Parallel to MIPI CSI-2 TX Bridge

Reference Design

FPGA-RD-02132-1.6

January 2020

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

© 2015-2020 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Contents

Figures

Tables

1. Introduction

The Mobile Industry Processor Interface (MIPI) has become a specification standard for interfacing components in consumer mobile devices. The MIPI Camera Serial Interface 2 (CSI-2) specification provides a protocol layer interface definition, which is used to interface with Cameras and Image Sensors. The Parallel to MIPI CSI-2 TX Bridge Reference Design allows users to deliver data to a MIPI CSI-2 compatible receiver such an ISP (Image Signal Processor) from a standard parallel video interface. See Figure 1.1.

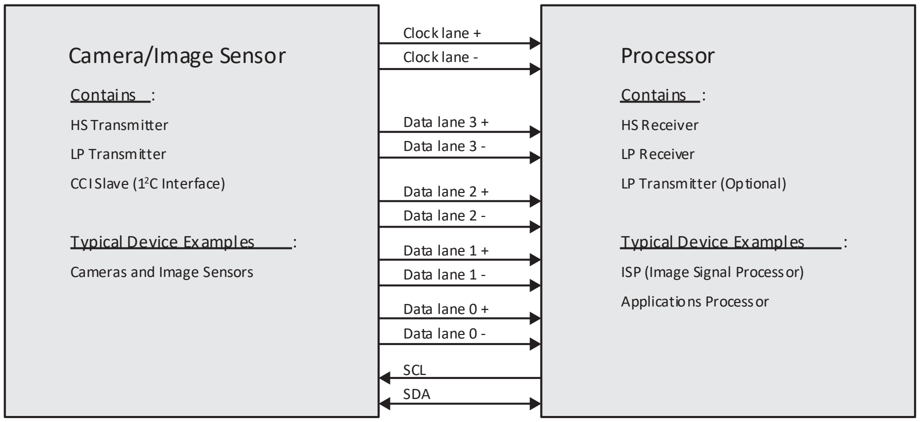


Figure 1.1. CSI-2 Interface

2. Key Features

- Interfaces to MIPI CSI-2 Receiving Devices.
- Supports Unidirectional HS (High Speed) Mode.
- Supports Bidirectional LP (Low Power) Mode.
- Serializes HS (High Speed) data from up to four data lanes.
- Supports all CSI-2 compatible video formats (RAW, YUV, RGB and User Defined).

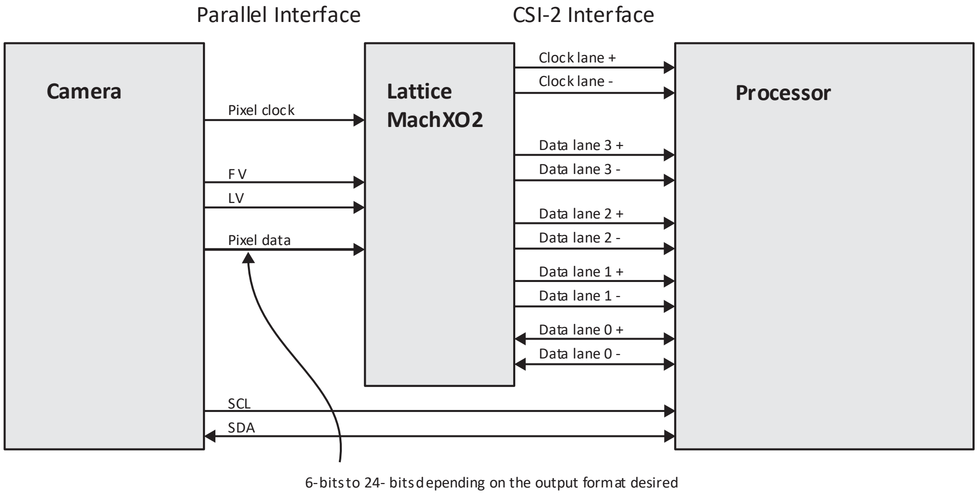


Figure 2.1. Parallel to MIPI CSI-2 TX Bridge

3. Functional Description

The Parallel to MIPI CSI-2 TX Bridge Reference Design converts a standard parallel video interface into CSI-2 byte packets. It then serializes HS data and controls LP (Low Power) and HS (High Speed) data transfers using the Lattice MIPI D-PHY Reference IP (FPGA-RD-02040). The input interface for the design consists of a data bus (PIXDATA), line and frame valid indicators (FV and LV) and a clock (PIXCLK). The output interface consists of HS and LP signals that must be connected together using an external resistor network, which is described in the Unidirectional Transmit HS Mode and Bidirectional LP Mode Interface Implementation section of this document. Further information regarding this resistor network can also be found in the Lattice MIPI D-PHY Reference IP (FPGA-RD-02040) documentation. HS and LP signals for the clock lane and data lanes are provided on DCK, D0, D1, D2, D3 and LPCLK, LP0, LP1, LP2, and LP3 signals respectively. Include parameters control the amount of data ports available for HS and LP modes at the top level depending on the number of data lanes used.

