



# MIPI D-PHY RX Controller Core User Guide

---

UG-CORE-MIPI-D-PHY-RX-v2.0

June 2025

[www.efinixinc.com](http://www.efinixinc.com)



# Contents

<b>Introduction.....</b>	<b>3</b>
<b>Features.....</b>	<b>3</b>
<b>Device Support.....</b>	<b>3</b>
<b>Resource Utilization and Performance.....</b>	<b>4</b>
Release Notes.....	4
<b>Functional Description.....</b>	<b>5</b>
Ports.....	5
<b>IP Manager.....</b>	<b>11</b>
<b>Customizing the MIPI D-PHY RX Controller.....</b>	<b>12</b>
<b>MIPI D-PHY RX Controller Example Design.....</b>	<b>14</b>
<b>MIPI D-PHY RX Controller Testbench.....</b>	<b>15</b>
<b>Revision History.....</b>	<b>15</b>

# Introduction

The MIPI D-PHY is a scalable physical layer for interfacing various components such as cameras and displays to host devices. The MIPI D-PHY RX Controller core allows you to control and configure the MIPI D-PHY RX interface. You use the MIPI D-PHY RX Controller core with the Titanium HSIO pins configured as MIPI RX lanes to build a custom MIPI interface.

Use the IP Manager to select IP, customize it, and generate files. The MIPI D-PHY RX Controller core has an interactive wizard to help you set parameters. The wizard also has options to create a testbench and/or example design targeting an Efinix® development board.

## Features

- 1, 2, 4, or 8 configurable data lanes
- Unidirectional (CSI) and bidirectional (DSI) data lane
- High-speed (HS) mode for data communication and low-power (LP) mode for data communication
- 100 MHz core clock frequency
- HS byte clock frequency from 10 MHz up to 187 MHz (from 80 Mbps up to 1,500 Mbps data rate)<sup>(1)</sup>
- Continuous HS byte clock and discontinuous HS byte clock
- Supports 8 bits HS data width
- Supports D-PHY-only, D-PHY with CSI, or D-PHY with DSI
- Supports PHY protocol interface (PPI)
- Supports end-of-transmission error, start-of-transmission sync error, control error, and LP escape error

## Device Support

*Table 1: MIPI D-PHY RX Controller Core Device Support*

FPGA Family	Supported Device
Trion	-
Titanium and Topaz	All

<sup>(1)</sup> The maximum data rate of IP depends on the devices. Refer to the respective device data sheet for more accurate information.

# Resource Utilization and Performance



**Note:** The resources and performance values provided are based on some of the supported FPGAs. These values are just guidance and can change depending on the device resource utilization, design congestion, and user design.

*Table 2: Titanium Resource Utilization and Performance*

FPGA	Data Lane	Logic and Adders	Flip-flops	Memory Blocks	DSP48 Blocks	f <sub>MAX</sub> (MHz) <sup>(2)</sup>		Efinity <sup>®</sup> Version <sup>(3)</sup>
						clk	clk_byte_HS	
Ti60 F225 C4	Unidirectional	1,053	567	0	0	401	471	2021.2
	Bidirectional	1,260	629	0	0	361	364	2021.2

## Release Notes

You can refer to the IP Core Release Notes for more information about the IP core changes. The IP Core Release Notes are available on the [Efinity Downloads](#) page under each Efinity software release version.



**Note:** You must be logged in to the Support Center to view the IP Core Release Notes.

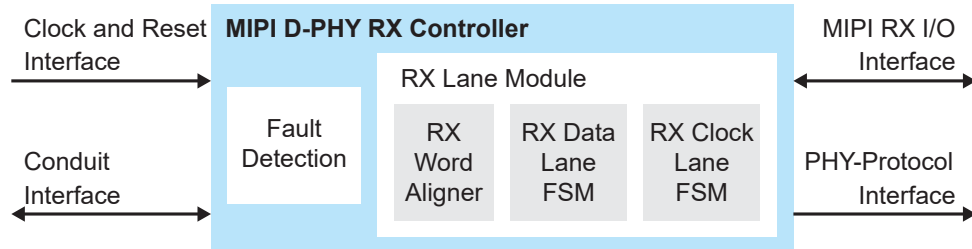
<sup>(2)</sup> Using default parameter settings.

<sup>(3)</sup> Using Verilog HDL.

# Functional Description

The MIPI D-PHY RX Controller consists of a fault detection block, HS data de-skew calibration block, word aligner block, data lane FSM block, and clock lane FSM block.

