MIPI 2.5G DSI TX Controller Core User Guide

UG-CORE-MIPI-2-5G-DSI-TX-v1.4
October 2023
www.elitestek.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 2.5G DSI TX 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
- HS mode byte clock frequency from 5 MHz to 156.25 MHz (80 Mbps to 2,500 Mbps data rate)
- 16-bit HS mode data width
- 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
- Supports initial auto-skew calibration and self-periodic skew calibration
- Supports PPI interface

Device Support

Table 1: MIPI 2.5G DSI TX Controller Core Device Support

FPGA Family	Supported Device
Trion	-
钛金系列	Ti90, Ti120, Ti180 Not including F529 and G529 packages

[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 2.5G DSI TX Controller with 4 data lanes.

FPGA	Logic Element (Logic, Adders, Flip-Flops, etc) [2]	Memory Blocks	DSP Blocks	Efinity® Version[3]
Ti180 M484 C4	119 / 172,800 (0.07%)	5 / 1,280 (0.39%)	0 / 640 (0.00%)	2022.2.322

[2] Using default parameter settings.
[3] Using SystemVerilog.

Installing the Core

The MIPI 2.5G DSI TX Controller core is an Early Access core and it is not included in the Efinity IP Manager. To obtain the core, download the **efx_dsi_tx_top-v<*version*>.zip** file from the Support Center.

The file contains:

File or Folder	Description
source\efx_dsi_tx_top.sv	MIPI 2.5G DSI TX Controller core RTL source file.
\project_new	Contains example design files.
\rtl	Contains support logic for the example design.

To install the MIPI 2.5G DSI TX Controller core:

1. Unzip and copy the files into your project directory.
2. Instantiate the MIPI 2.5G DSI TX Controller core in your top-level wrapper file (e.g. **top.v**).

   Example:

```
module top (
     <ports>
 );

efx_dsi_tx_top_<uuid> #  (
    <parameters>
)  dsi_tx_ctrl_inst  (
    <ports>
);
endmodule
```

   The <uuid> is included in the source\efx_dsi_tx_top.sv file. For example, `define IP_UUID _5a7e90b0930911eda1eb0242ac120002 then the <uuid> is 5a7e90b0930911eda1eb0242ac120002.

   Note: You can also customize the core using parameters in your design file. The core uses the default settings if no parameter is set.

3. Open your project in the Efinity software, click **File > Edit Project**, and in the **Design** tab, add design file and select **source\efx_dsi_tx_top.sv**.

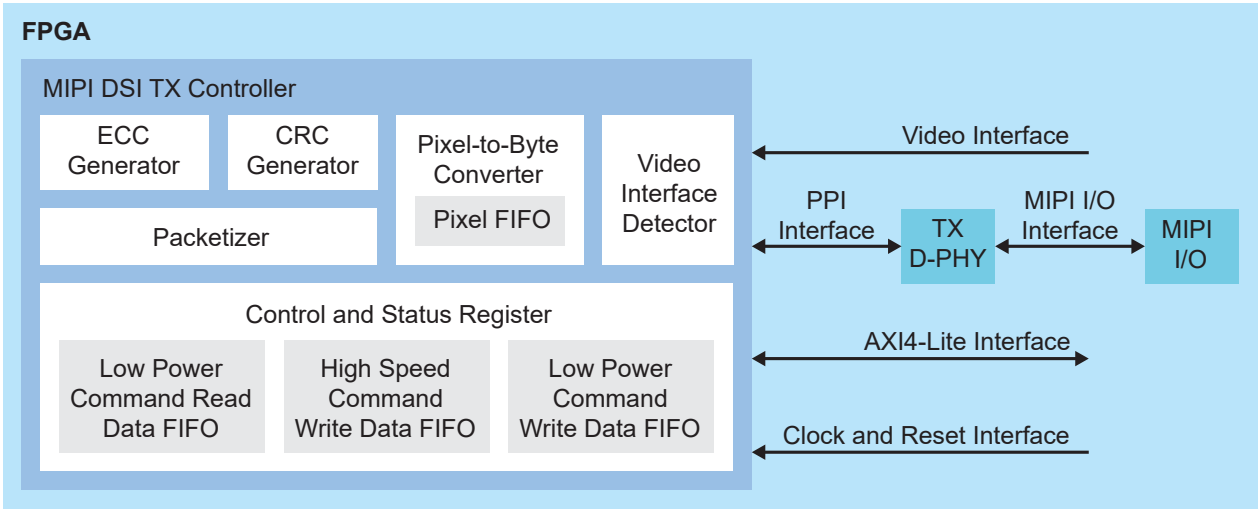
   Note: If required, update the Synopsys Design Constraints (**.sdc**) file before compiling your design.

