 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)

For further details on SCPI command language, refer to *M8020A Programming Guide*.

SCPI Command Reference

The C-PHY Editor has the following commands:

Table 27 SCPI Commands

Command	Description under
:PLUGin:CPHYplugin:DELete	For details, see :PLUGin:CPHYplugin:DELete on page 72.
:PLUGin:CPHYplugin:HSMode:SYMRate[?]	For details, see :PLUGin:CPHYplugin:HSMode:SYMRate[?] on page 72.
:PLUGin:CPHYplugin:HSMode:TERMination:RESistor[?]	For details, see :PLUGin:CPHYplugin:HSMode:TERMination:RESistor[?] on page 72.
:PLUGin:CPHYplugin:HSMode:TERMination:VOLTage[?]	For details, see :PLUGin:CPHYplugin:HSMode:TERMination:VOLTage[?] on page 72.
:PLUGin:CPHYplugin:HSMode:VOLTage:A:HIGH[?]	For details, see :PLUGin:CPHYplugin:HSMode:VOLTage:A:HIGH[?] on page 73.
:PLUGin:CPHYplugin:HSMode:VOLTage:A:LOW[?]	For details, see :PLUGin:CPHYplugin:HSMode:VOLTage:A:LOW[?] on page 73.
:PLUGin:CPHYplugin:HSMode:VOLTage:A:MID[?]	For details, see :PLUGin:CPHYplugin:HSMode:VOLTage:A:MID[?] on page 73.
:PLUGin:CPHYplugin:HSMode:VOLTage:B:HIGH[?]	For details, see :PLUGin:CPHYplugin:HSMode:VOLTage:B:HIGH[?] on page 74.
:PLUGin:CPHYplugin:HSMode:VOLTage:B:LOW[?]	For details, see :PLUGin:CPHYplugin:HSMode:VOLTage:B:LOW[?] on page 74.
:PLUGin:CPHYplugin:HSMode:VOLTage:B:MID[?]	For details, see :PLUGin:CPHYplugin:HSMode:VOLTage:B:MID[?] on page 74.
:PLUGin:CPHYplugin:HSMode:VOLTage:C:HIGH[?]	For details, see :PLUGin:CPHYplugin:HSMode:VOLTage:C:HIGH[?] on page 74.
:PLUGin:CPHYplugin:HSMode:VOLTage:C:LOW[?]	For details, see :PLUGin:CPHYplugin:HSMode:VOLTage:C:LOW[?] on page 75.

Command	Description under
:PLUGin:CPHYplugin:HSMode:VOLTage:C:MID[?]	For details, see :PLUGin:CPHYplugin:HSMode:VOLTage:C:MID[?] on page 75.
:PLUGin:CPHYplugin:IMPairments:HSFTime[?]	For details, see :PLUGin:CPHYplugin:IMPairments:HSFTime[?] on page 75.
:PLUGin:CPHYplugin:IMPairments:HSRTime[?]	For details, see :PLUGin:CPHYplugin:IMPairments:HSRTime[?] on page 76.
:PLUGin:CPHYplugin:IMPairments:JITTer:BUJ:RMS[?]	For details, see :PLUGin:CPHYplugin:IMPairments:JITTer:BUJ:RMS[?] on page 76.
:PLUGin:CPHYplugin:IMPairments:JITTer:SINusodial:AMPlitude[?]	For details, see :PLUGin:CPHYplugin:IMPairments:JITTer:SINusodial:AMPlitude[?] on page 76.
:PLUGin:CPHYplugin:IMPairments:JITTer:SINusodial:FREQuency[?]	For details, see :PLUGin:CPHYplugin:IMPairments:JITTer:SINusodial:FREQuency[?] on page 77.
:PLUGin:CPHYplugin:IMPairments:SKEW:A[?]	For details, see :PLUGin:CPHYplugin:IMPairments:SKEW:A[?] on page 77.
:PLUGin:CPHYplugin:IMPairments:SKEW:B[?]	For details, see :PLUGin:CPHYplugin:IMPairments:SKEW:B[?] on page 77.
:PLUGin:CPHYplugin:IMPairments:SKEW:C[?]	For details, see :PLUGin:CPHYplugin:IMPairments:SKEW:C[?] on page 77.
:PLUGin:CPHYplugin:IMPairments:HISPeed:DCD:A/B/C[?]	For details, see :PLUGin:CPHYplugin:IMPairments:HISPeed:DCD:A/B/C[?] on page 78.
:PLUGin:CPHYplugin:IMPairments:HISPeed:DCD:A/B/C[?]	For details, see :PLUGin:CPHYplugin:IMPairments:HISPeed:DCD:A/B/C[?] on page 78.
:PLUGin:CPHYplugin:IMPairments:HISPeed:DCD:A/B/C[?]	For details, see :PLUGin:CPHYplugin:IMPairments:HISPeed:DCD:A/B/C[?] on page 78.
:PLUGin:CPHYplugin:LPMode:SYMRate[?]	For details, see :PLUGin:CPHYplugin:LPMode:SYMRate[?] on page 78.
:PLUGin:CPHYplugin:LPMode:TERMination:RESistor[?]	For details, see :PLUGin:CPHYplugin:LPMode:TERMination:RESistor[?] on page 79.

Command	Description under
:PLUGin:CPHYplugin:LPMode:TERMination:VOLTage[?]	For details, see :PLUGin:CPHYplugin:LPMode:TERMination:VOLTage[?] on page 79.
:PLUGin:CPHYplugin:LPMode:VOLTage:HIGH[?]	For details, see :PLUGin:CPHYplugin:LPMode:VOLTage:HIGH[?] on page 79.
:PLUGin:CPHYplugin:LPMode:VOLTage:LOW[?]	For details, see :PLUGin:CPHYplugin:LPMode:VOLTage:LOW[?] on page 79.
:PLUGin:CPHYplugin:LPMode:VOLTage:A:HIGH[?]	For details, see :PLUGin:CPHYplugin:LPMode:VOLTage:A:HIGH[?] on page 80.
:PLUGin:CPHYplugin:LPMode:VOLTage:A:LOW[?]	For details, see :PLUGin:CPHYplugin:LPMode:VOLTage:A:LOW[?] on page 80.
:PLUGin:CPHYplugin:LPMode:VOLTage:B:HIGH[?]	For details, see :PLUGin:CPHYplugin:LPMode:VOLTage:B:HIGH[?] on page 80.
:PLUGin:CPHYplugin:LPMode:VOLTage:B:LOW[?]	For details, see :PLUGin:CPHYplugin:LPMode:VOLTage:B:LOW[?] on page 80.
:PLUGin:CPHYplugin:LPMode:VOLTage:C:HIGH[?]	For details, see :PLUGin:CPHYplugin:LPMode:VOLTage:C:HIGH[?] on page 80.
:PLUGin:CPHYplugin:LPMode:VOLTage:C:LOW[?]	For details, see :PLUGin:CPHYplugin:LPMode:VOLTage:C:LOW[?] on page 81.
:PLUGin:CPHYplugin:NEW	For details, see :PLUGin:CPHYplugin:NEW on page 81.
:PLUGin:CPHY:PATTern:INIT:PATH[?]	For details, see :PLUGin:CPHY:PATTern:INIT:PATH[?] on page 81.
:PLUGin:CPHY:PATTern:LOOP:TYPE[?]	For details, see :PLUGin:CPHY:PATTern:LOOP:TYPE[?] on page 81.
:PLUGin:CPHY:PATTern:LOOP:PATH[?]	For details, see :PLUGin:CPHY:PATTern:LOOP:PATH[?] on page 82.
:PLUGin:CPHY:PATTern:LOOP:SEED[?]	For details, see :PLUGin:CPHY:PATTern:LOOP:SEED[?] on page 82.

