



MIPI DSI TX Controller Core User Guide

UG-CORE-MIPI-DSI-TX-v2.1
June 2023
www.elitestek.com



Contents

Introduction.....	3
Features.....	3
Device Support.....	3
Resource Utilization and Performance.....	4
Release Notes.....	5
Functional Description.....	6
Ports.....	6
Clocking.....	9
Register Definition.....	10
Video Mode Configuration.....	12
Command Packet Data Types.....	12
Sync Event Packet Data Type.....	13
Video Mode Pixel Encoding.....	13
MIPI Video Data DATA[63:0] Formats.....	14
Pixel Clock Calculation.....	14
Video Timing Parameters.....	15
IP Manager.....	16
Customizing the MIPI DSI TX Controller.....	17
MIPI DSI TX Controller Example Design.....	19
MIPI DSI TX Controller Testbench.....	21
Revision History.....	22

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 TX Controller core implements the MIPI DSI interface in the FPGA and allows you to configure the related parameters.

Use the IP Manager to select IP, customize it, and generate files. The MIPI DSI 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 易灵思® development board.

Features

- Supports 1,2, and 4 lanes
- Supports continuous or discontinuous clock mode
- IP core clock frequency at 100 MHz
- 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 end of transmission packet
- Supports command transmission in HS or LP mode

Device Support

Table 1: MIPI DSI TX Controller Core Device Support

FPGA Family	Supported Device
Trion	–
钛金系列	All

⁽¹⁾ Source: MIPI Alliance.

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: 钛金系列 Resource Utilization and Performance

MIPI DSI TX Controller with 4 data lanes.

FPGA	Logic and Adders	Flip-flops	Memory Blocks	f_{MAX} (MHz) ⁽²⁾				Efinity [®] Version ⁽³⁾
				clk	axi_clk	clk_byte_HS	clk_pixel	
Ti60 F225 C4	4,932	1,849	13	258	283	263	317	2021.2

⁽²⁾ Using default parameter settings.

⁽³⁾ Using SystemVerilog.

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.

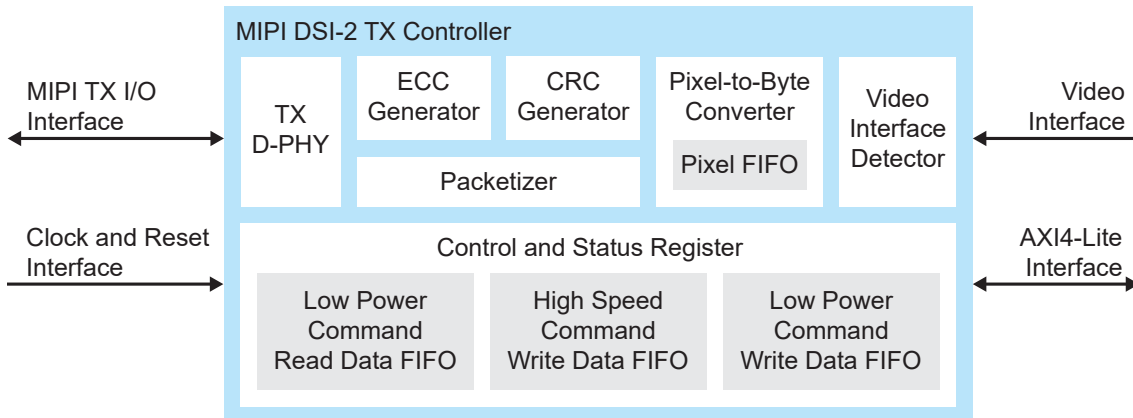


Note: You must be logged in to the Support Portal to view the IP Core Release Notes.