4. Compile your project.

Functional Description

The MIPI 2.5G DSI TX Controller consists of a control status registers, ECC generator, CRC generator, packetizer, pixel-to-byte converter, and video interface detector. The core has a video interface which is running at pixel clock domain, AXI4-lite interface, clock and reset interface, and PPI interface which is running at MIPI byte clock domain. The DSI Tx controller communicates with the MIPI hard D-PHY in the core device through the PPI interface.

Figure 1: MIPI 2.5G DSI TX Controller System Block Diagram



Ports

Table 3: Clock and Reset Ports

Port	Direction	Description
clk_esc	Input	Transmit Escape mode clock (~20 MHz).
reset_esc_n	Input	Transmit Escape mode reset signal.
clk_byte_HS	Input	MIPI TX 8-bit parallel data clock signal. MIPI data rate/8-bit data width. This clock has to be running at 2 times the phy_clk_byte_HS clock frequency. clk_byte_HS = 2 * phy_clk_byte_HS
reset_byte_HS_n	Input	MIPI TX 8-bit parallel data reset signal.
phy_clk_byte_HS	Input	MIPI TX 16-bit parallel data clock signal. MIPI data rate / 16-bit data width. Supplied by MIPI hard D-PHY. clk_byte_HS = 2 * phy_clk_byte_HS
RxClkEsc	Input	Receive Escape mode clock. Supply by MIPI hard D-PHY.
clk_pixel	Input	Pixel clock.
reset_pixel_n	Input	Pixel reset signal.
axi_clk	Input	AXI4-Lite interface clock.
axi_reset_n	Input	AXI4-Lite interface reset.

Table 4: PHY Protocol interface (PPI)

Port	Direction	Description
TxUlpsClk	Output	Transmit ULPS on MIPI hard D-PHY clock lane. This active high signal is asserted to cause a clock lane to enter the ULPS. It is clocked with clk_esc.
TxUlpsExitClk	Output	Transmit ULPS Exit Sequence on MIPI hard D-PHY clock lane. This active high signal is asserted when ULPS is active and the protocol is ready to leave ULPS. It is clocked with clk_esc.
TxUlpsActiveClkNot	Input	ULPS (not) Active. This active-low signal is asserted to indicate that the MIPI hard D-PHY clock lane is in ULPS. It is clocked with clk_esc.
TxUlpsEsc [NUM_DATA_LANE-1:0]	Output	Transmit ULPS Escape Mode. This active-high signal is asserted with TxRequestEsc to cause the MIPI hard D-PHY data lane to enter the ULPS. It is clocked with clk_esc.
TxUlpsExit [NUM_DATA_LANE-1:0]	Output	Transmit ULPS Exit Sequence. This active-high signal is asserted when ULPS is active and the protocol is ready to leave ULPS on MIPI hard D-PHY data lane. It is clock with clk_esc.
TxUlpsActiveNot [NUM_DATA_LANE-1:0]	Input	ULPS (not) Active. This active-low signal is asserted to indicate that the MIPI hard D-PHY data lane is in ULPS. It is clocked with clk_esc.
TxRequestEsc [NUM_DATA_LANE-1:0]	Output	Transmit Escape Mode Request. This active-high signal is used to request escape sequences on MIPI hard D-PHY data lane. It is clocked with clk_esc.