Command	Description under
:PLUGin:CPHYplugin:PROTocol:TRANsition:AUTO[?]	For details, see :PLUGin:CPHYplugin:PROTocol:TRANsition:AUTO[?] on page 82.
:PLUGin:CPHYplugin:PROTocol:REQuest:DURation[?]	For details, see :PLUGin:CPHYplugin:PROTocol:REQuest:DURation[?] on page 83.
:PLUGin:CPHYplugin:PROTocol:PREPare:DURation[?]	For details, see :PLUGin:CPHYplugin:PROTocol:PREPare:DURation[?] on page 83.
:PLUGin:CPHYplugin:PROTocol:PREAmble:BEGin:PATTern[?]	For details, see :PLUGin:CPHYplugin:PROTocol:PREAmble:BEGin:PATTern[?] on page 83.
:PLUGin:CPHYplugin:PROTocol:PREAmble:PROGsequence:PATTern[?]	For details, see :PLUGin:CPHYplugin:PROTocol:PREAmble:PROGsequence:PATTern[?] on page 83.
:PLUGin:CPHYplugin:PROTocol:PREAmble:END:PATTern[?]	For details, see :PLUGin:CPHYplugin:PROTocol:PREAmble:END:PATTern[?] on page 84.
:PLUGin:CPHYplugin:PROTocol:SYNCword:PATTern[?]	For details, see :PLUGin:CPHYplugin:PROTocol:SYNCword:PATTern[?] on page 84.
:PLUGin:CPHYplugin:PROTocol:POST:PATTern[?]	For details, see :PLUGin:CPHYplugin:PROTocol:POST:PATTern[?] on page 84.
:PLUGin:CPHYplugin:PROTocol:EXIT:DURation[?]	For details, see :PLUGin:CPHYplugin:PROTocol:EXIT:DURation[?] on page 84.
:PLUGin:CPHYplugin:RESet[?]	For details, see :PLUGin:CPHYplugin:RESet[?] on page 85.
:PLUGin:CPHYplugin:LINK:SAMPlerate[?]	For details, see :PLUGin:CPHYplugin:LINK:SAMPlerate[?] on page 85.
:PLUGin:CPHYplugin:LINK:AWGA[?]	For details, see :PLUGin:CPHYplugin:LINK:AWGA[?] on page 85.
:PLUGin:CPHYplugin:LINK:AWGB[?]	For details, see :PLUGin:CPHYplugin:LINK:AWGB[?] on page 85.
:PLUGin:CPHYplugin:LINK:OSCilloscpe[?]	For details, see :PLUGin:CPHYplugin:LINK:OSCilloscpe[?] on page 86.

Command	Description under
:PLUGin:CPHYplugin:LINK:STAMode[?]	For details, see :PLUGin:CPHYplugin:LINK:STAMode[?] on page 86.
:PLUGin:DPHYplugin:LINK:TRIGger:EXECute	For details, see :PLUGin:DPHYplugin:LINK:TRIGger[:EXECute] on page 86.
:PLUGin:CPHYplugin:LINK:CALibration:PORT[?]	For details, see :PLUGin:CPHYplugin:LINK:CALibration:PORT[?] on page 86.
:PLUGin:CPHYplugin:LINK:CALibration:DESKew:A?	For details, see :PLUGin:CPHYplugin:LINK:CALibration:DESKew:A? on page 87.
:PLUGin:CPHYplugin:LINK:CALibration:DESKew:B?	For details, see :PLUGin:CPHYplugin:LINK:CALibration:DESKew:B? on page 87.
:PLUGin:CPHYplugin:LINK:CALibration:DESKew:C?	For details, see :PLUGin:CPHYplugin:LINK:CALibration:DESKew:C? on page 87.
:PLUGin:CPHYplugin:LINK:CALibration:DESKew:D?	For details, see :PLUGin:CPHYplugin:LINK:CALibration:DESKew:D? on page 87.
:PLUGin:CPHYplugin:RUN[:STATus]?	For details, see :PLUGin:CPHYplugin:RUN[:STATus]? on page 87.
:PLUGin:CPHYplugin:RUN:PROGress?	For details, see :PLUGin:CPHYplugin:RUN:PROGress? on page 87.
:PLUGin:CPHYplugin:STARt	For details, see :PLUGin:CPHYplugin:STARt on page 88.
:PLUGin:CPHYplugin:STOP	For details, see :PLUGin:CPHYplugin:STOP on page 88.
:PLUGin:CPHYplugin:TRIGger:STARt[?]	For details, see :PLUGin:CPHYplugin:TRIGger:STARt[?] on page 88.
:PLUGin:CPHYplugin:TRIGger:VOLTage:HIGH[?]	For details, see :PLUGin:CPHYplugin:TRIGger:VOLTage:HIGH[?] on page 88.
:PLUGin:CPHYplugin:TRIGger:VOLTage:LOW[?]	For details, see :PLUGin:CPHYplugin:TRIGger:VOLTage:LOW[?] on page 89.
:PLUGin:CPHYplugin:ESPike:MODE[?]	For details, see :PLUGin:CPHYplugin:ESPike:MODE[?] on page 89.
:PLUGin:CPHYplugin:ESPike:AREA[?]	For details, see :PLUGin:CPHYplugin:ESPike:AREA[?] on page 89.