Functional Description

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

Figure 1: MIPI DSI TX Controller System Block Diagram



Ports

Table 3: 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 TX parallel clock. This clock is a HS mode transmission clock.
reset_byte_HS_n	Input	MIPI TX 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 4: MIPI TX I/O interface

Port	Direction	Description
Tx_LP_CLK_P	Output	LP mode TX clock single-ended P signal.
Tx_LP_CLK_P_OE	Output	Output enable for LP mode TX clock single-ended P signal.
Tx_LP_CLK_N	Output	LP mode TX clock single-ended N signal.
Tx_LP_CLK_N_OE	Output	Output enable for LP mode TX clock single-ended N signal.
Tx_HS_C [7:0]	Output	HS mode differential clock bus.
Tx_HS_enable_C	Output	Signal to enable HS mode clock lane.
Tx_LP_D_P [NUM_DATA_LANE-1:0]	Output	LP mode TX data single-ended P 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 [NUM_DATA_LANE-1:0]	Output	LP mode TX data single-ended N 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_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.
Rx_LP_D_N	Input	LP mode RX data single-ended N signal.

Table 5: Video Interface

All signals are clocked with `clk_pixel` and `reset_pixel_n`.

Port	Direction	Description
hsync	Input	Active-low horizontal sync.
vsync	Input	Active-low vertical sync.
datatype [5:0]	Input	Data type of the HS packet. Sampled at Hsync rising edge.
pixel_data [63:0]	Input	Video Data. Sampled when <code>pixel_data_valid</code> is high. The actual width is dependent on pixel type. See Video Mode Pixel Encoding on page 13.
pixel_data_valid	Input	Active-high pixel data enable.
haddr [15:0]	Input	16 bit horizontal number of pixels. Sampled at Hsync rising edge.
vc [1:0]	Input	2-bit virtual channel signal.

Table 6: Conduit Interface

Port	Direction	Description
TurnRequest_dbg	Input	User control 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 <code>clk</code> . 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.
irq	Output	Interrupt signal for Interrupt Status Register.

Table 7: AXI4-Lite Interface

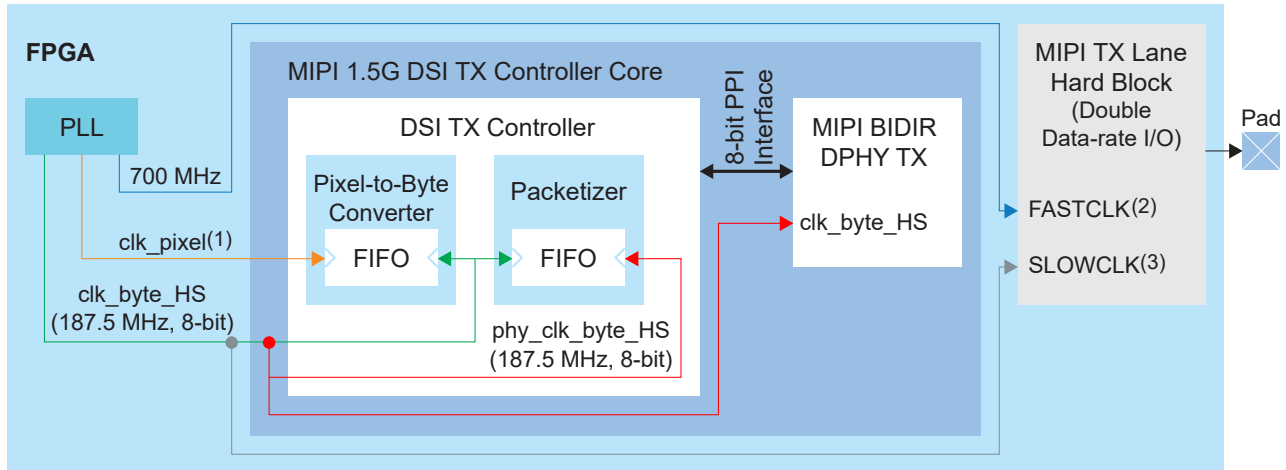
All signals are clocked with axi_clk.

Port	Direction	Description
axi_awaddr [6:0]	Input	AXI4-Lite write address bus.
axi_awvalid	Input	AXI4-Lite write address valid strobe.
axi_wready	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.

Clocking

The following diagram illustrates the example of clock settings for a 1.5 Gbps DSI TX implementation.

