MIPI DSI RX Controller Core User Guide

UG-CORE-MIPI-DSI-RX-v1.0
February 2024
www.efinixinc.com

Contents

Introduction

The MIPI DSI specifies the physical link between the chip and display in devices such as smartphones, tablets, AR/VR headsets and connected cars[1]. It defines a serial bus and a communication protocol between the host, the source of the image data, and the destination, for example, display peripherals. The MIPI DSI RX Controller core implements the MIPI DSI interface in the FPGA and allows you to configure the related parameters.

Features

- Supports 1,2, and 4 lanes
- Supports continuous or discontinuous clock mode
- IP core clock frequency at 100 MHz
- 8-bit HS mode data width
- HS mode byte clock frequency from 10 MHz to 187 MHz (80 Mbps to 1500 Mbps data rate)
- Includes AXI4-Lite interface for register access
- Error correction code (ECC) generation for packet headers
- Cyclic redundancy check (CRC) generation for data bytes
- Supports non-burst with sync pulses, non-burst with sync events, and burst mode
- Supports command transmission in HS or LP mode
- Supports all Titanium FPGAs

Device Support

Table 1: MIPI DSI RX Controller Core Device Support

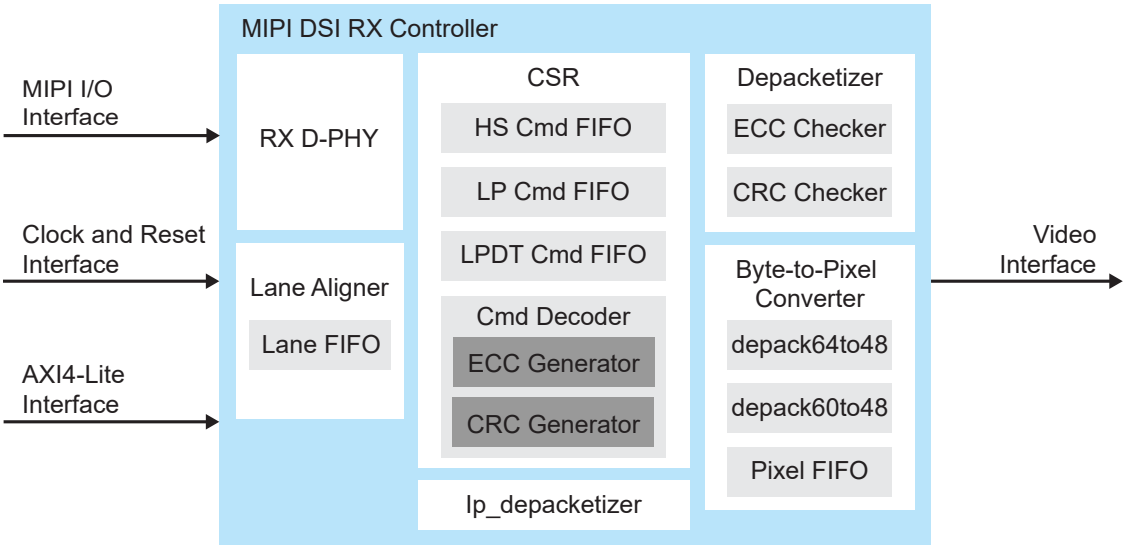
FPGA Family	Supported Device
Trion	-
Titanium	All

[1] Source: MIPI Alliance.

Functional Description

The MIPI DSI RX Controller consists of a RX D-PHY block, control status registers, ECC and CRC checkers, packetizer, pixel-to-byte converter, and video interface detector. The core has a video, AXI4-lite, MIPI RX I/O, and clock and reset interfaces.