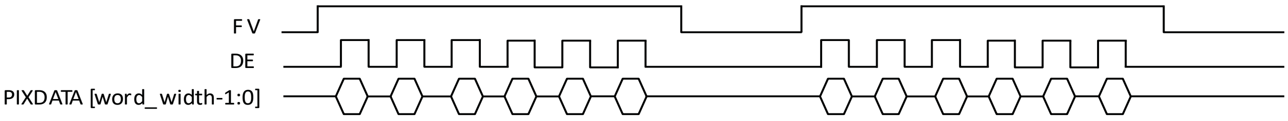


Figure 3.1. Bridge Input Signal Format

The top level design (top.v) consists of five modules:

- byte_packetizer.v – Converts parallel data to byte packets. Appends Packet Header and Checksum.
- lp_hs_dly_ctrl.v – Controls time delay between clock and data lanes when entering and exiting HS mode. Controls time delay from when HS mode is entered to when data is placed on the data bus.
- dphy_tx_inst.v – Serializes byte data using iDDRx4 gearbox primitives. Controls high impedance and bi-directional states of HS and LP signals.
- pll_pix2byte_gen.v – Converts pixel clock to HS clock and byte clocks. Output frequencies depend on input clock, input bus width and number of MIPI data lanes.
- colorbar_gen.v – Pattern generator capable of generating a colorbar or walking 1's pattern.

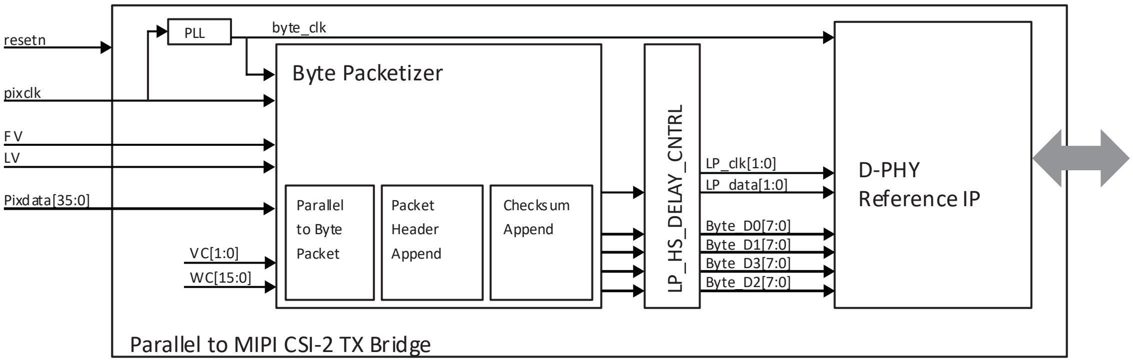


Figure 3.2. Top Level Block Diagram

To control the ports defined at the top level, `define compiler directives are used. These compiler directives can be found in compiler_directives.v.

Table 3.1. Compiler Directives Defined in compiler_directives.v

Directive	Description
`define HS_3	Generates IO for four HS data lanes.
`define HS_2	Generates IO for three HS data lanes. Overridden if HS_3 is defined.
`define HS_1	Generates IO for two HS data lanes Overridden if HS_3 or HS_2 is defined.
`define HS_0	Generates IO for one HS data lane. Overridden if HS_3, HS_2, or HS_1 is defined.
`define LP_CLK	Generates IO for LP mode on clock lane
`define LP_0	Generates IO for LP mode on data lane 0
`define LP_1	Generates IO for LP mode on data lane 1
`define LP_2	Generates IO for LP mode on data lane 2
`define LP_3	Generates IO for LP mode on data lane 3

Design parameters control other features of the design. These design parameters are located at the top of the module declaration in top.v.

Table 3.2. Top Level Module Parameters

Parameter	Options	Description
VC	2-bit Virtual Channel value	Virtual Channel number appended to the Packet Header
WC	16-bit Word Count value	Word Count number appended to the Packet Header. Correlates to the number of bytes to be transferred in a Long Packet.
word_width	Up to 36 bits	Bus width of pixel data bus input
DT	6-bit Data Type Value	Data Type appended to Packet Header for Long Packet transfers
testmode	0 = off 1 = on	Adds colorbar pattern generator for testing purposes. Pattern generator utilizes reset_n and PIXCLK.
crc16	0 = off 1 = on	Appends checksum after Long Packet transfers. Turning off will reduce resource utilization and append 16'hFFFF in place of checksum.

Top level IO ports are defined as follows for top.v. The number of IO is dependent on the number of data lanes defined by compiler_directives.v.