Figure 2: 1.5 Gbps DSI TX Clocking Example



- (1) Refer to Pixel Clock Calculation section for the `clk_pixel` value.
- (2) **Serial Clock Pin Name** in the Interface Designer.
- (3) **Parallel Clock Pin Name** in the Interface Designer.

Register Definition

Table 8: Control Status Registers

Word Offset	Bits	Name	R/W	Width (bits)
0x00	4:0	Interrupt status register.	R/W1C	5
0x04	4:0	Interrupt enable register.	R/W	5
0x08	7:0	PHY stop state status.	RO	8
0x0C	0	TxUlpsActiveClkNot.	RO	9
	8:1	TxUlpsActiveNot_7: TxUlpsActiveNot_0.		
0x10	7:0	Skew calibration high speed.	R/W	8
0x14	0	UlpsClk.	R/W	13
	8:1	UlpsEsc[7:0].		
	12:9	TxTriggerEsc.		
0x18	0	Reserved.	R/W	4
	1	Reserved.		
	2	Reserved.		
	3	video_stream_en. 1: Turn on HS video stream on the MIPI lane 0: Turn off HS video stream on the MIPI lane		
0x1C	HS Command Queue. Ensure that the bit 11 of the status register is low before issuing the next command.		R/W	24
	7:0	datatype.		
	15:8	Parameter 1.		
	23:16	Parameter 2.		
0x20	LP Command Queue. Ensure that the bit 10 of the status register is low before issuing the next command.		R/W	24
	7:0	datatype.		
	15:8	Parameter 1.		
	23:16	Parameter 2.		
0x24	19:0	Status register.	RO	20
0x28	31:0	Low power command write long data FIFO. You must write the LP write data to this FIFO before issuing the LP command packet to register 0x20. Store only one complete write packet data in the FIFO at a time.	WO	32
0x2C	31:0	High speed command write long data FIFO. You must write the HS write data to this FIFO before issuing the HS command packet to register 0x1C. Store only one complete write packet data in the FIFO. ⁽⁴⁾	WO	32
0x30	7:0	Low power command read long data FIFO. The bus turnaround read data is pushed into this FIFO. Store only one complete read packet data in the FIFO at a time. The MIPI DSI TX Controller stores all the return read data into the read FIFO and does not check whether the return read data matches maximum return packet size (MRPS).	RO	8
0x34, 0x38, 0x3C	Reserved			

⁽⁴⁾ The word count for a HS write long command data has to be larger or equal than the number of MIPI data lane (NUM_DATA_LANE).

Word Offset	Bits	Name	R/W	Width (bits)
0x40	31:0	Total H line word count in byte ⁽⁵⁾ .	R/W	32
0x44	15:0	Horizontal sync active (HSA) in byte ⁽⁵⁾ . Only write to this register when it is sync pulse mode.	R/W	16
0x48	15:0	Horizontal black porch (HBP) in byte ⁽⁵⁾ . For burst event mode, factor in HSA value into the HBP value.	R/W	16
0x4C	15:0	Horizontal front porch (HFP) in byte ⁽⁵⁾ .	R/W	16
0x50	7:0	Vertical sync active (VSA) in line ⁽⁵⁾ . The minimum number of lines is 1.	R/W	8
0x54	7:0	Vertical black porch (VBP) in line ⁽⁵⁾ . The minimum number of lines is 1.	R/W	8
0x58	7:0	Vertical front porch (VFP) in line ⁽⁵⁾ . The minimum number of lines is 2.	R/W	8
0x5C	15:0	Vertical active (VACT) in line ⁽⁵⁾ . The minimum number of lines is 1.	R/W	16

Table 9: 0x24 Status Register Definition

Bit	Description
0	lp_dcs_rfifo_full. LP command read data FIFO full.
1	lp_dcs_rfifo_empty. LP command read data FIFO empty.
2	lp_dcs_wfifo_full. LP command write data FIFO full.
3	lp_dcs_wfifo_empty. LP command write data FIFO empty.
4	hs_dcs_wfifo_full. HS command write data FIFO full.
5	hs_dcs_wfifo_empty. HS command write data FIFO empty.
6	Reserved.
7	Reserved.
8	Reserved.
9	Reserved.
10	lp_cmd_in_progress. LP command transmission in LP lane is in progress.
11	hs_cmd_in_progress. HS command transmission in HS lane is in progress.

