MIPI D-PHY TX Controller Core User Guide

UG-CORE-MIPI-D-PHY-TX-v1.4
June 2023
www.efinixinc.com

Contents

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 TX Controller core allows you to control and configure the MIPI D-PHY TX interface. You use the MIPI D-PHY TX Controller core with the Titanium HSIO pins configured as MIPI TX lanes to build a custom MIPI interface.

Use the IP Manager to select IP, customize it, and generate files. The MIPI D-PHY TX 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 to 187 MHz (data rate of 80 Mbps to 1,500 Mbps)
- 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 TX Controller Core Device Support

FPGA Family	Supported Device
Trion	-
Titanium	All

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 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_{MAX} (MHz)[1]		Efinity® Version[2]
						clk	clk_byte_HS	
Ti60 F225 C4	Unidirectional	897	302	0	0	330	456	2021.2
	Bidirectional	1,162	384	0	0	400	434	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 is available in the Efinity Downloads page under each Efinity software release version.

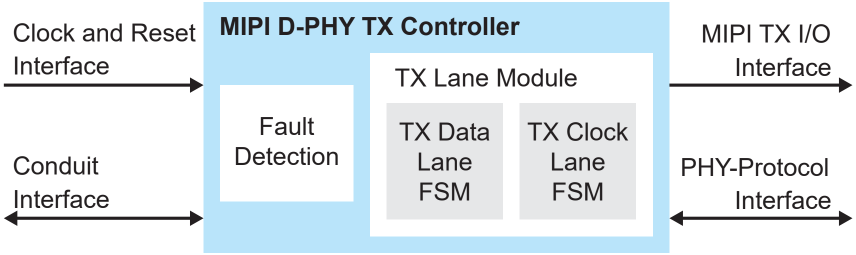
Note: You must be logged in to the Support Portal to view the IP Core Release Notes.

[1] Using default parameter settings.
[2] Using Verilog HDL.

Functional Description

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

Figure 1: MIPI D-PHY TX 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 TX parallel clock. This is a HS mode transmission clock.
reset_byte_HS_n	Input	MIPI TX parallel clock reset signal.

Table 4: MIPI TX I/O Interface

Port	Direction	Description
Tx_LP_CLK_N	Output	LP mode TX clock single-ended N signal.
Tx_LP_CLK_P	Output	LP mode TX clock single-ended P signal.
Tx_LP_CLK_N_OE	Output	Output enable for LP mode TX clock single-ended N signal.
Tx_LP_CLK_P_OE	Output	Output enable for LP mode TX clock single-ended P signal.
Tx_HS_enable_C	Output	Signal to enable HS mode clock lane.
Tx_HS_C [7:0]	Output	HS mode differential clock bus.
Tx_LP_D_P [NUM_DATA_LANE-1:0]	Output	LP mode TX data single-ended P signal.
Tx_LP_D_N [NUM_DATA_LANE-1:0]	Output	LP mode TX data single-ended N signal.
Tx_LP_D_P_OE [NUM_DATA_LANE-1:0]	Output	Output enable for LP mode TX data single-ended P signal.
Tx_LP_D_N_OE [NUM_DATA_LANE-1:0]	Output	Output enable for LP mode TX data single-ended N signal.
Tx_HS_D_0[7:0]	Output	HS mode differential lane 0 data bus.
Tx_HS_D_1[7:0]	Output	HS mode differential lane 1 data bus.
Tx_HS_D_2[7:0]	Output	HS mode differential lane 2 data bus.
Tx_HS_D_3[7:0]	Output	HS mode differential lane 3 data bus.
Tx_HS_D_4[7:0]	Output	HS mode differential lane 4 data bus.
Tx_HS_D_5[7:0]	Output	HS mode differential lane 5 data bus.
Tx_HS_D_6[7:0]	Output	HS mode differential lane 6 data bus.
Tx_HS_D_7[7:0]	Output	HS mode differential lane 7 data bus.
Tx_HS_enable_D [NUM_DATA_LANE-1:0]	Output	Signal to enable HS mode data lane.
Rx_LP_D_P	Input	LP mode RX data single-ended P signal for bidirectional data lane.
Rx_LP_D_N	Input	LP mode RX data single-ended N signal for bidirectional data lane.