*Figure 1: MIPI D-PHY RX Controller System Block Diagram*



## Ports

*Table 3: Clock and Reset Interface*

Port	Direction	Description
clk	Input	IP core clock consumed by controller logics. 100 MHz.
reset_n	Input	IP core reset signal.
clk_byte_HS	Input	MIPI RX parallel clock. This is a HS mode transmission clock.
reset_byte_HS_n	Input	MIPI RX parallel clock reset signal.

Table 4: MIPI RX I/O Interface

Port	Direction	Description
Rx_LP_CLK_P	Input	LP mode RX clock single-ended P signal.
Rx_LP_CLK_N	Input	LP mode RX clock single-ended N signal.
Rx_HS_enable_C	Output	Signal to enable HS mode clock lane.
LVDS_termen_C	Output	Signal to enable HS mode clock lane termination .
Rx_LP_D_P [NUM_DATA_LANE-1:0]	Input	LP mode RX data single-ended P signal.
Rx_LP_D_N [NUM_DATA_LANE-1:0]	Input	LP mode RX data single-ended N signal.
Rx_HS_D_0 [7:0]	Input	HS mode differential lane 0 data bus.
Rx_HS_D_1 [7:0]	Input	HS mode differential lane 1 data bus.
Rx_HS_D_2 [7:0]	Input	HS mode differential lane 2 data bus.
Rx_HS_D_3 [7:0]	Input	HS mode differential lane 3 data bus.
Rx_HS_D_4 [7:0]	Input	HS mode differential lane 4 data bus.
Rx_HS_D_5 [7:0]	Input	HS mode differential lane 5 data bus.
Rx_HS_D_6 [7:0]	Input	HS mode differential lane 6 data bus.
Rx_HS_D_7 [7:0]	Input	HS mode differential lane 7 data bus.
Rx_HS_enable_D [NUM_DATA_LANE-1:0]	Output	Signal to enable HS mode data lane.
LVDS_termen_D [NUM_DATA_LANE-1:0]	Output	Signal to enable HS mode data lane termination.
fifo_rd_enable [NUM_DATA_LANE-1:0]	Output	Rx HS mode data lane FIFO read enable signal.
fifo_rd_empty [NUM_DATA_LANE-1:0]	Input	Rx HS mode data lane FIFO empty signal.
DLY_enable_D [NUM_DATA_LANE-1:0]	Output	Reserved port.
DLY_inc_D [NUM_DATA_LANE-1:0]	Output	Reserved port.
u_dly_enable_D [NUM_DATA_LANE-1:0]	Input	Reserved port. Tie the signal to zero.
u_dly_inc_D [NUM_DATA_LANE-1:0]	Input	Reserved port. Tie the signal to zero.
Tx_LP_D_P	Output	LP mode TX data single-ended P signal for bidirectional data lane.
Tx_LP_D_P_OE	Output	Output enable for LP mode TX data single-ended P signal for bidirectional data lane.
Tx_LP_D_N_OE	Output	Output enable for LP mode TX data single-ended N signal for bidirectional data lane.
Tx_LP_D_N	Output	LP mode TX data single-ended N signal for bidirectional data lane.