Port	Direction	Description
TxStopStateD [NUM_DATA_LANE-1:0]	Input	Data lane in Stop State. This is an asynchronous active-high signal from the MIPI hard D-PHY indicates that the data lane is in stop state.
TxStopStateC	Input	Clock lane in Stop State. This is an asynchronous active-high signal from the MIPI hard D-PHY indicates that the clock lane is in stop state.
TxSkewCalHS [NUM_DATA_LANE-1:0]	Output	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 MIPI hard D-PHY to end the transmission of a skew calibration pattern, and initiate an end-of-transmission sequence.
TxReadyHS [NUM_DATA_LANE-1:0]	Input	HS Transmit Ready. This active-high signal indicates that TxDataHS is accepted by the lane module to be serially transmitted.
TxRequestHS [NUM_DATA_LANE-1:0]	Output	HS Transmit Request and Data Valid. A low-to-high transition causes the lane module to initiate a start-oftransmission 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.
TxRequestHSc	Output	HS Transmit Request and Data Valid. A low-to-high transition causes the lane module to initiate a startof-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.
TxDataHS_n [HS_DATA_WIDTH-1:0]	Output	HS data to be transmitted for MIPI hard D-PHY data lane. n = lane 0 to 3
TxReqValidHSn [1:0]	Output	High-Speed Transmit Word Data Valid. When the High-Speed Transmit Data width is greater than 8 bits, it is necessary to indicate which 8-bit segments contain valid transmit data to be able to transmit any number of words. n = lane 0 to 3

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 pixel_data_valid is high. The actual width is dependent on pixel type. See Video Mode Pixel Encoding on page 15.
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 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.
irq	Output	Interrupt signal for Interrupt Status Register.

Table 7: AXI4-Lite Interface

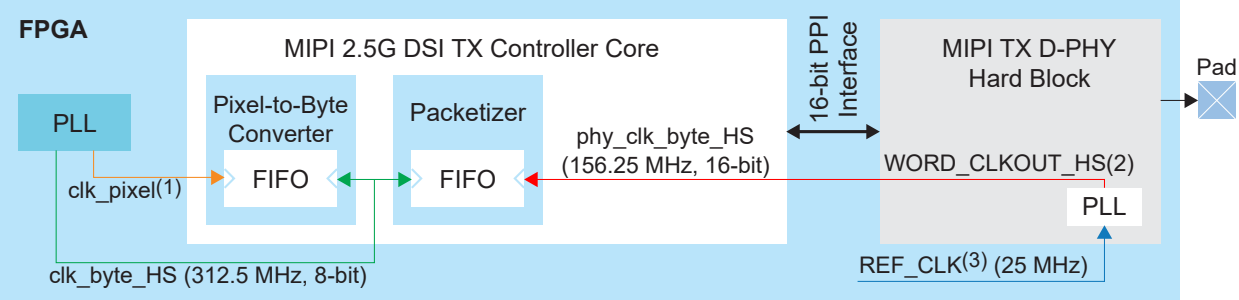
All signals are clocked with axi_clk.

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

Clocking

The following diagram illustrates the example of clock settings for a 2.5 Gbps DSI TX implementation.

Figure 2: 2.5 Gbps DSI TX Clocking Example



(1) Refer to Pixel Clock Calculation section for the clk_pixel value.

(2) **HS Transmit Byte/Word Clock Pin Name** setting in the Interface Designer.

(3) **Reference Clock** setting in the Interface Designer. You can select the source to be from either the core, GPIO, or PLL.