Command	Description under
:PLUGin:CPHYplugin:ESPike:LHIGhinvol[?]	For details, see :PLUGin:CPHYplugin:ESPike:LHIGhinvol[?] on page 90.
:PLUGin:CPHYplugin:ESPike:LLOWintvol[?]	For details, see :PLUGin:CPHYplugin:ESPike:LLOWintvol[?] on page 90.
:PLUGin:CPHYplugin:ESPike:MODE:A[?	For details, see :PLUGin:CPHYplugin:ESPike:MODE:A[? on page 91.
:PLUGin:CPHYplugin:ESPike:MODE:B[?	For details, see :PLUGin:CPHYplugin:ESPike:MODE:B[? on page 91.
:PLUGin:CPHYplugin:ESPike:MODE:C[?	For details, see :PLUGin:CPHYplugin:ESPike:MODE:C[? on page 91.
:PLUGin:CPHYplugin:VOLTage:OFFSet[?]	For details, see :PLUGin:CPHYplugin:VOLTage:OFFSet[?] on page 91.
:PLUGin:CPHYplugin:VOLTage:AMPlitude[?]	For details, see :PLUGin:CPHYplugin:VOLTage:AMPlitude[?] on page 92.
:PLUGin:CPHYplugin:VOLTage:TERMination[?]	For details, see :PLUGin:CPHYplugin:VOLTage:TERMination[?] on page 92.
:PLUGin:CPHYplugin:LINK:CALibration:EXECute	For details, see :PLUGin:CPHYplugin:LINK:CALibration[:EXECute] on page 92.
:PLUGin:CPHYplugin:ISI:ENABle[?]	For details, see :PLUGin:CPHYplugin:ISI:ENABle[?] on page 92.
:PLUGin:CPHYplugin:ISI:PATH:A[?]	For details, see :PLUGin:CPHYplugin:ISI:PATH:A[?] on page 93.
:PLUGin:CPHYplugin:ISI:PATH:B[?]	For details, see :PLUGin:CPHYplugin:ISI:PATH:B[?] on page 93.
:PLUGin:CPHYplugin:ISI:PATH:C[?]	For details, see :PLUGin:CPHYplugin:ISI:PATH:C[?] on page 93.
:PLUGin:CPHYplugin:ISI:NUMPorts[?]	For details, see :PLUGin:CPHYplugin:ISI:NUMPorts[?] on page 93.
:PLUGin:CPHYplugin:ISI:PATH[?]	For details, see :PLUGin:CPHYplugin:ISI:PATH[?] on page 94.
:PLUGin:CPHYplugin:ISI:SCALe[?]	For details, see :PLUGin:CPHYplugin:ISI:SCALe[?] on page 94.
:PLUGin:CPHYplugin:HYSTeresis:VOLTage:A/B/C[?]	For details, see :PLUGin:CPHYplugin:HYSTeresis:VOLTage:A/B/C[?] on page 95.
:PLUGin:CPHYplugin:LINK:RSEQuence	For details, see :PLUGin:CPHYplugin:LINK:RSEQuence on page 95.
:PLUGin:CPHYplugin:LINK:SEQType[?]	For details, see :PLUGin:CPHYplugin:LINK:SEQType[?] on page 95.

:PLUGin:CPHYplugin:DELete

Syntax :PLUGin:CPHYplugin:DELete

Description This command is used to delete the C_PHY Plugin.

:PLUGin:CPHYplugin:HSMode:SYMRate[?]

Syntax :PLUGin:CPHYplugin:HSMode:SYMRate 'MIPI C-PHY Editor 1',1.00000000000000E+08

:PLUGin:CPHYplugin:HSMode:SYMRate? 'MIPI C-PHY Editor 1'

Description This command sets the symbol rate of C-PHY signal. This is the actual rate of circuit. The effective data rate of the transmission signal will be 2.28 higher than the selected value.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:TERMination:RESistor[?]

Syntax :PLUGin:CPHYplugin:HSMode:TERMination:RESistor 'MIPI C-PHY Editor 1',2.50000000000000E+01

:PLUGin:CPHYplugin:HSMode:TERMination:RESistor? 'MIPI C-PHY Editor 1'

Description This command sets resistance of the termination resistors. There is only one value for all three lines.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:TERMination:VOLTage[?]

Syntax :PLUGin:CPHYplugin:HSMode:TERMination:VOLTage 'MIPI C-PHY Editor 1',2.00000000000000E-01

:PLUGin:CPHYplugin:HSMode:TERMination:VOLTage? 'MIPI C-PHY Editor 1'

Description This command sets the internal termination voltage of the DUT's receiver circuit. For the symmetrical case where the resistance of the receiver circuit line terminator are 50 Ohms each and the voltage of all three lines

are symmetrical around the same mid level, this value needs to be set to the mid level. For the asymmetrical case, the actual termination voltage at the receiver needs to be chosen accordingly. If this is not done the voltages seen at the receiver input will be different from the selected value.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:VOLTage:A:HIGH[?]

Syntax :PLUGin:CPHYplugin:HSMode:VOLTage:A:HIGH 'MIPI C-PHY Editor 1',4.00000000000000E-01

:PLUGin:CPHYplugin:HSMode:VOLTage:A:HIGH? 'MIPI C-PHY Editor 1'

Description This command sets the high level of line A.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:VOLTage:A:LOW[?]

Syntax :PLUGin:CPHYplugin:HSMode:VOLTage:A:LOW 'MIPI C-PHY Editor 1',2.00000000000000E-01

:PLUGin:CPHYplugin:HSMode:VOLTage:A:LOW? 'MIPI C-PHY Editor 1'

Description This command sets the low level of line A.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:VOLTage:A:MID[?]

Syntax :PLUGin:CPHYplugin:HSMode:VOLTage:A:MID 'MIPI C-PHY Editor 1',3.00000000000000E-01

:PLUGin:CPHYplugin:HSMode:VOLTage:A:MID? 'MIPI C-PHY Editor 1'

Description This command sets the mid level of line A.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:VOLTage:B:HIGH[?]

Syntax :PLUGin:CPHYplugin:HSMode:VOLTage:B:HIGH 'MIPI C-PHY Editor 1',4.00000000000000E-01

:PLUGin:CPHYplugin:HSMode:VOLTage:B:HIGH? 'MIPI C-PHY Editor 1'

Description This command sets the high level of line B.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:VOLTage:B:LOW[?]

Syntax :PLUGin:CPHYplugin:HSMode:VOLTage:B:LOW 'MIPI C-PHY Editor 1',2.00000000000000E-01

:PLUGin:CPHYplugin:HSMode:VOLTage:B:LOW? 'MIPI C-PHY Editor 1',

Description This command sets the low level of line B.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:VOLTage:B:MID[?]

Syntax :PLUGin:CPHYplugin:HSMode:VOLTage:B:MID 'MIPI C-PHY Editor 1',3.00000000000000E-01

:PLUGin:CPHYplugin:HSMode:VOLTage:B:MID? 'MIPI C-PHY Editor 1'

Description This command sets the mid level of line B.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:VOLTage:C:HIGH[?]

Syntax :PLUGin:CPHYplugin:HSMode:VOLTage:C:HIGH 'MIPI C-PHY Editor 1',4.00000000000000E-01

:PLUGin:CPHYplugin:HSMode:VOLTage:C:HIGH? 'MIPI C-PHY Editor 1'

Description This command sets the high level of line C.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:VOLTage:C:LOW[?]

Syntax :PLUGin:CPHYplugin:HSMode:VOLTage:C:LOW 'MIPI C-PHY Editor 1',2.00000000000000E-01

:PLUGin:CPHYplugin:HSMode:VOLTage:C:LOW? 'MIPI C-PHY Editor 1',

Description This command sets the low level of line C.

This query returns the present setting.

:PLUGin:CPHYplugin:HSMode:VOLTage:C:MID[?]