Figure 1: MIPI DSI RX Controller System Block Diagram



Ports

Table 2: Clock and Reset Ports

Port	Direction	Description
clk	Input	IP core clock consumed by controller logic. 100 MHz.
reset_n	Input	IP core reset signal.
clk_byte_HS	Input	MIPI RX parallel clock. This clock is a HS mode reception clock.
reset_byte_HS_n	Input	MIPI RX parallel clock reset signal.
clk_pixel	Input	Pixel clock.
reset_pixel_n	Input	Pixel clock reset signal.
axi_clk	Input	AXI4-Lite interface clock.
axi_reset_n	Input	AXI4-Lite interface reset.

Table 3: 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 data bus.
Rx_HS_D_1[7:0]	Input	HS mode differential lane data bus.
Rx_HS_D_2[7:0]	Input	HS mode differential lane data bus.
Rx_HS_D_3[7:0]	Input	HS mode differential lane 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	Enable dynamic delay for Rx data lane.
DLY_inc_D[NUM_DATA_LANE-1:0]	Output	Increment dynamic delay for Rx data lane.
u_dly_enable_D[NUM_DATA_LANE-1:0]	Input	Controls the RX data lane dynamic delay. Used together with u_dly_inc_D. Refer to u_dly_inc_D for usage examples. Available when ENABLE_USER_DESKEWCAL = 1.

Port	Direction	Description
u_dly_inc_D[NUM_DATA_LANE-1:0]	Input	Controls the RX data lane dynamic delay. Example: u_dly_inc_D = 1 and u_dly_enable_D =1, the delay step increases every clock cycle. u_dly_inc_D = 0 and u_dly_enable_D = 1, the delay step decreases every clock cycle. u_dly_enable = 0, the delay value stays put. Available when ENABLE_USER_DESKEWCAL = 1.
Tx_LP_D_P	Output	LP mode TX data single-ended P signal.
Tx_LP_D_P_OE	Output	Output enable for LP mode TX data single-ended P signal.
Tx_LP_D_N	Output	LP mode TX data single-ended N signal.
Tx_LP_D_N_OE	Output	Output enable for LP mode TX data single-ended N signal.

Table 4: Video Interface

All signals are clocked by clk_pixel.

Port	Direction	Description
hsync	Output	Active-low horizontal sync signal.
vsync	Output	Active-low vertical sync signal.
pixel_data [63:0]	Output	Video Data. The actual data width of this port is dependent on pixel type. See Video Mode Pixel Encoding on page 12.
pixel_data_valid	Output	Active-high pixel data enable.
pixel_vc[1:0]	Output	Virtual channel signal of packet received by DSI RX controller.
pixel_format[5:0]	Output	Pixel data format of packet received by DSI RX controller.

Table 5: Side band Interface

Allsignals are clocked by clk_byte_HS except irq which is clocked by axi_clk.

Port	Direction	Description
video_format	Input	Set to related video format datatype for cycle-accurate pixel interface generation when converting the decoded packets (in byte clock domain) to pixel interface (in pixel clock domain). E.g. set to 6'h3E if using RGB888 format. Tie to zeros if there is no concern for inaccuracy of pixel interface timing (especially on HSA, and HBP), then the pixel interface (hsync, vsync, pixel_data, pixel_data_valid) will be decoded based on byte clock 1domain.
vc[1:0]	Output	2-bit virtual channel signal.
word_count[15:0]	Output	Byte count of the long packet received by DSI RX controller.
datatype[5:0]	Output	Decoded data type of packet received by DSI RX controller.
irq	Output	Interrupt signal for Interrupt Status Register.

Table 6: AXI4-Lite Interface

All signals are clocked by axi_clk.