Table 5: PHY Protocol Interface

Port	Direction	Description
RxUlpsClkNot	Output	Receive ULPS on Clock Lane. This active-low signal is asserted to indicate that the clock lane module has entered the ULPS due to the detection of a request to enter the ULPS.
RxUlpsActiveClkNot	Output	ULPS (not) Active. This active-low signal is asserted to indicate that the lane is in ULPS.
RxUlpsEsc [NUM_DATA_LANE-1:0]	Output	Escape ULPS. This active-high signal is asserted to indicate that the lane module has entered the ULPS, due to the detection of a received ULPS command.
RxUlpsActiveNot [NUM_DATA_LANE-1:0]	Output	ULPS (not) Active. This active-low signal is asserted to indicate that the lane is in ULPS.
RxLPDTEsc [NUM_DATA_LANE-1:0]	Output	Escape LP Data Receive Mode. This active-high signal is asserted to indicate that the lane module is in LP data receive mode.
RxValidEsc [NUM_DATA_LANE-1:0]	Output	Escape LP Data Receive Mode. This active-high signal is asserted to indicate that the lane module is in LP data receive mode.
RxStopState [NUM_DATA_LANE-1:0]	Output	Lane in Stop State. This active-high signal indicates that the lane module is currently in stop state.
RxDataEsc_0 [7:0]	Output	Escape Mode Receive Data. This is the eight-bit escape mode LP data received by the lane module. The signal connected to RxDataEsc_0[0] was received first. Data is transferred on rising edges of clk.
RxDataEsc_1 [7:0]	Output	Escape Mode Receive Data. This is the eight-bit escape mode LP data received by the lane module. The signal connected to RxDataEsc_1[0] was received first. Data is transferred on rising edges of clk.
RxDataEsc_2 [7:0]	Output	Escape Mode Receive Data. This is the eight-bit escape mode LP data received by the lane module. The signal connected to RxDataEsc_2[0] was received first. Data is transferred on rising edges of clk.
RxDataEsc_3 [7:0]	Output	Escape Mode Receive Data. This is the eight-bit escape mode LP data received by the lane module. The signal connected to RxDataEsc_3[0] was received first. Data is transferred on rising edges of clk.
RxDataEsc_4 [7:0]	Output	Escape Mode Receive Data. This is the eight-bit escape mode LP data received by the lane module. The signal connected to RxDataEsc_4[0] was received first. Data is transferred on rising edges of clk.
RxDataEsc_5 [7:0]	Output	Escape Mode Receive Data. This is the eight-bit escape mode LP data received by the lane module. The signal connected to RxDataEsc_5[0] was received first. Data is transferred on rising edges of clk.
RxDataEsc_6 [7:0]	Output	Escape Mode Receive Data. This is the eight-bit escape mode LP data received by the lane module. The signal connected to RxDataEsc_6[0] was received first. Data is transferred on rising edges of clk.

Port	Direction	Description
RxDataEsc_7 [7:0]	Output	Escape Mode Receive Data. This is the eight-bit escape mode LP data received by the lane module. The signal connected to RxDataEsc_7[0] was received first. Data is transferred on rising edges of clk.
RxTriggerEsc [3:0]	Output	EscapeMode Receive Trigger. These active high signals indicate that a trigger event has been received. The asserted RxTriggerEsc signal remains active until a stop state is detected on the lane interconnect. Applicable to bidirectional mode only. RxTriggerEsc[0] corresponds to reset-trigger. RxTriggerEsc[1] corresponds to entry sequence for HS test mode trigger. RxTriggerEsc[2] corresponds to unknown-4 trigger. RxTriggerEsc[3] corresponds to unknown-5 trigger.
RxErrEsc [NUM_DATA_LANE-1:0]	Output	Escape Entry Error. If an unrecognized escape entry command is received in LP mode, this active-high signal is asserted and remains asserted until the next transaction starts, so that the protocol can properly process the error.
RxErrControl [NUM_DATA_LANE-1:0]	Output	Control Error. This active-high signal is asserted when an incorrect line state sequence is detected in LP and ALP modes. Once asserted, this signal remains asserted until the next transaction starts, so that the protocol can properly process the error.
RxErrSotSynchS [NUM_DATA_LANE-1:0]	Output	Start-of-Transmission Synchronization Error. If the HS SoT leader sequence is corrupted in a way that proper synchronization cannot be expected, this active-high signal is asserted for one cycle of RxWordClkHS. When ErrSotSynchS is asserted, RxSynchS, ErrSotHS, and RxValidHS is not asserted.
RxDataHS_0 [7:0]	Output	HS Receive Data. High-speed data received by the lane module.
RxDataHS_1 [7:0]	Output	HS Receive Data. High-speed data received by the lane module.
RxDataHS_2 [7:0]	Output	HS Receive Data. High-speed data received by the lane module.
RxDataHS_3 [7:0]	Output	HS Receive Data. High-speed data received by the lane module.
RxDataHS_4 [7:0]	Output	HS Receive Data. High-speed data received by the lane module.
RxDataHS_5 [7:0]	Output	HS Receive Data. High-speed data received by the lane module.
RxDataHS_6 [7:0]	Output	HS Receive Data. High-speed data received by the lane module.
RxDataHS_7 [7:0]	Output	HS Receive Data. High-speed data received by the lane module.
RxValidHS [NUM_DATA_LANE-1:0]	Output	HS Receive Data Valid. This active-high signal indicates that the lane module is driving data to the protocol layer on the RxDataHS output.