Syntax :PLUGin:CPHYplugin:HSMode:VOLTage:C:MID 'MIPI C-PHY Editor 1',3.00000000000000E-01

:PLUGin:CPHYplugin:HSMode:VOLTage:C:MID? 'MIPI C-PHY Editor 1'

Description This command sets the mid level of line C.

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:HSFTime[?]

Syntax :PLUGin:CPHYplugin:IMPairments:HSFTime 'MIPI C-PHY Editor 1',1.00000000000000E-10

:PLUGin:CPHYplugin:IMPairments:HSFTime? 'MIPI C-PHY Editor 1'

Description This command sets the fall time in high speed mode. Foe integral dividers of sampling rate to symbol rate it is possible to set the transition time to 0 for getting a maximally clean signal. For non-integral dividers the transition times are needed to hide sampling artifacts. This is especially true when jitter is added to the signal. As the jitter will shift the phase of the signal in an arbitrary step size it is necessary for the jittered waveform to have transition times containing at least 2 samples. If this AWG restriction is not met the resulting output signal will contain quantized jitter in the granularity of a sample period.

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:HSRTime[?]

Syntax :PLUGin:CPHYplugin:IMPairments:HSRTime 'MIPI C-PHY Editor 1',1.00000000000000E-10

:PLUGin:CPHYplugin:IMPairments:HSRTime? 'MIPI C-PHY Editor 1'

Description This command sets the rise time in the high speed mode. For integral dividers of sampling rate to symbol rate it is possible it is possible to set the transition times to 0 for getting a maximally clean signal. For non-integral dividers the transition times are needed to hide sampling artifacts. This is especially true when jitter is added to the signal. As the jitter will shift the phase of the signal in an arbitrary step size it is necessary for the jittered waveform to have transition time containing at least 2 samples. If the AWG restriction is not met the resulting output signal will contain quantized jitter in the granularity of a sample period.

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:JITTer:BUJ:RMS[?]

Syntax :PLUG:CPHY:IMP:JIT:BUJ:RMS 'MIPI C-PHY Editor 1',1.00000000000000E-09

:plug:CPHY:IMP:JIT:BUJ:RMS? 'MIPI C-PHY Editor 1'

Description This command sets the bounded uncorrelated jitter (BUJ, Random Jitter Simulation) applied to the lane in RMS.

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:JITTer:SINusodial:AMPlitude[?]

Syntax :PLUG:CPHY:IMP:JIT:SIN:AMP 'MIPI C-PHY Editor 1',2.00000000000000E-09

:PLUG:CPHY:IMP:JIT:SIN:AMP? 'MIPI C-PHY Editor 1'

Description This command sets the amount of sinusoidal amplitude peak-to-peak applied to the lane.

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:JITTer:SINusodial:FREQuency[?]

Syntax :PLUG:CPHY:IMP:JIT:SIN:FREQ 'MIPI C-PHY Editor 1',2.00000000000000E+00

:PLUG:CPHY:IMP:JIT:SIN:FREQ? 'MIPI C-PHY Editor 1'

Description This command sets the amount of sinusoidal jitter applied to the lane.

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:SKEW:A[?]

Syntax :PLUGin:CPHYplugin:IMPairments:SKEW:A 'MIPI C-PHY Editor 1',1.00000000000000E-09

:PLUGin:CPHYplugin:IMPairments:SKEW:A? 'MIPI C-PHY Editor 1'

Description This command sets the skew of line A.

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:SKEW:B[?]

Syntax :PLUGin:CPHYplugin:IMPairments:SKEW:B 'MIPI C-PHY Editor 1',1.00000000000000E-09

:PLUGin:CPHYplugin:IMPairments:SKEW:B? 'MIPI C-PHY Editor 1'

Description This command sets the skew of line B.

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:SKEW:C[?]

Syntax :PLUGin:CPHYplugin:IMPairments:SKEW:C 'MIPI C-PHY Editor 1',1.00000000000000E-09

:PLUGin:CPHYplugin:IMPairments:SKEW:C? 'MIPI C-PHY Editor 1'

Description This command sets the skew of line C.

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:HISPeed:DCD:A/B/C[?]

Syntax :PLUGin:CPHYplugin:IMPairments:HISPeed:DCD:A/B/C 'MIPI C-PHY Editor 1',0.2

:PLUGin:CPHYplugin:IMPairments:HISPeed:DCD:A/B/C?

Description This command sets the DCD value for High Speed symbols on line A/B/C, respectively. Positive DCD value shifts the rising and falling edges outwards, increasing the duration of levels. Negative DCD value shifts the rising and falling edges inwards, decreasing the duration of levels. DCD changes effects only on the respective line (A/B/C).

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:LOWPower:PULSe:A/B/C[?]

Syntax :PLUGin:CPHYplugin:IMPairments:LOWPower:PULSe:A/B/C 'MIPI C-PHY Editor 1',40e-9

:PLUGin:CPHYplugin:IMPairments:LOWPower:PULSe:A/B/C?

Description This command sets the duration of high levels by desired pulse value for Low Power symbols and adjust the difference of 2*UI and pluse width duration in low levels on line A/B/C, respectively.

This query returns the present setting.

:PLUGin:CPHYplugin:IMPairments:ENABle:PULSe[?]

Syntax :PLUGin:CPHYplugin:IMPairments:ENABle:PULSe 'MIPI C-PHY Editor 1',1|0

:PLUGin:CPHYplugin:IMPairments:LOWPower:PULSe?

Description This command enables or disables the pulse width change.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:SYMRate[?]

Syntax :PLUGin:CPHYplugin:LPMode:SYMRate 'MIPI C-PHY Editor 1',1.00000000000000E+06

:PLUGin:CPHYplugin:LPMode:SYMRate? 'MIPI C-PHY Editor 1'

Description This command sets the symbol rate of low power transmission.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:TERMination:RESistor[?]

Syntax :PLUGin:CPHYplugin:LPMode:TERMination:RESistor 'MIPI C-PHY Editor 1',3.00000000000000E+01

:PLUGin:CPHYplugin:LPMode:TERMination:RESistor? 'MIPI C-PHY Editor 1'

Description This command sets the resistance of the termination resistors in low power mode.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:TERMination:VOLTage[?]

Syntax :PLUGin:CPHYplugin:LPMode:TERMination:VOLTage 'MIPI C-PHY Editor 1',2.00000000000000E-01

:PLUGin:CPHYplugin:LPMode:TERMination:VOLTage? 'MIPI C-PHY Editor 1'

Description This command sets the termination voltage for low power mode. The receiver input is transmitted against this voltage.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:VOLTage:HIGH[?]

Syntax :PLUGin:CPHYplugin:LPMode:VOLTage:HIGH 'MIPI C-PHY Editor 1', 0.8

:PLUGin:CPHYplugin:LPMode:VOLTage:HIGH? 'MIPI C-PHY Editor 1'

Description This command sets the high level of low power mode.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:VOLTage:LOW[?]