Port	Direction	Description
axi_awaddr [6:0]	Input	AXI4-Lite write address bus.
axi_awvalid	Input	AXI4-Lite write address valid strobe.
axi_awready	Output	AXI4-Lite write address ready signal.
axi_wdata [31:0]	Input	AXI4-Lite write data.
axi_wvalid	Input	AXI4-Lite write data valid strobe.
axi_wready	Output	AXI4-Lite write ready signal.
axi_bvalid	Output	AXI4-Lite write response valid strobe.
axi_bready	Input	AXI4-Lite write response ready signal.
axi_araddr [6:0]	Input	AXI4-Lite read address bus.
axi_arvalid	Input	AXI4-Lite read address valid strobe.
axi_arready	Output	AXI4-Lite read address ready signal.
axi_rdata [31:0]	Output	AXI4-Lite read data.
axi_rvalid	Output	AXI4-Lite read data valid strobe.
axi_rready	Input	AXI4-Lite read data ready signal.

Register Definition

Table 7: Control Status Registers

Word Offset	Bits	Name	R/W	Width (bits)
0x00	16:0	Interrupt status register.	R/W1C	17
0x04	16:0	Interrupt enable register.	R/W	17
0x08	6:0	Rx DPHY Data lane0 status..	RO	7
0x0C	6:0	Rx DPHY Data lane1 status.	RO	7
0x10	6:0	Rx DPHY Data lane2 status.	RO	7
0x14	6:0	Rx DPHY Data lane3 status.	RO	7
0x18 - 0x24	N/A	Reserved.	N/A	N/A
0x28	1:0	Rx DPHY clock lane status.	RO	2
0x2C	31:0	Received high speed write command/payload for short/long packet (hs_cmdfifo_data).	RO	32
0x30	7:0	Received low power write command/payload for short/long packet (lp_cmdfifo_data).	RO	8
0x34	31:0	LPDT read command data return. Write into this register to transmit the data return from peripheral (DSI RX) to host (DSI TX) (lp_dcs_rfifo_data)	WO	32
0x38	23:0	ESC mode LPDT command.	WO	24
0x2C - 0x40	N/A	Reserved	N/A	N/A
0x44	15:0	Horizontal sync active (HSA) in pixel count. Only write to this register when it is sync pulse mode. sync pulse mode: minimum = 2, other modes: minimum = 1	R/W	16
0x48	15:0	Horizontal black porch (HBP) in pixel count. For burst event mode, factor in HSA value into the HBP value (not in used as per current implemenation).	R/W	16
0x4C	15:0	Horizontal front porch (HFP) in byte (not in used as per current implemenation).	R/W	16
0x50	7:0	Vertical sync active (VSA) in line. The minimum number of lines is 1.	R/W	8
0x54	7:0	Vertical black porch (VBP) in line. The minimum number of lines is 1 (not in used as per current implemenation).	R/W	8
0x58	7:0	Vertical front porch (VFP) in line. The minimum number of lines is 2 (not in used as per current implemenation).	R/W	8

Table 8: Interrupt Status Register Definition (0x00)

Bit	Name	Description
0	pixel_fifo_full	Pixel FIFO full Pixel FIFO in the byte-to-pixel converter module is full.
1	pixel_fifo_empty	Pixel FIFO empty Pixel FIFO in the byte-to-pixel converter module is empty.
2	undersize_pkt_error	Undersize packet error The incoming MIPI HS data byte is lesser than the wordcount value.
3	receive_error	Initialization error MIPI HS data is received before tInit is completed.
4	hs_cmdfifo_full	HS command fifo is full.
5	lp_cmdfifo_full	LP command fifo is full.
6	lp_dcs_rfifo_full	LP DCS read data fifo is full.
7	hs_cmdfifo_empty	HS command fifo is empty.
8	lp_cmdfifo_empty	LP command fifo is empty.
9	lp_dcs_rfifo_empty	LP DCS read data fifo is empty.
10	lp_cmd_in_progress	LP command transmission in LP lane is in progress.
11	hs_crc_error	HS packet received CRC error indicator.
12	hs_ecc_1bit_error	HS packet received ECC 1-bit error indicator.
13	hs_ecc_2bit_error	HS packet received ECC 2-bit error indicator.
14	lp_crc_error	LP packet received CRC error indicator.
15	lp_ecc_1bit_error	LP packet received ECC 1-bit error indicator.
16	lp_ecc_2bit_error	LP packet received ECC 2-bit error indicator.