Port	Direction	Description
RxActiveHS [NUM_DATA_LANE-1:0]	Output	HS Reception Active. This active-high signal indicates that the lane module is actively receiving a HS transmission from the lane interconnect.
RxSyncHS [NUM_DATA_LANE-1:0]	Output	Receiver Synchronization Observed. This active-high signal indicates that the lane module has seen an appropriate synchronization event.
RxSkewCalHS [NUM_DATA_LANE-1:0]	Output	HS Receive Skew Calibration. This optional active-high signal indicates that the high speed deskew burst is being received.
TxRequestEsc	Input	Escape Mode Transmit Request. This active-high signal is used to request escape sequences.
TxTriggerEsc [3:0]	Input	Escape Mode Receive Trigger. These active high signals indicate that a trigger event has been received. The asserted TxTriggerEsc signal remains active until a stop state is detected on the lane interconnect. Applicable to bidirectional mode only. TxTriggerEsc[0] corresponds to reset-trigger. TxTriggerEsc[1] corresponds to entry sequence for HS test mode trigger. TxTriggerEsc[2] corresponds to unknown-4 trigger. TxTriggerEsc[3] corresponds to unknown-5 trigger.
TxUlpsEsc	Input	Escape Mode Transmit ULPS. This active-high signal is asserted with TxRequestEsc to cause the lane module to enter the ULPS. Applicable to bidirectional mode only.
TxUlpsExit	Input	Transmit ULPS Exit Sequence. This active-high signal is asserted when ULPS is active and the protocol is ready to leave ULPS. Applicable to bidirectional mode only.
TxLpdtEsc	Input	Escape Mode Transmit LP Data. This active-high signal is asserted with TxRequestEsc to cause the lane module to enter LP data transmission mode. Applicable to bidirectional mode only.
TxDataEsc [7:0]	Input	Escape Mode Transmit Data. This is the eight bit Escape Mode data to be transmitted in LP data transmission mode. The signal connected to TxDataEsc[0] is transmitted first. Data is captured on rising edges of clk. Applicable to bidirectional mode only.
TxValidEsc	Input	Escape Mode Transmit Data Valid. This active-high signal indicates that the protocol is driving valid data on TxDataEsc to be transmitted. Applicable to bidirectional mode only.
TxReadyEsc	Output	Escape Mode Transmit Ready. This active-high signal indicates that TxDataEsc is accepted by the lane module to be serially transmitted. TxReadyEsc is valid on rising edges of TxClkEsc. Applicable to bidirectional mode only.

Port	Direction	Description
TxStopState	Output	Lane in Stop State. This active-high signal indicates that the lane module is in stop state. Applicable to bidirectional mode only.
TxUlpsActiveNot	Output	ULPS (not) Active. This active-low signal is asserted to indicate that the lane is in ULPS. Applicable to bidirectional mode only.
Turnaround_timeout	Output	Indicates that there is no acknowledgement from the RX D-PHY for the turnaround request and TX D-PHY ends the turnaround request. Applicable to bidirectional mode only.
TurnRequest	Input	Turnaround Request. This active high signal is used to indicate that the protocol desires to initiate a bidirectional data lane turnaround, to allow the other side to begin transmissions. TurnRequest is valid on rising edge of clk. TurnRequest is only meaningful for a bidirectional data lane module that is currently the transmitter (Direction=0). If the bidirectional data lane module is in receive mode (Direction=1), this signal is ignored. A low-to-high transition on TurnRequest can only happen when Stopstate is asserted. Applicable to bidirectional mode only.
TurnRequest_done	Output	Indicates that the RX D-PHY acknowledges the bus turnaround or timeout. If this signal is high together with turnaround timeout, it indicates that there is no acknowledgement from the RX on the turnaround request. Applicable to bidirectional mode only.

# IP Manager

The Efinity® IP Manager is an interactive wizard that helps you customize and generate Efinity® IP cores. The IP Manager performs validation checks on the parameters you set to ensure that your selections are valid. When you generate the IP core, you can optionally generate an example design targeting an Efinity development board and/or a testbench. This wizard is helpful in situations in which you use several IP cores, multiple instances of an IP core with different parameters, or the same IP core for different projects.



**Note:** Not all Efinity IP cores include an example design or a testbench.

## Generating a Core with the IP Manager

The following steps explain how to customize an IP core with the IP Configuration wizard.

1. Open the IP Catalog.
2. Choose an IP core and click **Next**. The **IP Configuration** wizard opens.
3. Enter the module name in the **Module Name** box.