Syntax :PLUGin:CPHYplugin:LPMode:VOLTage:LOW 'MIPI C-PHY Editor 1', 0.1

:PLUGin:CPHYplugin:LPMode:VOLTage:LOW? 'MIPI C-PHY Editor 1'

Description This command sets the low level of low power mode.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:VOLTage:A:HIGH[?]

Syntax :PLUGin:CPHYplugin:LPMode:VOLTage:A:HIGH 'MIPI C-PHY Editor 1', 0.8

:PLUGin:CPHYplugin:LPMode:VOLTage:A:HIGH? 'MIPI C-PHY Editor 1'

Description This command sets the high level of low power mode on line A.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:VOLTage:A:LOW[?]

Syntax :PLUGin:CPHYplugin:LPMode:VOLTage:A:LOW 'MIPI C-PHY Editor 1', 0.3

:PLUGin:CPHYplugin:LPMode:VOLTage:A:LOW? 'MIPI C-PHY Editor 1'

Description This command sets the low level of low power mode on line A.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:VOLTage:B:HIGH[?]

Syntax :PLUGin:CPHYplugin:LPMode:VOLTage:B:HIGH 'MIPI C-PHY Editor 1', 0.7

:PLUGin:CPHYplugin:LPMode:VOLTage:B:HIGH? 'MIPI C-PHY Editor 1'

Description This command sets the high level of low power mode on line B.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:VOLTage:B:LOW[?]

Syntax :PLUGin:CPHYplugin:LPMode:VOLTage:B:LOW 'MIPI C-PHY Editor 1', 0.2

:PLUGin:CPHYplugin:LPMode:VOLTage:B:LOW? 'MIPI C-PHY Editor 1'

Description This command sets the low level of low power mode on line B.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:VOLTage:C:HIGH[?]

Syntax :PLUGin:CPHYplugin:LPMode:VOLTage:C:HIGH 'MIPI C-PHY Editor 1', 0.6

:PLUGin:CPHYplugin:LPMode:VOLTage:C:HIGH? 'MIPI C-PHY Editor 1'

Description This command sets the high level of low power mode on line C.

This query returns the present setting.

:PLUGin:CPHYplugin:LPMode:VOLTage:C:LOW[?]

Syntax :PLUGin:CPHYplugin:LPMode:VOLTage:C:LOW 'MIPI C-PHY Editor 1', 0

:PLUGin:CPHYplugin:LPMode:VOLTage:C:LOW? 'MIPI C-PHY Editor 1'

Description This command sets the low level of low power mode on line C.

This query returns the present setting.

:PLUGin:CPHYplugin:NEW

Syntax :PLUGin:CPHYplugin:NEW 'MIPI C-PHY Editor 1'

Description This command is used to create the new C-PHY Plugin.

:PLUGin:CPHY:PATTern:INIT:PATH[?]

Syntax :PLUGin:CPHY:PATTern:INIT:PATH 'MIPI C-PHY Editor 1','InitPatternPath.ptrn'

:PLUGin:CPHY:PATTern:INIT:PATH? 'MIPI C-PHY Editor 1'

Description This command sets the file path for a file containing a C-PHY pattern. The path is relative in the user documentation folder. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window. This pattern is played one time after starting the AWG.

This query returns the present setting.

:PLUGin:CPHY:PATTern:LOOP:TYPE[?]

Syntax :PLUGin:CPHY:PATTern:LOOP:TYPE 'MIPI C-PHY Editor 1',P9 | P11 |P18

:PLUGin:CPHY:PATTern:LOOP:TYPE? 'MIPI C-PHY Editor 1'

Description This command selects the type of loop pattern. You can either select the user defines pattern file or PRBS generated.

This query returns the present setting.

:PLUGin:CPHY:PATTern:LOOP:PATH[?]

Syntax :PLUGin:CPHY:PATTern:LOOP:PATH 'MIPI C-PHY Editor 1','TestLoopPath'

:PLUGin:CPHY:PATTern:LOOP:PATH? 'MIPI C-PHY Editor 1'

Description This command sets the file path for a file containing a C-PHY pattern. The path is relative in the user documentation folder. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window. This pattern is looped continuously after the initial pattern.

This query returns the present setting.

:PLUGin:CPHY:PATTern:LOOP:SEED[?]

Syntax :PLUGin:CPHY:PATTern:LOOP:SEED 'MIPI C-PHY Editor 1','0x1523'

:PLUGin:CPHY:PATTern:LOOP:SEED? 'MIPI C-PHY Editor 1'

Description This command sets the PRBS seed which is either prefixed with "0b" in binary format, "0x" in hex format, or no prefix is present in integer format. Depending on the selected PRBS, a specific number of bits is used for filling the seed register. Surplus bits are ignored. For PRBS 9 and 11, the seed register is 16 bit wide for PRBS 18 it is 18 bit wide. For all three PRBS the tap where the actual PRBS data is taken from is at bit position 16. This leads to PRBS 9 having 7 bits definable by the seed which are only being generated once while afterwards the PRBS loops. For PRBS 11 this is true for the first 5 bits while for PRBS 18 the tap is still at position 16 but the PRBS data will repeat immediately.

This query returns the present setting.

:PLUGin:CPHYplugin:PROTocol:TRANsition:AUTO[?]

Syntax :PLUGin:CPHYplugin:PROTocol:TRANsition:AUTO 'MIPI C-PHY Editor 1', ON | OFF | 1 | 0

:PLUGin:CPHYplugin:PROTocol:TRANsition:AUTO? 'MIPI C-PHY Editor 1'

Description This command enables the toggle switch to automatically generate LP -> HS and HS –> LP transition. In case there is no start wire state for the HS pattern defined in the pattern file a default wire state will be used. The initial wire state of the preamble pattern is either a start wire state defined in the preamble pattern or in the HS burst pattern definition causing the LP-> HS transition to be generated. In case neither of these two pattern define an initial wire state a default state is used.

This query returns the present state.

:PLUGin:CPHYplugin:PROTocol:REQuest:DURation[?]

Syntax :PLUGin:CPHY:PROT:REQ:DUR 'MIPI C-PHY Editor 1',20e-6

:PLUGin:CPHY:PROT:REQ:DUR? 'MIPI C-PHY Editor 1'

Description This command sets the length of TX-HS-Request state.

This query returns the present setting.

:PLUGin:CPHYplugin:PROTocol:PREPare:DURation[?]

Syntax :PLUGin:CPHY:PROT:PREP:DUR 'MIPI C-PHY Editor 1',20e-6

:PLUGin:CPHY:PROT:PREP:DUR? 'MIPI C-PHY Editor 1'

Description This command sets the length of TX-HS-Prepare state.

This query returns the present setting.

:PLUGin:CPHYplugin:PROTocol:PREAmble:BEGin:PATTern[?]

Syntax :PLUGin:CPHY:PROT:PREA:BEG:PATT 'MIPI C-PHY Editor 1','PreambleBegin'

:PLUGin:CPHY:PROT:PREA:BEG:PATT? 'MIPI C-PHY Editor 1'

Description This command sets the TX-HS-Preamble pattern in high-speed C-PHY pattern format.