Table 9: Interrupt Enable Register Definition (0x04)

Each enabled interrupt status bit is aggregated to the irq output port as an indicator. By default, all interrupt enable registers are set to 1'b0 (disabled).

Bit	Name	Description
0	pixel_fifo_full	Enable interrupt generation for pixel_fifo_full status register.
1	pixel_fifo_empty	Enable interrupt generation for pixel_fifo_empty status register.
2	undersize_pkt_error	Enable interrupt generation for undersize_pkt_error status register.
3	receive_error	Enable interrupt generation for receive_error status register.
4	hs_cmdfifo_full	Enable interrupt generation for hs_cmdfifo_full status register.
5	lp_cmdfifo_full	Enable interrupt generation for lp_cmdfifo_full status register.
6	lp_dcs_rfifo_full	Enable interrupt generation for lp_dcs_rfifo_full status register.
7	hs_cmdfifo_empty	Enable interrupt generation for hs_cmdfifo_empty status register.
8	lp_cmdfifo_empty	Enable interrupt generation for lp_cmdfifo_empty status register.
9	lp_dcs_rfifo_empty	Enable interrupt generation for lp_dcs_rfifo_empty status register.
10	lp_cmd_in_progress	Enable interrupt generation for LP command transmission in progress.
11	hs_crc_error	Enable interrupt generation for CRC error during hs packet reception.
12	hs_ecc_1bit_error	Enable interrupt generation for ECC 1-bit error during hs packet reception.
13	hs_ecc_2bit_error	Enable interrupt generation for ECC 2-bit error during hs packet reception.

Bit	Name	Description
14	lp_crc_error	Enable interrupt generation for CRC error during lp packet reception.
15	lp_ecc_1bit_error	Enable interrupt generation for ECC 1-bit error during lp packet reception.
16	lp_ecc_2bit_error	Enable interrupt generation for ECC 2-bit error during lp packet reception.

Table 10: D-PHY Status for Data Lanes Register Definition (0x08 – 0x24)

Bit	Name	Description
0	RxErrSotSyncHS	Start-of-Transmission(SoT) Synchronization Error. The core asserts this signal high for one cycle of RxWordClkHS if the HS SoT leader sequence is corrupted in a way that proper synchronization cannot be expected.
1	RxErrControl	Control Error. The core asserts this signal high 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.
2	RxErrEsc	Escape Entry Error. The core asserts this signal high if an unrecognized escape entry command is receivedin LP mode. Once asserted, this signal remains asserted until the next transaction starts, so that the protocol can properly process the error.
3	RxStopState	Lane is in stop state.
4	Reserved	Reserved.
5	RxUlpsActiveNot	Ultra Low Power State (ULPS) (not) Active. The core deasserts this signal low to indicate that the data lane is in ULP state.
6	RxUlpsEsc	EscapeULPS (Receive) mode. The core asserts this signal high to indicate that the lane module has entered the ULPS,due to the detection of a received ULPS command. The lane module remains in this mode with RxUlpsEsc asserted until a Stop state is detected on the interconnect lane.
7	Reserved	Reserved.

Table 11: D-PHY Status for Clock Lane Register Definition (0x28)

Bit	Name	Description
0	RxUlpsActiveClkNot	ULPS (not) Active. The core asserts this signal high to indicate that the clock lane is in ULPS.
1	RxUlpsClkNot	ReceiveULPS on Clock Lane. The core deasserts this signal low to indicate that the clock lane module has entered the ULPS due to the detection of a request to enter the ULPS. The lane module remains in this mode with RxUlpsClkNot asserted until a stop state is detected on the interconnect lane.