Table 10: 0x00 Interrupt Status Register Definition

Bit	Description
0	Pixel FIFO full.
1	Pixel FIFO empty.
2	Unsupported video data type.
3	Turnaround timeout.

⁽⁵⁾ Refer to the MIPI DSI TX tab of the 钛金系列 MIPI Utility-v<version>.xslm file to calculate the correct value.

Video Mode Configuration

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

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

The following table describes the configuration for each video mode based on blanking or low-power interval (BLLP) mode setting.

Table 11: Video Mode Settings

Mode	HSA	HBP	HFP	BLLP
Non-burst with Sync Pulse	HS Blank Packet	HS Blank Packet	HS Blank Packet	HS Blank Packet
Non-burst with Sync Event	HS Blank Packet	HS Blank Packet	HS Blank Packet	HS Blank Packet
Burst	HS Blank Packet	HS Blank Packet	LP-11	LP-11

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 display peripherals.

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

MIPI Video Data DATA[63:0] Formats

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

Table 15: 20-bit YCbCr

63	48	47	24	23	0
0		Pixel 2		Pixel 1	

Table 16: 24-bit YCbCr

63	48	47	24	23	0
0		Pixel 2		Pixel 1	

Table 17: 16-bit YCbCr

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

Table 18: 12-bit YCbCr

63	48	47	36	35	24	23	12	11	0
0		Pixel 4		Pixel 3		Pixel 2		Pixel 1	

Table 19: RGB565

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

Table 20: RGB666 (24-bit)

63	48	47	24	23	0
0		Pixel 2		Pixel 1	

Table 21: RGB888

63	48	47	24	23	0
0		Pixel 2		Pixel 1	

Pixel Clock Calculation

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

$$\text{PIX_CLK_MHZ} < (\text{DATARATE_MBPS} * \text{NUM_DATA_LANE}) / \text{PACK_BIT},$$

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 13.

Video Timing Parameters

The following waveforms show the video interface signals relationship.

Figure 3: Video Timing Waveform (Horizontal)

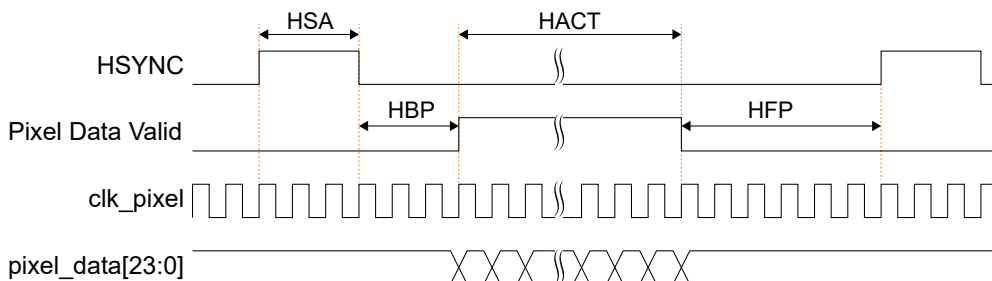


Figure 4: Video Timing Waveform (Vertical)

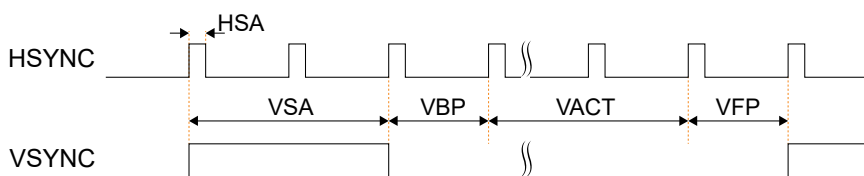


Table 22: Video Timing Parameter Definitions

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 TX 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 易灵思® 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 易灵思 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 易灵思 IP cores include an example design or a testbench.