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

Word Offset	Bits	Name	R/W	Width (bits)
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 2.5G 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			
0x40	31:0	Total H line word count in byte.	R/W	32
0x44	15:0	Horizontal sync active (HSA) in byte. Only write to this register when it is sync pulse mode.	R/W	16
0x48	15:0	Horizontal black porch (HBP) in byte. For burst event mode, factor in HSA value into the HBP value.	R/W	16
0x4C	15:0	Horizontal front porch (HFP) in byte.	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.	R/W	8
0x58	7:0	Vertical front porch (VFP) in line. The minimum number of lines is 2.	R/W	8
0x5C	15:0	Vertical active (VACT) in line. The minimum number of lines is 1.	R/W	16

(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).

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.

Video Mode Configuration

The MIPI 2.5G 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`.

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 15.

Video Timing Parameters

Figure 3: Video Timing Waveform (Horizontal)

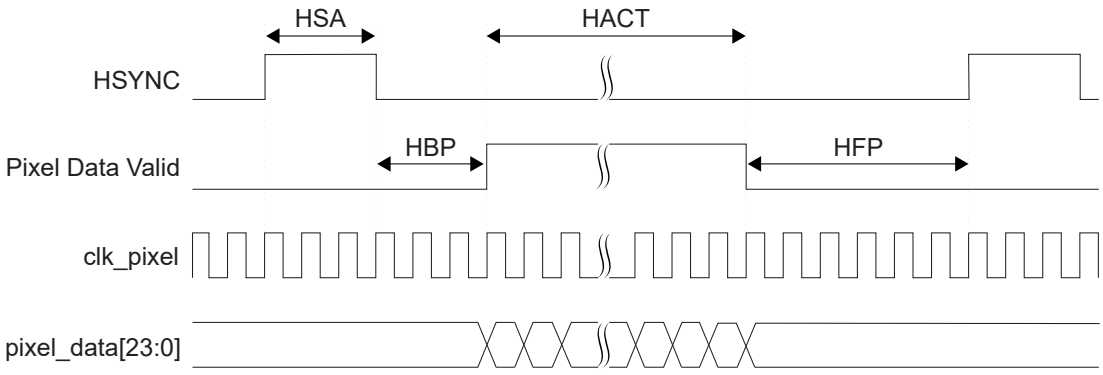


Figure 4: Video Timing Waveform (Vertical)

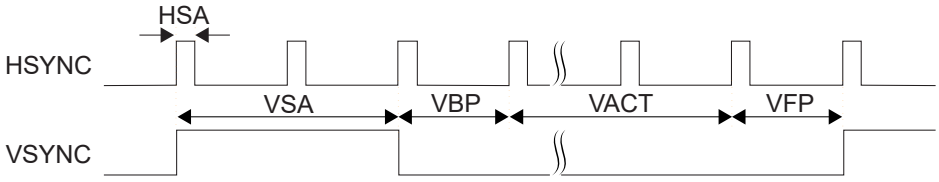


Table 22: MIPI Video Timing Parameters

Parameter	Definition
tLine	Total horizontal line period.
HACT	Total number of actual pixel per line.
VACT	Total number of actual pixel horizontal 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.
MIPI speed	DSI TX MIPI speed in MBPS.
No. data lane	Number of MIPI data lane.