Video Mode Configuration

The MIPI DSI RX Controller core supports the following video modes:

- Non-burst with sync pulses
- Non-burst with sync events
- Burst mode

Command Packet Data Types

The following table describes the supported command packet data types. The command packets are sent through the command register. In LP mode, the command packets are sent through lane 0. Use the command to send non-video packets to the DSI RX Controller.

Table 12: Command Packet Data Types

Type	Data Type	Packet Size	Transfer Mode
0x2	Color mode off command	Short	LP/HS
0x12	Color mode on command	Short	LP/HS
0x22	Shutdown peripheral command	Short	LP/HS
0x32	Turn on peripheral command	Short	LP/HS
0x3	Generic short write, no parameters	Short	LP/HS
0x13	Generic short write, 1 parameters	Short	LP/HS
0x23	Generic short write, 2 parameters	Short	LP/HS
0x4	Generic short read, no parameters	Short	LP/HS
0x14	Generic short read, 1 parameters	Short	LP/HS
0x24	Generic short read, 2 parameters	Short	LP/HS
0x5	DCS short write, no parameters	Short	LP/HS
0x15	DCS short write, 1 parameters	Short	LP/HS
0x6	DCS read	Short	LP/HS
0x37	Set Max Return packet size	Short	LP/HS
0x29	Generic Long write	Long	LP/HS
0x39	DCS long write	Long	LP/HS

Sync Event Packet Data Type

Sync events are short packets and can accurately represent events like the start and end of sync pulses.

Table 13: Sync Events

Data type	Description	Packet Size
0x1	V sync start	Short
0x11	V sync end	Short
0x21	H sync start	Short
0x31	H sync end	Short

Video Mode Pixel Encoding

Table 14: Video Mode Pixel Encoding

TYPE[5:0]	Data Type	Bits per Pixel	Pixels per Pixel Clock	Bytes	Pack Bits	Packet Size	Transfer Mode
0xC	20-bit YCbCr	24	2	6	48	Long	HS
0x1C	24-bit YCbCr	24	2	6	48	Long	HS
0x2C	16-bit YCbCr	16	4	8	64	Long	HS
0x3D	12-bit YCbCr	12	4	6	48	Long	HS
0xE	RGB565	16	4	8	64	Long	HS
0x2E	RGB666 (24-bit)	24	2	6	48	Long	HS
0x3E	RGB888	24	2	6	48	Long	HS
0x0D	RGB101010	30	2	60/8	60	Long	HS

MIPI Video Data DATA[63:0] Formats

The format depends on the data type. New data arrives on every pixel clock.

Table 15: Loosely Packed Pixel Stream, 20-bit YCbCr (2-pixels per clock)

63 48	47 0						
0	Pixel 1 and Pixel 2						
N/A	[47:38] Y	[37:36] 2'b00	[35:26] Cr	[25:24] 2'b00	[23:14] Y	[13:12] 2'b00	[11:2] Cb	[1:0] 2'b00

Table 16: Packed Pixel Stream, 24-bit YCbCr (2-pixels per clock)

63 48	47 0			
0	Pixel 1 and Pixel 2			
N/A	[47:36] Y	[35:24] Cr	[23:12] Y	[11:0] Cb

Table 17: Packed Pixel Stream, 16-bit YCbCr (4-pixels per clock)

63 32	31 0						
Pixel 3 and Pixel 4				Pixel 1 and Pixel 2			
[63:56] Y	[55:48] Cr	[47:40] Y	[39:32] Cb	[31:24] Y	[23:16] Cr	[15:8] Y	[7:0] Cb

Table 18: Packed Pixel Stream, 12-bit YCbCr (4-pixels per clock)

63 48	47 24		23 0		
Odd Lines					
0	Pixel 3 and Pixel 4		Pixel 1 and Pixel 2		
N/A	[47:40] Y	[39:32] Y	[31:24] Cb	[23:16] Y	[15:8] Y	[7:0] Cb
Even Lines					
0	Pixel 3 and Pixel 4		Pixel 1 and Pixel 2		
N/A	[47:40] Y	[39:32] Y	[31:24] Cr	[23:16] Y	[15:8] Y	[7:0] Cb