**Note:** You cannot generate the core without a module name.

4. Customize the IP core using the options shown in the wizard. For detailed information on the options, refer to the IP core's user guide or on-line help.
5. (Optional) In the **Deliverables** tab, specify whether to generate an IP core example design targeting an Efinity® development board and/or testbench. For SoCs, you can also optionally generate embedded software example code. These options are turned on by default.
6. (Optional) In the **Summary** tab, review your selections.
7. Click **Generate** to generate the IP core and other selected deliverables.
8. In the **Review configuration generation** dialog box, click **Generate**. The Console in the **Summary** tab shows the generation status.



**Note:** You can disable the **Review configuration generation** dialog box by turning off the **Show Confirmation Box** option in the wizard.

9. When generation finishes, the wizard displays the **Generation Success** dialog box. Click **OK** to close the wizard.

The wizard adds the IP to your project and displays it under **IP** in the Project pane.

## Generated Files

The IP Manager generates these files and directories:

- **<module name>\_define.vh**—Contains the customized parameters.
- **<module name>\_tpl.v**—Verilog HDL instantiation template.
- **<module name>\_tpl.vhd**—VHDL instantiation template.
- **<module name>.v**—IP source code.
- **settings.json**—Configuration file.
- **<kit name>\_devkit**—Has generated RTL, example design, and Efinity® project targeting a specific development board.



**Note:** Refer to the IP Manager chapter of the Efinity® Software User Guide for more information about the Efinity® IP Manager.

# Customizing the MIPI D-PHY RX Controller

You can choose to generate the testbench when generating the core in the IP Manager Configuration window. To generate testbench, the **Testbench Deliverables Option** signals must be enabled.



**Note:** You must include all **.v** files generated in the **/Testbench** directory in your simulation.



**Important:** Efinix tested the testbench generated with the default parameter options only.

Efinix provides a simulation script for you to run the testbench quickly using the Modelsim software. To run the Modelsim testbench script, run `vsim -do modelsim.do` in a terminal application. You must have Modelsim installed on your computer to use this script.

*Table 6: MIPI D-PHY RX Controller Core Parameter*

Name	Options	Description
t <sub>LPX</sub> (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 50
t <sub>CLK_TERM_EN</sub> (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 38
t <sub>D_TERM_EN</sub> (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 35
t <sub>HS_SETTLE</sub> (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 85
t <sub>HS_PREPARE_ZERO</sub> (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 145
HS BYTECLK (MHz)	10 - 187	MIPI parallel clock frequency in MHz to support data rate of 80 Mbps to 1,500 Mbps. Default: 100
Data Lane	1, 2, 4, 8	Number of data lanes. Default: 4
D-PHY Clock Mode	Continuous, Discontinuous	Enables discontinuous or continuous HS mode clock. Default: continuous
Enable Bidir	0 or 1	Enable bidirectional data lane. Applicable to bidirectional mode only. Default: 1
t <sub>HS_PREPARE</sub> (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Applicable to bidirectional mode only. Default: 40
t <sub>WAKEUP</sub> (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Applicable to bidirectional mode only. Default: 1000

Name	Options	Description
t <sub>HS_ZERO</sub> (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Applicable to bidirectional mode only. Default: 105
t <sub>HS_TRAIL</sub> (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns before adding 4UI. Applicable to bidirectional mode only. Actual THS TRAIL = t <sub>HS_TRAIL_NS</sub> + 4UI or 8UI (whichever bigger). Default: 60
CLOCK_FREQ_MHZ	40 - 100	Core clock frequency in Mhz. Default: 100
t <sub>HS_EXIT</sub> (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 100

# MIPI D-PHY RX Controller Example Design

You can choose to generate the example design when generating the core in the IP Manager Configuration window. Compile the example design project and download the **.hex** or **.bit** file to your board. To generate example design, the **Example Design Deliverables Option** signal must be enabled.