This query returns the present setting.

:PLUGin:CPHYplugin:PROTocol:PREAmble:PROGsequence:PATTern[?]

Syntax :PLUGin:CPHY:PROT:PREA:PROG:PATT 'MIPI C-PHY Editor 1','ProgramSequence'

:PLUGin:CPHY:PROT:PREA:PROG:PATT? 'MIPI C-PHY Editor 1'

Description This command sets the optional TX-HS-Prog-Seq pattern n high-speed C-PHY pattern format.

This query returns the present setting.

:PLUGin:CPHYplugin:PROTocol:PREAmble:END:PATTern[?]

Syntax :PLUGin:CPHY:PROT:PREA:END:PATT 'MIPI C-PHY Editor 1','PreambleEnd'

:PLUGin:CPHY:PROT:PREA:END:PATT? 'MIPI C-PHY Editor 1'

Description This command sets the TX-HS-Pre-End pattern in high-speed C-PHY pattern format. By default, this pattern contains a total of 7 symbols of type 3.

This query returns the present setting.

:PLUGin:CPHYplugin:PROTocol:SYNCword:PATTern[?]

Syntax :PLUGin:CPHY:PROT:SYNC:PATT 'MIPI C-PHY Editor 1','SyncWord'

:PLUGin:CPHY:PROT:SYNC:PATT? 'MIPI C-PHY Editor 1'

Description This command sets the TX-HS-Sync-Word pattern which is send immediately before staring high-speed transmission. The default pattern is 3444443.

This query returns the present setting.

:PLUGin:CPHYplugin:PROTocol:POST:PATTern[?]

Syntax :PLUGin:CPHY:PROT:POST:PATT 'MIPI C-PHY Editor 1','PostPattern'

:PLUGin:CPHY:PROT:POST:PATT? 'MIPI C-PHY Editor 1'

Description This command sets the TX-HS-Post pattern which is send immediately after staring high-speed transmission.

This query returns the present setting.

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

Syntax :PLUGin:CPHY:PROT:EXIT:DUR 'MIPI C-PHY Editor 1',20e-6

:PLUGin:CPHY:PROT:EXIT:DUR? 'MIPI C-PHY Editor 1'

Description This command sets the length of LP-111 state following a high-speed burst.

This query returns the present setting.

:PLUGin:CPHYplugin:RESet[?]

Syntax :PLUGin:CPHYplugin:RESet 'MIPI C-PHY Editor 1'

Description This SCPI is used to reset the C_PHY Plugin.

This query returns the present setting.

:PLUGin:CPHYplugin:LINK:SAMPlerate[?]

Syntax :PLUGin:CPHY:LINK:SAMP 'MIPI C-PHY Editor 1',1.00000000000000E+08

:PLUGin:CPHY:LINK:SAMP? 'MIPI C-PHY Editor 1'

Description This command sets the AWG sample rate.

This query returns the present setting.

:PLUGin:CPHYplugin:LINK:AWGA[?]

Syntax :PLUGin:CPHYplugin:LINK:AWGA 'MIPI C-PHY Editor 1','AWGA Visa Address'

:PLUGin:CPHYplugin:LINK:AWGA? 'MIPI C-PHY Editor 1'

Description This command specifies the Visa connection string so as to establish a connection with AWG 1.

This query returns the present setting.

:PLUGin:CPHYplugin:LINK:AWGB[?]

Syntax :PLUGin:CPHYplugin:LINK:AWGB 'MIPI C-PHY Editor 1','AWGA Visa Address'

:PLUGin:CPHYplugin:LINK:AWGB? 'MIPI C-PHY Editor 1'

Description This command specifies the visa connection string so as to establish a connection with AWG 2.

This query returns the present setting.

:PLUGin:CPHYplugin:LINK:OSCilloscpe[?]

Syntax :PLUGin:CPHYplugin:LINK:OSCilloscpe 'MIPI C-PHY Editor 1','Oscilloscope Visa Address'

:PLUGin:CPHYplugin:LINK:OSCilloscpe? 'MIPI C-PHY Editor 1'

Description This command specifies the visa connection string so as to establish a connection with oscilloscope.

This query returns the present setting.

:PLUGin:CPHYplugin:LINK:STAMode[?]

Syntax :PLUGin:CPHYplugin:LINK:STAMode 'MIPI C-PHY Editor 1', IMMediate | TRIGgered

:PLUGin:CPHYplugin:LINK:STAMode? 'MIPI C-PHY Editor 1'

Description This command sets the startup mode whether signal generation starts immediately or if it should wait on a trigger event. Triggered selection will fully prepare AWGs for immediate startup.

This query returns the present setting.

:PLUGin:DPHYplugin:LINK:TRIGger[:EXECute]

Syntax :PLUGin:DPHYplugin:LINK:TRIGger:EXECute 'MIPI C-PHY Editor 1'

Description This command starts the trigger event. It breaks from LP Stop state into waveform playback. It starts the waveform generation after AWGs have been programmed and initialization voltage level at LP high is active.

:PLUGin:CPHYplugin:LINK:CALibration:PORT[?]

Syntax :PLUGin:CPHYplugin:LINK:CALibration:PORT 'MIPI C-PHY Editor 1', NORMal|COMPlement

:PLUGin:CPHYplugin:LINK:CALibration:PORT? 'MIPI C-PHY Editor 1'

Description This command allow you to select between Normal and Complement port.

This query returns the present setting.

:PLUGin:CPHYplugin:LINK:CALibration:DESKew:A?

Syntax :PLUGin:CPHYplugin:LINK:CALibration:DESKew:A? 'MIPI C-PHY Editor 1'

Description This query returns the measured output of skew during calibration on the channel A of AWG.

:PLUGin:CPHYplugin:LINK:CALibration:DESKew:B?

Syntax :PLUGin:CPHYplugin:LINK:CALibration:DESKew:B? 'MIPI C-PHY Editor 1'

Description This query returns the measured output of skew during calibration on the channel B of AWG.

:PLUGin:CPHYplugin:LINK:CALibration:DESKew:C?

Syntax :PLUGin:CPHYplugin:LINK:CALibration:DESKew:C? 'MIPI C-PHY Editor 1'

Description This query returns the measured output of skew during calibration on the channel C of AWG.

:PLUGin:CPHYplugin:LINK:CALibration:DESKew:D?

Syntax :PLUGin:CPHYplugin:LINK:CALibration:DESKew:D? 'MIPI C-PHY Editor 1'

Description This query returns the measured output of skew during calibration on the channel D of AWG.

:PLUGin:CPHYplugin:RUN[:STATus]?

Syntax :PLUGin:CPHYplugin:RUN:STATus 'MIPI C-PHY Editor 1'

Description This query returns the running status of the frame generation.

:PLUGin:CPHYplugin:RUN:PROGress?

Syntax :PLUGin:CPHYplugin:RUN:PROGress 'MIPI C-PHY Editor 1'

Description This query returns the progress of the current frame generation.