Table 19: RGB565 (4-pixels per clock)

63 48	47 32	31 16	15 0
Pixel 4	Pixel 3	Pixel 2	Pixel 1

[63:59] Blue	[58:53] Green	[52:48] Red	[47:43] Blue	[42:37] Green	[36:32] Red	[31:27] Blue	[26:21] Green	[20:16] Red	[15:11] Blue	[10:5] Green	[4:0] Red

Table 20: RGB666 (2-pixels per clock)

63 48	47 24	23 0
0	Pixel 2	Pixel 1

N/A	[47:42] Blue	[41:40] 2'b00	[39:34] Green	[33:32] 2'b00	[31:26] Red	[25:24] 2'b00	[23:18] Blue	[17:16] 2'b00	[15:10] Green	[9:8] 2'b00	[7:2] Red	[1:0] 2'b00

Table 21: RGB888 (2-pixels per clock)

63 48	47 24	23 0
0	Pixel 2	Pixel 1

N/A	[47:40] Blue	[39:32] Green	[31:24] Red	[23:16] Blue	[15:8] Green	[7:0] Red

Table 22: RGB101010 (2-pixels per clock)

63 60	59 30	29 0
0	Pixel 2	Pixel 1

N/A	[59:50] Blue	[49:40] Green	[39:30] Red	[29:20] Blue	[19:10] Green	[9:0] Red

Pixel Clock Calculation

The following formula calculates the pixel clock frequency that you need to drive the pixel clock input port, `clk_pixel`.

PIX_CLK_MHZ = (DATARATE_MBPS * NUM_DATA_LANE) / PACK_BIT,

where:

- PIX_CLK_MHZ is the pixel clock in MHz
- DATARATE_MBPS is the MIPI data rate in Mbps
- NUM_DATA_LANE is the number of data lanes
- PACK_BIT is the Pixel data bits per pixel clock from Video Mode Pixel Encoding on page 12.

There is a FIFO for data transfer from the byte clock domain to the pixel clock domain. Pixel clock frequency should be applied approximately equal to the formula given above (or, within ±2% of the calculated value). If the pixel clock frequency is too low, there will be an occurrence of FIFO overflow (the read clock is slower than the write clock, thus, overflow happens). If the pixel clock frequency is too high, there will be an occurrence of FIFO underflow (which will cause the pixel_data_valid in a toggling fashion within one-pixel line).

Video Timing Parameters

The following waveforms show the video interface signals relationship.

Figure 2: Video Timing Waveform (Horizontal)

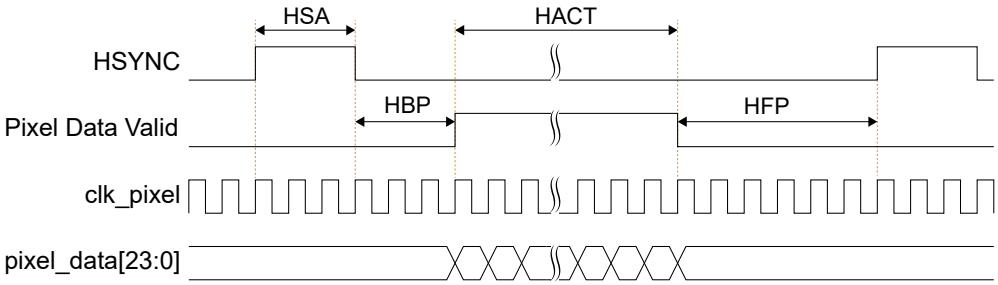


Figure 3: Video Timing Waveform (Vertical)