Customizing the MIPI 2.5G DSI TX Controller

Table 23: MIPI 2.5G DSI TX Controller Core Parameter

Name	Option	Description
tINIT_NS	Values according to MIPI D-PHY specifications	Initialization time for the MIPI DSI TX controller in ns. Default: 100,000
tINIT_SKEWCAL_NS	Values according to MIPI D-PHY specifications	Initial skew calibration time for the MIPI DSI TX controller in ns. Default: 100,000 Skew Calibration is carried out after initialization period.
NUM_DATA_LANE	1, 2, 4	Number of data lanes. Default: 4
HS_BYTECLK_MHZ	5 – 156	MIPI parallel clock frequency in MHz to support data rate from 80 Mbps to 2,500 Mbps. Default: 125
DPHY_CLOCK_MODE	Continuous, Discontinuous	D-PHY clock mode. Default: Continuous
PACK_TYPE	2'b00, 2'b01, 2'b10, 2'b11	**PACK_TYPE[0]:** Turn on pack 48-bit datatype, for example, 20-bit YCbCr, 24-bit YCbCr, 12-bit YCbCr, RGB666 (24-bit), or RGB888. 1'b1: Enable pack 48-bit datatype 1'b0: Disable pack 48-bit datatype Default: 2'b11 (Enable) **PACK_TYPE[1]:** Turn on pack 64-bit datatype, for example, 16-bit YCbCr or RGB565. 1'b1: Enable pack 64-bit datatype 1'b0: Disable pack 64-bit datatype Default: 2'b11 (Enable)
ASYNC_STAGE	2 – 8	Cross clock domain control signal synchronization stage. Default: 2
ENABLE_V_LPM_BTA	0, 1	Enables the bus turnaround during the last vertical front porch line which goes into LP-11 state. 1'b1: Enable 1'b0: Disable Default: Disable
MAX_HRES	Values according to video display	Maximum horizontal pixel resolution. Default: 1080

Name	Option	Description
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
HS_CMD_WDATAFIFO_DEPTH	8 – 2048[5]	HS command write data FIFO depth. Default: 512
LP_CMD_WDATAFIFO_DEPTH	8 – 2048[5]	LP command write data FIFO depth. Default: 512
LP_CMD_RDATAFIFO_DEPTH	8 – 2048[5]	Bus turnaround read data depth. Default: 2048
PIXEL_FIFO_DEPTH	256 – 8192[5]	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
HS_LANE_FIFO_DEPTH	8 – 2048[5]	PHY protocol interface high speed data FIFO depth. Default: 512
ENABLE_EOTP	1'b1, 1'b0	Enables or disables the End Of Transmission Packet. 1'b1: Enable 1'b0: Disable (Default)
HS_DATA_WIDTH	16	HS mode data width. Default: 16

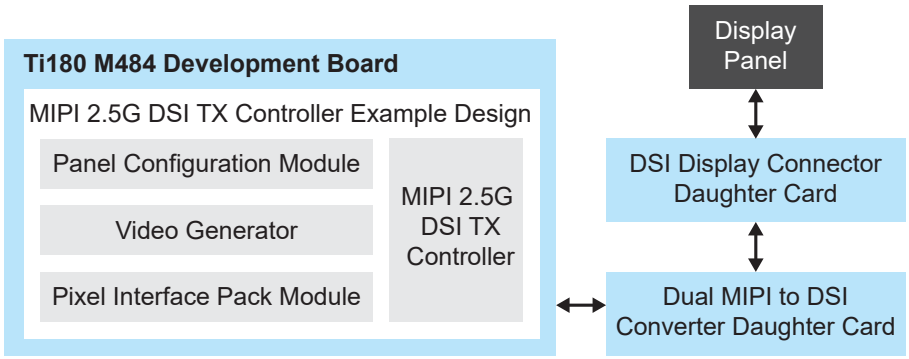
MIPI 2.5G DSI TX Controller Example Design

The example design targets the 钛金系列 Ti180 M484 Development Board. This design generates a video stream and sends the video data to a display panel through the MIPI 2.5G DSI TX Controller. Apart from the 钛金系列 Ti180 M484 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

[5] 2^n, where n can be from 3 to 11.

Figure 5: MIPI 2.5G DSI TX Controller Core Example Design



After power-up, program the example design bit file into the FPGA device and press the reset button (SW3), then you should be able to see a video (color bar) displayed on the panel module.