Table 5: PHY-Protocol Interface

Port	Direction	Description
TxRequestHS [NUM_DATA_LANE-1:0]	Input	HS Transmit Request and Data Valid. A low-to-high transition causes the lane module to initiate a start-of-transmission sequence. A high-to-low transition on causes the lane module to initiate an end-of-transmission sequence. This active-high signal also indicates that the protocol is driving valid data on TxDataHS to be transmitted.
TxDataHS_0 [7:0]	Input	HS Transmit Data Bus. HS data to be transmitted.
TxDataHS_1 [7:0]	Input	HS Transmit Data Bus. HS data to be transmitted.
TxDataHS_2 [7:0]	Input	HS Transmit Data Bus. HS data to be transmitted.
TxDataHS_3 [7:0]	Input	HS Transmit Data Bus. HS data to be transmitted.
TxDataHS_4 [7:0]	Input	HS Transmit Data Bus. HS data to be transmitted.
TxDataHS_5 [7:0]	Input	HS Transmit Data Bus. HS data to be transmitted.
TxDataHS_6 [7:0]	Input	HS Transmit Data Bus. HS data to be transmitted.
TxDataHS_7 [7:0]	Input	HS Transmit Data Bus. HS data to be transmitted.
TxSkewCalHS[3] [NUM_DATA_LANE-1:0]	Input	HS Transmit Skew Calibration. This is an optional signal to initiate the periodic deskew burst at the transmitter. A low-to-high transition causes the PHY to initiate the transmission of a skew calibration pattern. A high-to-low transition causes the PHY to end the transmission of a skew calibration pattern, and initiate an end-of-transmission sequence.
TxReadyHS [NUM_DATA_LANE-1:0]	Output	HS Transmit Ready. This active-high signal indicates that TxDataHS is accepted by the lane module to be serially transmitted.
TxRequestEsc [NUM_DATA_LANE-1:0]	Input	Escape Mode Transmit Request. This active-high signal is used to request escape sequences.
TxStopStateD [NUM_DATA_LANE-1:0]	Output	Lane in Stop State. This active-high signal indicates that the lane module is in stop state.
TxUlpsExit [NUM_DATA_LANE-1:0]	Input	Transmit ULPS Exit Sequence. This active-high signal is asserted when ULPS is active and the protocol is ready to leave ULPS.
TxUlpsActiveNot [NUM_DATA_LANE-1:0]	Output	ULPS (not) Active. This active-low signal is asserted to indicate that the lane is in ULPS.

[3] HS Transmit Skew Calibration is not supported in v2021.1EA.

Port	Direction	Description
TxUlpsEsc [NUM_DATA_LANE-1:0]	Input	Escape Mode Transmit ULPS. This active-high signal is asserted with TxRequestEsc to cause the lane module to enter the ULPS.
TxLpdtEsc [NUM_DATA_LANE-1:0]	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.
TxValidEsc [NUM_DATA_LANE-1:0]	Input	Escape Mode Transmit Data Valid. This active-high signal indicates that the protocol is driving valid data on TxDataEsc to be transmitted.
TxDataEsc_0 [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[0] is transmitted first. Data is captured on rising edges of clk.
TxDataEsc_1 [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_1[0] is transmitted first. Data is captured on rising edges of clk.
TxDataEsc_2 [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_2[0] is transmitted first. Data is captured on rising edges of clk.
TxDataEsc_3 [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_3[0] is transmitted first. Data is captured on rising edges of clk.
TxDataEsc_4 [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_4[0] is transmitted first. Data is captured on rising edges of clk.
TxDataEsc_5 [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_5[0] is transmitted first. Data is captured on rising edges of clk.
TxDataEsc_6 [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_6[0] is transmitted first. Data is captured on rising edges of clk.
TxDataEsc_7 [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_7[0] is transmitted first. Data is captured on rising edges of clk.
TxReadyEsc [NUM_DATA_LANE-1:0]	Output	Escape Mode Transmit Ready. This active-high signal indicates that TxDataEsc is accepted bythe lane module to be serially transmitted. TxReadyEsc is valid on rising edges of TxClkEsc.

Port	Direction	Description
TxRequestHSc	Input	HS Transmit Request and Data Valid. A low-to-high transition causes the lane module to initiate a start-of-transmission sequence. A high-to-low transition causes the lane module to initiate an end-of-transmission sequence. This active-high signal causes the lane module to begin transmitting a HS clock.
TxReadyHSc	Output	HS Transmit Ready. This active-high signal indicates that the lane is transmitting a HS clock.
TxUlpsClk	Input	Transmit ULPS on Clock Lane. This active-high signal is asserted to cause a Clock lane module to enter the ULPS.
TxUlpsExitClk	Input	Transmit ULPS Exit Sequence. This active-high signal is asserted when ULPS is active and the protocol is ready to leave ULPS.
TxUlpsActiveClkNot	Output	ULPS (not) Active. This active-low signal is asserted to indicate that the lane is in ULPS.
TxStopStateC	Output	Lane in Stop state. This active-high signal indicates that the lane module is in stop state.
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.
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.
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.
RxUlpsEsc	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. This signal is used only in Bidir data lane mode.
RxUlpsActiveNot	Output	ULPS (not) Active. This active-low signal is asserted to indicate that the lane is in ULPS. This signal is used only in bidirectional data lane mode.
RxLPDTEsc	Output	Escape LP Data Receive Mode. This active-high signal is asserted to indicate that the lane module is in LP data receive mode. This signal is used only in bidirectional data lane mode.
RxValidEsc	Output	Escape LP Data Receive Mode. This active-high signal is asserted to indicate that the lane module is in LP data receive mode. This signal is used only in bidirectional data lane mode.