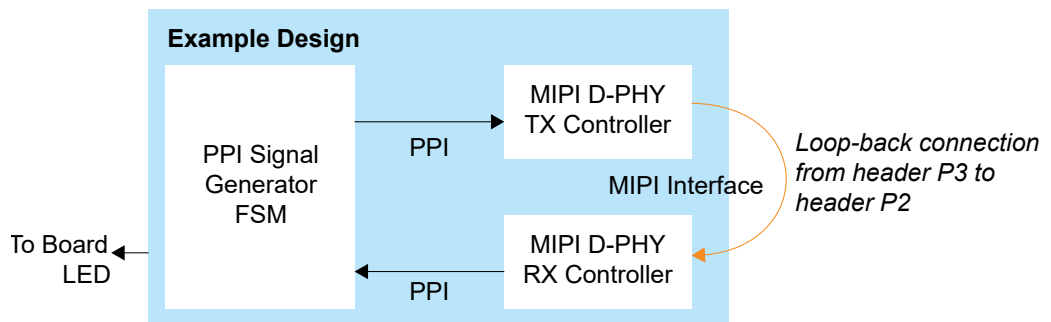


**Important:** Efinix tested the example design generated with the default parameter options only.

The example design targets the Titanium Ti60 F225 Development Board. The design instantiates both MIPI D-PHY TX Controller and MIPI D-PHY RX Controller cores. This design requires a female-to-female QTE header cable.

The PPI signal generator FSM generates a data and sends back the data through the MIPI D-PHY TX Controller and MIPI D-PHY RX Controller. The PPI signal generator FSM compares the sent and received data, and outputs the results using the board LEDs.

*Figure 2: MIPI D-PHY RX Controller Core Example Design*



After power-up, LEDs D19 and D18 turn on if the received data and the generated data matches. The RX clock to RX data skew can vary from board to board, therefore, there is a possibility that the RX clock might not be able to capture the RX data correctly. In this case, the LEDs do not turn on. Use the **Static Mode Delay Setting** in the Interface Designer for the `mipi_dphy_rx_clk` to adjust the delay up or down.

*Table 7: Example Design Implementation*

FPGA	Data Lane	Logic and Adders	Flip-flops	Memory Blocks	DSP48 Blocks	$f_{MAX}$ (MHz) <sup>(4)</sup>			Efinity® Version <sup>(5)</sup>
						clk1	clk2	clk3	
Ti60 F225 C4	Unidirectional	1,632	783	0	0	295	331	340	2021.2
	Bidirectional	1,931	913	0	0	317	353	331	2021.2

- `clk1`—`mipi_clk`
- `clk2`—`mipi_dphy_rx_clk_CLKOUT`
- `clk3`—`mipi_dphy_tx_SLOWCLK`

<sup>(4)</sup> Using default parameter settings.

<sup>(5)</sup> Using Verilog HDL.

# MIPI D-PHY RX Controller Testbench



**Note:** Contact Efinix support for MIPI D-PHY RX Controller core simulation testbench.

## Revision History

*Table 8: Revision History*

Date	Document Version	IP Version	Description
June 2025	2.0	Unidirectional - 5.7 Bidirectional - 5.11	Add support for trigger escape command. (SIP-953) Added RxTriggerEsc [3:0] and changed TxTriggerEsc [3:0] to input in PHY Protocol Interface table.
May 2025	1.9	Unidirectional - 5.7 Bidirectional - 5.10	Update default parameter value. (SIP-910) Example Design IO bank update (HVIO 3.3V) (SIP-907) Updated Customizing the MIPI D-PHY RX Controller.
March 2025	1.8	Unidirectional - 5.6 Bidirectional - 5.9	Remove unsupported dynamic delay control feature. (SIP-841)
November 2024	1.7	5.8	Added Topaz in Features and Device Support. (DOC-2102) Added IP Version in Revision History. (DOC-2185)
September 2024	1.6	-	Added 8 lanes support in Features and <b>Table 6: MIPI D-PHY RX Controller Core Parameter</b> on page 12. (SIP-677)
June 2024	1.5	-	Revised supported lane number. Removed 8 lane in Features and table MIPI D-PHY RX Controller Core Parameter. (SIP-578) Added important note in Example Design and Testbench regarding using default parameters options only. (DOC-1781)
June 2023	1.4	-	Added Device Support and release notes sections. (DOC-1234) Updated supported data rate. (DOC-1217) Updated HS BYTECLK and CLOCK_FREQ_MHZ parameter. Editorial changes.
February 2023	1.3	-	Added note about the resource and performance values in the resource and utilization table are for guidance only.
January 2022	1.2	-	Updated resource utilization table. (DOC-700)

Date	Document Version	IP Version	Description
October 2021	1.1	-	<p>Added note to state that the <math>f_{MAX}</math> in Resource Utilization and Performance, and Example Design Implementation tables were based on default parameter settings.</p> <p>Updated design example target board to production Titanium Ti60 F225 Development Board and updated Resource Utilization and Performance, and Example Design Implementation tables. (DOC-553)</p>
June 2021	1.0	-	Initial release.