Figure 6: MIPI 2.5G DSI TX Controller Example Design Design Set Up

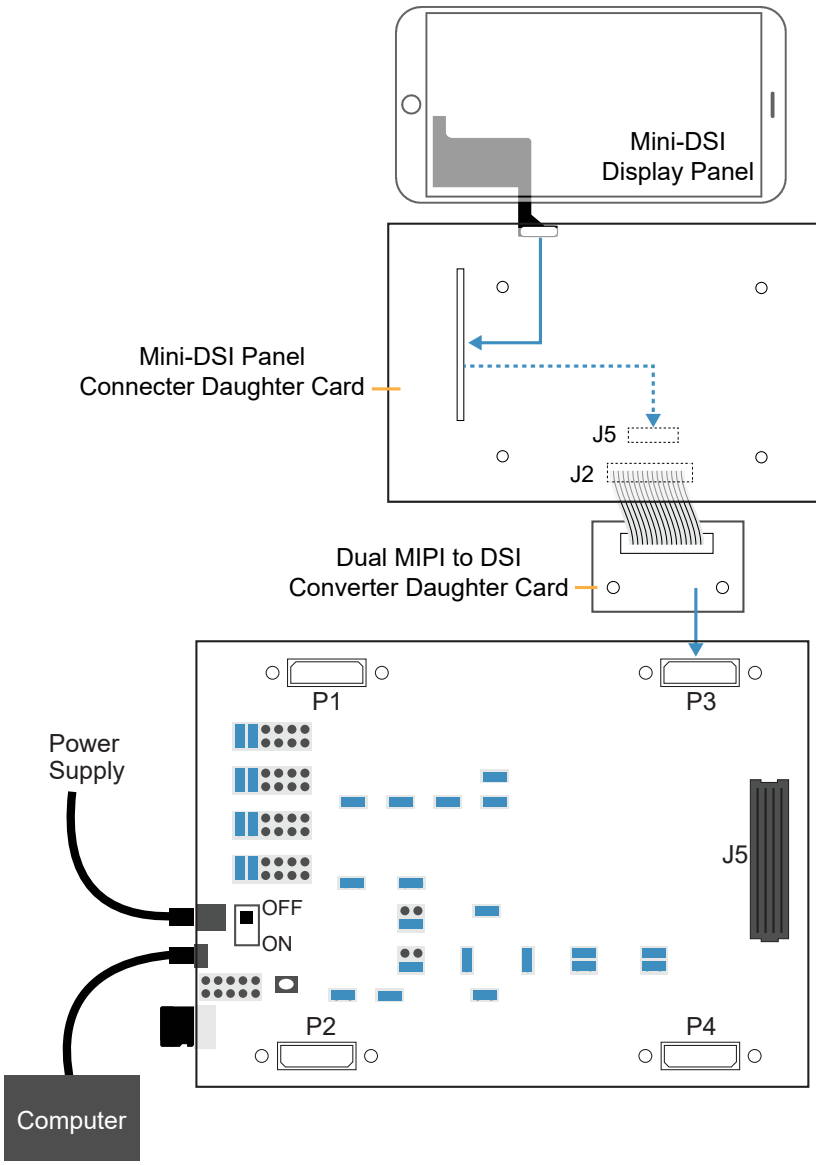


Table 24: Example Design Implementation

FPGA	Logic Element (Logic, Adders, Flip Flops, etc)	Memory Blocks	DSP Blocks	Clock Frequency, f (MHz)				Efinity® Version[6]
				clk_esc	axi_clk	clk_byte_hs	clk_pixel	
Ti180 M484 C4	4,060 / 172,800 (2.35%)	26 / 1,280 (2.03%)	0 / 640 (0.00%)	20	25	125	125	2022.2.322

Revision History

Table 25: Revision History

Date	Version	Description
October 2023	1.4	Removed references to 钛金系列 MIPI Utility. (DOC-1519)
September 2023	1.3	Corrected supported HS data width. (DOC-1437)
June 2023	1.2	Added Device Support section. (DOC-1234) Updated supported data rate. (DOC-1217) Updated FIFO Pixel Depth Size and HS_BYTECLK_MHZ. Editorial changes.
May 2023	1.1	Added MIPI Video Data Formats. (DOC-1233) Improved steps to install the core. (DOC-1244)
February 2023	1.0	Initial release.

[6] Using SystemVerilog.