:PLUGin:CPHYplugin:STARt

Syntax :PLUGin:CPHYplugin:STARt 'MIPI C-PHY Editor 1'

Description This command starts the frame generation.

:PLUGin:CPHYplugin:STOP

Syntax :PLUGin:CPHYplugin:STOP 'MIPI C-PHY Editor 1'

Description This command stops the frame generation.

:PLUGin:CPHYplugin:TRIGger:STARt[?]

Syntax :PLUGin:CPHYplugin:TRIGger:STARt 'MIPI C-PHY Editor 1',LPLOw|LPHIgh

:PLUGin:CPHYplugin:TRIGger:STARt? 'MIPI C-PHY Editor 1'

Description This command sets the static LP STOP signal. To test the minimum initialization period (T_{INIT}) needed by DUT to recognize the valid test sequence, it is done by sending the LP111 sequence and slowly increasing the length of LP111 state. When DUT starts recognizing the valid sequences of LP and HS then record the duration of LP111. In order to achieve this test, DUT must be started in state other than LP111, and to support this DUT can now be started with LP000 state in triggered mode. So if DUT is started in triggered mode then DUT will only see LP000 state i.e. low on all three lines, without actually running any patterns applied by the user. When triggered mode is changed to Immediate mode then user patterns are applied and DUT can start seeing the valid sequence depending on the length of LP111.

This query returns the present setting.

:PLUGin:CPHYplugin:TRIGger:VOLTage:HIGH[?]

Syntax :PLUGin:CPHYplugin:TRIGger:VOLTage:HIGH 'MIPI C-PHY Editor 1',5.00000000000000E-01

:PLUGin:CPHYplugin:TRIGger:VOLTage:HIGH? 'MIPI C-PHY Editor 1'

Description This command sets the high level of trigger.

This query returns the present setting.

:PLUGin:CPHYplugin:TRIGger:VOLTage:LOW[?]

Syntax :PLUGin:CPHYplugin:TRIGger:VOLTage:LOW 'MIPI C-PHY Editor 1',1.00000000000000E-01

:PLUGin:CPHYplugin:TRIGger:VOLTage:LOW? 'MIPI C-PHY Editor 1'

Description This command sets the low level of trigger.

This query returns the present setting.

:PLUGin:CPHYplugin:ESPike:MODE[?]

Syntax :PLUGin:CPHYplugin:ESPike:MODE 'MIPI C-PHY Editor 1', HLEVels | LLEVels| BLEVels|OFF

:PLUGin:CPHYplugin:ESPike:MODE? 'MIPI C-PHY Editor 1'

Description This command turns the eSpike insertion on or off. In the enabled state, it allows you to select whether the eSpike should occur inside high level only, low levels only or in both high and low levels.

This query returns the present setting.

:PLUGin:CPHYplugin:ESPike:AREA[?]

Syntax :PLUGin:CPHYplugin:ESPike:AREA 'MIPI C-PHY Editor 1', 1.25e-10

:PLUGin:CPHYplugin:ESPike:AREA? 'MIPI C-PHY Editor 1'

Description This command defines the area within which the eSpike will be generated. This command sets the same area on all 3 lines.

This query returns the present setting.

The following commands defines the area within which the eSpike will be generated on each individual line A, B & C.

:PLUGin:CPHYplugin:ESPike:AREA:A[?] 'MIPI C-PHY Editor 1', 1.25e-10

:PLUGin:CPHYplugin:ESPike:AREA:B[?] 'MIPI C-PHY Editor 1', 1.25e-10

:PLUGin:CPHYplugin:ESPike:AREA:C[?] 'MIPI C-PHY Editor 1', 1.25e-10

:PLUGin:CPHYplugin:ESPike:LHIGhinvol[?]

Syntax :PLUGin:CPHYplugin:ESPike:LHIGhinvol 'MIPI C-PHY Editor 1', 2.00000000000000E-01

:PLUGin:CPHYplugin:ESPike:LHIGhinvol? 'MIPI C-PHY Editor 1'

Description This command sets the lower receiver detection threshold for a logical 1 which is the voltage level at which the eSpike area inside this logical 1 is to be calculated from. This command sets the same value on all 3 lines.

This query returns the present setting.

The following commands sets the lower receiver detection threshold for a logical 1 which is the voltage level at which the eSpike area inside this logical 1 is to be calculated from on each individual line A, B & C.

:PLUGin:CPHYplugin:ESPike:LHIGhinvol:A[?] 'MIPI C-PHY Editor 1', 7.4e-1

:PLUGin:CPHYplugin:ESPike:LHIGhinvol:B[?] 'MIPI C-PHY Editor 1', 7.4e-1

:PLUGin:CPHYplugin:ESPike:LHIGhinvol:C[?] 'MIPI C-PHY Editor 1', 7.4e-1

:PLUGin:CPHYplugin:ESPike:LLOWintvol[?]

Syntax :PLUGin:CPHYplugin:ESPike:LLOWintvol 'MIPI C-PHY Editor 1', 3.00000000000000E-01

:PLUGin:CPHYplugin:ESPike:LLOWintvol? 'MIPI C-PHY Editor 1'

Description This command sets the upper receiver detection threshold for a logical 0 which is the voltage level at which the eSpike area inside this logical 0 is to be calculated from. This command sets the same value on all 3 lines.

This query returns the present setting.

The following commands sets the upper receiver detection threshold for a logical 0 which is the voltage level at which the eSpike area inside this logical 0 is to be calculated from on each individual line A, B & C.

:PLUGin:CPHYplugin:ESPike:LLOWintvol:A[?] 'MIPI C-PHY Editor 1', 1e-1

:PLUGin:CPHYplugin:ESPike:LLOWintvol:B[?] 'MIPI C-PHY Editor 1', 14e-1

:PLUGin:CPHYplugin:ESPike:LLOWintvol:C[?] 'MIPI C-PHY Editor 1', 1e-1

:PLUGin:CPHYplugin:ESPike:MODE:A[?

Syntax :PLUGin:CPHYplugin:ESPike:MODE:A 'MIPI C-PHY Editor 1', LLEVels|HLEVels|BLEVels|OFF

:PLUGin:CPHYplugin:ESPike:MODE:A 'MIPI C-PHY Editor 1'

Description This command sets the eSpike mode for line A.

This query returns the present setting.

:PLUGin:CPHYplugin:ESPike:MODE:B[?

Syntax :PLUGin:CPHYplugin:ESPike:MODE:B 'MIPI C-PHY Editor 1', LLEVels|HLEVels|BLEVels|OFF

:PLUGin:CPHYplugin:ESPike:MODE:B 'MIPI C-PHY Editor 1'

Description This command sets the eSpike mode for line B.

This query returns the present setting.

:PLUGin:CPHYplugin:ESPike:MODE:C[?

Syntax :PLUGin:CPHYplugin:ESPike:MODE:C 'MIPI C-PHY Editor 1', LLEVels|HLEVels|BLEVels|OFF

:PLUGin:CPHYplugin:ESPike:MODE:C 'MIPI C-PHY Editor 1'

Description This command sets the eSpike mode for line C.