Port	Direction	Description
RxStopState	Output	Lane in Stop State. This active-high signal indicates that the lane module is currently in stop state. This signal is used only in bidirectional data lane mode.
RxDataEsc [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] was received first. Data is transferred on rising edges of clk. This signal is used only in bidirectional data lane mode.
RxTriggerEsc [3:0]	Output	Escape Mode 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.
ErrEsc	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. Applicable to bidirectional mode only.
ErrControl	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. Applicable to bidirectional mode only.

IP Manager

The Efinity® IP Manager is an interactive wizard that helps you customize and generate Efinix® 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 Efinix 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 Efinix 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 Efinix® 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>_tmpl.v**—Verilog HDL instantiation template.
- **<module name>_tmpl.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 TX Controller

You can choose to generate the testbench when generating the core in the IP Manager Configuration window.

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

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 TX Controller Core Parameter

Name	Option	Description
tLPX (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 50
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
DPHY Clock Mode	Continuous, Discontinuous	Enables discontinuous or continuous HS clock. Default: continuous
tLP_EXIT (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 100
tWAKEUP (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 1000
tHS_EXIT (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 100
tHS_ZERO (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 105
tHS_TRAIL (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns before adding 4UI. Actual THS TRAIL = tHS_TRAIL_NS + 4UI or 8UI (whichever bigger). Default: 60
tCLK_ZERO (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 262
tCLK_TRAIL (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 60
tCLK_POST (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns before adding UI52. Default: 60
tCLK_PREPARE (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 38

Name	Option	Description
tCLK_PRE (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns before adding UI52. Default: 10
Bus Turnaround Timeout (ns)	–	Bus turnaround timeout parameter in ns. If the D-PHY RX does not respond within this period, D-PHY TX ends the turnaround request. Applicable to bidirectional mode only. Default: 100000
tD_TERM_EN (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Applicable to bidirectional mode only. Default: 35
tHS_PREPARE_ZERO (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Applicable to bidirectional mode only. Default: 145
Enable Bidir Mode	0 or 1	Enable bidirectional data lane. Applicable to bidirectional mode only. Default: 1
CLOCK_FREQ_MHZ	40 - 100	Core clock frequency in MHz. Default: 100
tHS_PREPARE (ns)	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 50

MIPI D-PHY TX Controller Example Design

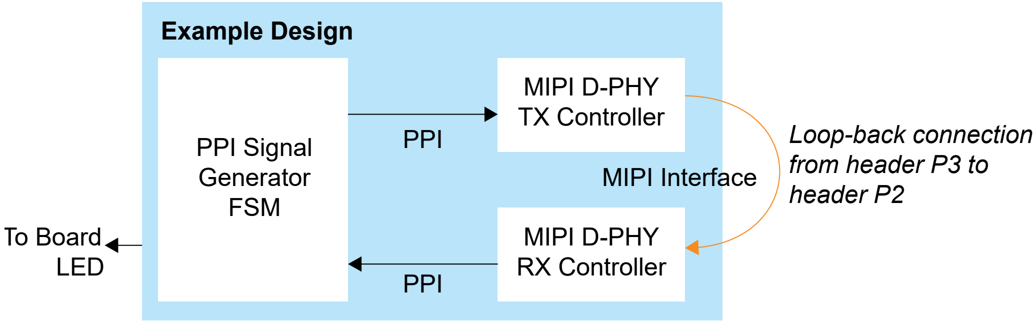
You can choose to generate the example design when generating the core in the IP Manager Configuration window. Compile the example design project and download the **.hex** or **.bit** file to your board.

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 TX 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)[4]			Efinity® Version[5]
						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

[4] Using default parameter settings.
[5] Using Verilog HDL.

MIPI D-PHY TX Controller Testbench

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

Revision History

Table 8: Revision History

Date	Version	Description
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)
October 2021	1.1	Added note to state that the f_{MAX} in Resource Utilization and Performance, and Example Design Implementation tables were based on default parameter settings. 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)
June 2021	1.0	Initial release.