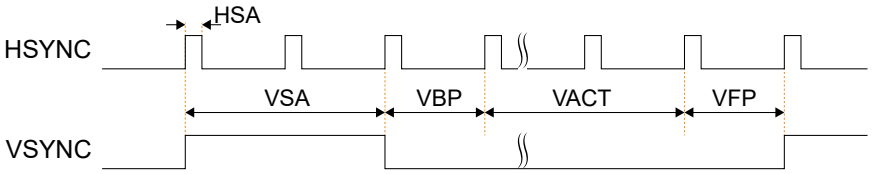


Table 23: Video Timing Parameter Definitions

MIPI Video Timing Parameters	Definition
HACT	Total number of pixel per line
VACT	Total number of line per frame
HSA	HSYNC pulse width
HBP	Horizontal back porch
HFP	Horizontal front porch
VSA	VSYNC pulse width
VBP	Vertical back porch
VFP	Vertical front porch
Pixel Clock	Video stream pixel clock frequency in MHz
MIPI Speed	DSI RX MIPI speed in Mbps
No. data lane	Number of MIPI data lane

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 the MIPI DSI RX Controller 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 **MIPI > MIPI DSI RX Controller** core and click **Next**. The **IP Configuration** wizard opens.
3. Enter the module name in the **Module Name** box.

Customizing the MIPI DSI RX Controller

The core has parameters so you can customize its function. You set the parameters in the General tab of the core's IP Configuration window.

Table 24: MIPI DSI RX Controller Core Parameter

Name	Option	Description
tLPX_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. The tLPX_NS ratio between host processor and peripheral must not exceed 3:2. The host processor is responsible for controlling its own clock frequency to match the peripheral. The host processor LP clock frequency must be in the range of 67% to 150% of peripheral LP clock frequency. Default: 50
tINIT_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 100000
tLP_EXIT_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 100
BTA_TIMEOUT_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 100000
tD_TERM_EN_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 35
tCLK_TERM_EN_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 38
tHS_PREPARE_ZERO_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 145
tHS_SETTLE_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 85
tHS_PREPARE_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 40
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. Default: 60
HS_BYTECLK_MHZ	10 – 187	MIPI parallel clock frequency in MHz. Round down the frequency value to an integer if there is any floating point from your calculation. Default: 125

Name	Option	Description
CLOCK_FREQ_MHZ	40 – 100	IP core clock frequency in MHz. Default: 100
NUM_DATA_LANE	1, 2, 4	Numberof data lanes. Default: 4
ENABLE_USER_DESKEWCAL	0,1	Allows user to control the RX data I/O lane dynamic delay. Default: 0
DPHY_CLOCK_MODE	Continuous, Discontinuous	DPHY clock mode. Default: Continuous
ENABLE_BIDIR	Enable, Disable	Turn on pack 48-bit datatype, for example, 20-bit YCbCr, 24-bit YCbCr, 12-bit YCbCr, RGB666 (24-bit), or RGB888. Default: Enable
PACK_TYPE	2'b11	bit[0]: Turn on depack 48-bit datatype, for example, 20-bit YCbCr, 24-bit YCbCr, 12-bit YCbCr, RGB666 (24- bit), or RGB888. bit[1]: Turn on depack 64-bit datatype, for example, 16-bit YCbCr, or RGB565. Default: 2'b11
PACKET_SEQUENCES	0, 1, 2	Select video mode: 0: Non-Burst Mode with Sync Pulses 1: Non-Burst Mode with Sync event (default) 2: Burst Mode
PIXEL_FIFO_DEPTH	256 – 8192	Pixel data FIFO depth. Default: 2048
HS_CMD_FIFO_DEPTH	8 – 2048	HS command write data FIFO depth. Default: 512
LP_CMD_FIFO_DEPTH	8 – 2048	LP command write data FIFO depth. Default: 512
LP_CMD_RDATAFIFO_DEPTH	8 – 2048	Bus turnaround read data depth. Default: 2048

Revision History

Table 25: Revision History

Date	Version	Description
February 2024	1.0	Initial release.