This query returns the present setting.

:PLUGin:CPHYplugin:VOLTage:OFFSet[?]

Syntax :PLUGin:CPHYplugin:VOLTage:OFFSet 'MIPI C-PHY Editor 1', 3.00000000000000E-01

:PLUGin:CPHYplugin:VOLTage:OFFSet? 'MIPI C-PHY Editor 1'

Description This command sets the internal offset for the amplifiers of the three C-PHY data lines.

This query returns the present setting.

:PLUGin:CPHYplugin:VOLTage:AMPlitude[?]

Syntax :PLUGin:CPHYplugin:VOLTage:AMPlitude 'MIPI C-PHY Editor 1', 1.00000000000000E+00

:PLUGin:CPHYplugin:VOLTage:AMPlitude? 'MIPI C-PHY Editor 1'

Description This command sets the internal amplitude for the amplifiers of the three C-PHY data lines.

This query returns the present setting.

:PLUGin:CPHYplugin:VOLTage:TERMination[?]

Syntax :PLUGin:CPHYplugin:VOLTage:TERMination 'MIPI C-PHY Editor 1', 3.00000000000000E-01, R50

:PLUGin:CPHYplugin:VOLTage:TERMination? 'MIPI C-PHY Editor 1'

Description This command sets value of termination voltage against 500 Ohms or 1 MOhm.

This query returns the present setting.

:PLUGin:CPHYplugin:LINK:CALibration[:EXECute]

Syntax :PLUGin:CPHYplugin:LINK:CALibration:EXECute 'MIPI C-PHY Editor 1'

Description This command calibrates the selected AWG properties like skew and amplitude.

:PLUGin:CPHYplugin:ISI:ENABle[?]

Syntax :PLUGin:CPHYplugin:ISI:ENAB 'MIPI C-PHY Editor 1' ON | OFF | 1 | 0

:PLUGin:CPHYplugin:ISI:ENAB? 'MIPI C-PHY Editor 1

Description This command enables/disables the ISI feature.

This query returns the present setting.

:PLUGin:CPHYplugin:ISI:PATH:A[?]

Syntax :PLUGin:CPHYplugin:ISI:PATH:A 'MIPI C-PHY Editor 1' 'SParameterFile.s2p'

:PLUGin:CPHYplugin:ISI:PATH:A? 'MIPI C-PHY Editor 1'

Description This command sets the file path for a file containing a S-Parameter information for line A. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window.

This query returns the present setting.

:PLUGin:CPHYplugin:ISI:PATH:B[?]

Syntax :PLUGin:CPHYplugin:ISI:PATH:B 'MIPI C-PHY Editor 1' 'SParameterFile.s2p'

:PLUGin:CPHYplugin:ISI:PATH:B? 'MIPI C-PHY Editor 1'

Description This command sets the file path for a file containing a S-Parameter information for line B. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window.

This query returns the present setting.

:PLUGin:CPHYplugin:ISI:PATH:C[?]

Syntax :PLUGin:CPHYplugin:ISI:PATH:C 'MIPI C-PHY Editor 1' 'SParameterFile.s2p'

:PLUGin:CPHYplugin:ISI:PATH:C? 'MIPI C-PHY Editor 1'

Description This command sets the file path for a file containing a S-Parameter information for line C. To access this folder, write “%USERPROFILE%\Documents” into a file explorer window.

This query returns the present setting.

:PLUGin:CPHYplugin:ISI:NUMPorts[?]

Syntax :PLUGin:CPHYplugin:ISI:NUMPorts 'MIPI C-PHY Editor 1' TWOPort | SIXPort

:PLUGin:CPHYplugin:ISI:NUMPorts? 'MIPI C-PHY Editor 1'

Description This command sets the no. of ports. Number of port defines number of input and output ports i.e. for 2 ports, there will be 2 input and 2 output ports.

This query returns the present setting.

:PLUGin:CPHYplugin:ISI:PATH[?]

Syntax :PLUGin:CPHYplugin:ISI:PATH 'MIPI C-PHY Editor 1', "SParameterFile.s6p"

:PLUGin:CPHYplugin:ISI:PATH? 'MIPI C-PHY Editor 1'

Description This command sets the file path for a file containing a 6 port (S6P) S-Parameter information for all lines.

This query returns the present setting.

:PLUGin:CPHYplugin:ISI:SCALe[?]

Syntax :PLUGin:CPHYplugin:ISI:SCALe 'MIPI C-PHY Editor 1', 1

:PLUGin:CPHYplugin:ISI:PATH? 'MIPI C-PHY Editor 1'

Description This command sets the scaling parameter value that increases or decreases the effect of S-parameter measurement freely.

The following command sets the same scaling parameter on line A,B and C:

:PLUGin:CPHYplugin:ISI:SCALe[?] 'MIPI C-PHY Editor 1', 1

However, the following commands sets the scaling parameter on line A,B and C, respectively:

:PLUGin:CPHYplugin:ISI:SCALe:A[?] 'MIPI C-PHY Editor 1', 1

:PLUGin:CPHYplugin:ISI:SCALe:B[?] 'MIPI C-PHY Editor 1', 2

:PLUGin:CPHYplugin:ISI:SCALe:C[?] 'MIPI C-PHY Editor 1', 4

This query returns the present setting.

:PLUGin:CPHYplugin:HYSTeresis:VOLTage:A/B/C[?]

Syntax :PLUGin:CPHYplugin:HYSTeresis:VOLTage:A/B/C 'MIPI C-PHY Editor 1', 0.020

:PLUGin:CPHYplugin:HYSTeresis:VOLTage:A/B/C? 'MIPI C-PHY Editor 1'

Description This command sets hysteresis or additive noise in LP signal on line A/B/C to check if DUT can still detect the valid LP sequence without any errors. The input hysteresis amplitude is deviation from the nominal LP 0/1 levels, which is half of its peak to peak value.

This query returns the present setting.

:PLUGin:CPHYplugin:LINK:RSEQuence

Syntax :PLUGin:CPHYplugin:LINK:RSEQuence 'MIPI C-PHY Editor 1'

Description This command restarts the waveform generation without pattern recalculation. Changing any parameter during run mode will result in a voltage glitch on init/loop pattern transition.

:PLUGin:CPHYplugin:LINK:SEQType[?]

Syntax :PLUGin:CPHYplugin:LINK:SEQType 'MIPI C-PHY Editor 1', VARParameters|HSEnd|LPEnd

:PLUGin:CPHYplugin:LINK:SEQType? 'MIPI C-PHY Editor 1'

Description This command selects the type of sequence statically before downloading any pattern in order to support different kinds of sequence scenarios. The available sequence types are Variable Parameters, High Speed End and Low Power End. For details, see Sequence on page 54.

This query returns the present setting.

Index

Edition 4.2, December 2015

www.keysight.com

KEYSIGHT
TECHNOLOGIES