Generating the MIPI DSI TX 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 TX Controller** 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 Customizing the MIPI DSI TX Controller section.
5. (Optional) In the **Deliverables** tab, specify whether to generate an IP core example design targeting an 易灵思® development board and/or testbench. 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.
- **Testbench**—Contains generated RTL and testbench files.

Customizing the MIPI DSI TX 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 23: MIPI DSI TX 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
Bus turnaround timeout period	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
tHS_PREPARE_ZERO_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 145
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. Default: 60
tCLK_PREPARE_NS	Values according to MIPI D-PHY specifications.	Soft D-PHY timing parameter in ns. Default: 38
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
NUM_DATA_LANE	1, 2, 4	Number of data lanes. Default: 4
MIPI Parallel Clock Frequency	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
IP Core Clock Frequency	40 - 100	IP core clock frequency in MHz. Default: 100

Name	Option	Description
DPHY_CLOCK_MODE	Continuous, Discontinuous	DHY clock mode. Default: Continuous
Pack Type 48	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 64	Enable, Disable	Turn on pack 64-bit datatype, for example, 16-bit YCbCr, or RGB565. Default: Enable
Enable Bus Turnaround in vertical low power mode	Enable, Disable	Enables the bus turnaround during the last vertical front porch line which goes into LP-11 state. Default: Disable
Maximum Horizontal Resolution	Values according to video display	Maximum horizontal pixel resolution. Default: 1080
Video Transmission Packet Sequences	Non-burst mode with Sync Pulses, Burst mode, Non-burst mode with Sync Events	Select video mode: 0: Non-Burst Mode with Sync Pulses 1: Non-Burst Mode with Sync event (default) 2: Burst Mode
High Speed Write Data FIFO DEPTH	8 – 2048 ⁽⁶⁾	HS command write data FIFO depth. Default: 512
Low power Write Data FIFO DEPTH	8 – 2048 ⁽⁶⁾	LP command write data FIFO depth. Default: 512
Low power Read Data FIFO DEPTH	8 – 2048 ⁽⁶⁾	Bus turnaround read data depth. Default: 2048
Pixel Data FIFO Depth Size	256 – 8192 ⁽⁶⁾	Pixel data FIFO depth. Set to the power of 2 value that is bigger than the 2 * (max horizontal pixel / 8). For example, when maximum horizontal pixel is 1280, set this parameter to 512. Default: 2048
Enable End Of Transmission Packet	Enable, Disable	Enables or disables the End Of Transmission Packet. Default: Disable
Enable bidirectional DPHY	Enable, Disable	To instantiate a unidirectional or bidirectional soft D-PHY. Default: Enable

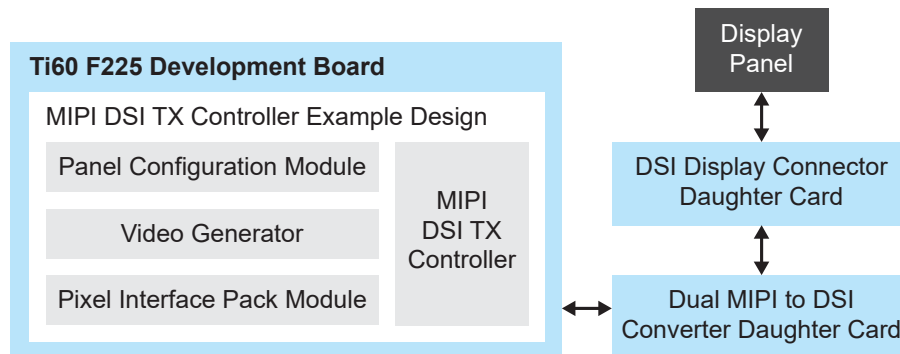
⁽⁶⁾ 2ⁿ, where n can be from 3 to 11.

MIPI DSI TX Controller Example Design

The example design targets the 钛金系列 Ti60 F225 Development Board. This design generates a video stream and sends the video data to a display panel through the MIPI DSI TX Controller. Apart from the 钛金系列 Ti60 F225 Development Board, the example design requires the following hardware:

- Dual MIPI to DSI Converter Daughter Card
- Mini-DSI Panel Connector Daughter Card
- Mini-DSI Panel display

Figure 5: MIPI DSI TX Controller Core Example Design



After power-up, press the reset button (SW5), then you should be able to see a video displayed on the panel module. The RX clock to RX data skew varies in different board, hence there is a possibility where the RX clock might not be able to capture the RX data correctly. In this case, you may not have display on the panel module. You have to try to increase the **Static Mode Delay Setting** of `mipi_dphy_tx_clk` in the Interface Designer.

Figure 6: MIPI DSI TX Controller Example Design Design Set Up



Table 24: Example Design Implementation

FPGA	Logic and Adders	Flip-flops	Memory Blocks	f _{MAX} (MHz) ⁽⁷⁾					Efinity® Version ⁽⁸⁾
				i_sys_clk	dsi_pclk	i_mipi_pclk	mipi_clk	i_mipi_tx_pclk	
Ti60 F225 C4	4,608	1,861	15	200	457	304	210	199	2021.2

⁽⁷⁾ Using default parameter settings.
⁽⁸⁾ Using Verilog HDL.

MIPI DSI TX Controller Testbench

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.

易灵思 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 25: Testbench Files

The IP Manager generates different encrypted source codes for you to simulate with different simulators.

Directory	Note
<code>../Testbench</code>	Contains the example design and testbench files.
<code>../Testbench/modelsim</code>	Contains the generated encrypted source code to simulate with the Modelsim simulator.
<code>../Testbench/ncsim</code>	Contains the generated encrypted source code to simulate with the NCSIM simulator.
<code>../Testbench/synopsys</code>	Contains the generated encrypted source code to simulate with the VCS simulator.

The simulation testbench simulates the example design. This design generates a video pattern to the MIPI DSI TX controller. The MIPI interface transaction from the controller is checked by a checker in the testbench.

After running the simulation successfully, the test prints the following message:

```
Received frame no. 1
Received frame no. 2
Received frame no. 3
Received frame no. 4
```

Revision History

Table 26: Revision History

Date	Version	Description
June 2023	2.1	Added MIPI Video Data Formats. (DOC-1233) Added Device Support and release notes sections. (DOC-1234) Updated supported data rate. (DOC-1217) Updated FIFO Pixel Depth Size parameter. Editorial changes.
April 2023	2.0	Updated MIPI Parallel Clock frequency parameter options and default value. (DOC-1186)
February 2023	1.9	Added note about the resource and performance values in the resource and utilization table are for guidance only. Corrected Titanium MIPI Utility-v<version>.xism file name.
January 2023	1.8	Corrected DATARATE_MPBS typo in Pixel Clock Calculation topic. Corrected Video Timing Waveform (Horizontal).
December 2022	1.7	Updated block diagram and added timing parameter waveforms. (DOC-1023)
August 2022	1.6	Added description to port tables about the clock domain used. (DOC-819) Corrected Control Status Registers definition. Corrected hsync and vsync video interface description. (DOC-879)
April 2022	1.5	Added minimum number of line for VSA, VBP, VFP, and VACT. (DOC-790)
March 2022	1.4	Updated description to HS and LP command queue register. (SIP-170)
January 2022	1.3	Updated resource utilization table. (DOC-700)
December 2021	1.2	Updated and added new IP Manager parameters. Update porch file to 钛金系列 MIPI Utility. Updated register definition. Updated core block diagram. Added simulation testbench.
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 钛金系列 Ti60 F225 Development Board and updated Resource Utilization and Performance, and Example Design Implementation tables. (DOC-553) Updated the display panel used in example design. Added 1 and 2 lanes support. Updated the MIPI TX I/O interface ports, Control Status Registers definition, Video Mode Configuration table, Video Mode Pixel Encoding table, and MIPI DSI TX Controller Core Parameters.
July 2021	1.0	Initial release.