Table 3.3. Top Level Design Port List

Signal	Direction	Description
reset_n	Input	Resets module (Active ‘low’)
DCK	Output	HS (High Speed) Clock
D0	Output	HS Data lane 0
D1	Output	HS Data lane 1
D2	Output	HS Data lane 2
D3	Output	HS Data lane 3
LPCLK [1:0]	Bidirectional	LP clock lane; LPCLK[1] = P wire, LPCLK[0] = N wire
LP0 [1:0]	Bidirectional	LP data lane 0; LP0[1] = P wire, LP0[0] = N wire
LP1 [1:0]	Bidirectional	LP data lane 1; LP1[1] = P wire, LP1[0] = N wire
LP2 [1:0]	Bidirectional	LP data lane 2; LP2[1] = P wire, LP2[0] = N wire
LP3 [1:0]	Bidirectional	LP data lane 3; LP3[1] = P wire, LP3[0] = N wire
PIXCLK	Input	Parallel Pixel Clock
FV	Input	Parallel Data Frame Valid Indicator
LV	Input	Parallel Data Line Valid Indicator
PIXDATA[*:0]	Input	Parallel Data Bus

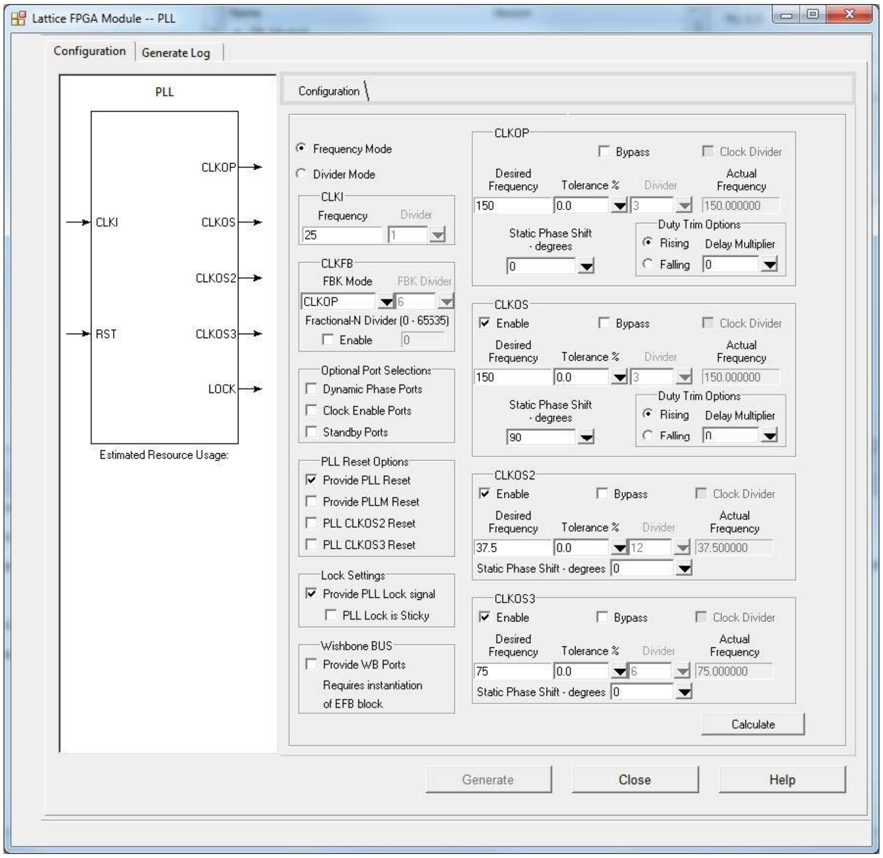
The top level module instantiates and connects five main modules. In addition, a PLL module controls clocking for the entire design. The input of the PLL is pixel clock. The PLL outputs two high speed oDDRx4 gearbox clocks (0 degree and one with 90 degree phase shifts), the byte clock and the CRC clock.

The clock equations for PLL output ports are shown in Table 3.4.

Table 3.4. Clocking for the PLL Output Ports

PLL Module Port Name	Clock description	Clock Equation
CLKI	PLL Input	CLKI
CLKOP	oDDRx4 gearbox Clock	CLKOP = CLKI * word_width / (8 * lane_width) *4
CLKOS	oDDRx4 gearbox Clock (90 degree shift)	Same as CLKOP, but with static phase shift of 90 degrees
CLKOS2	Byte Clock	CLKOS2 = CLKI * word_width / (8 * lane_width)

The PLL is configured using IPExpress in the Lattice Diamond® Software. The PLL comes pre-configured for the appropriate clock conversion ratios based on the mode and number of MIPI data lanes used. It can also be adjusted and reconfigured to individual design needs by double clicking on pll_pix2byte.ipx in the file list. An IPExpress configuration GUI will open to adjust the PLL.



● Frequency Mode
○ Divider Mode

CLKI
Frequency: 25
Divider: 1

CLKFB
FBK Mode: CLKOP
FBK Divider: 6
Fractional-N Divider (0 - 65535)
☐ Enable 0

Optional Port Selections
☐ Dynamic Phase Ports
☐ Clock Enable Ports
☐ Standby Ports

PLL Reset Options
☑ Provide PLL Reset
☐ Provide PLLM Reset
☐ PLL CLKOS2 Reset
☐ PLL CLKOS3 Reset

Lock Settings
☑ Provide PLL Lock signal
☐ PLL Lock is Sticky

Wishbone BUS
☐ Provide WB Ports
Requires instantiation of EFB block

CLKOP
☐ Bypass
☐ Clock Divider
Desired Frequency: 150
Tolerance %: 0.0
Divider: 3
Actual Frequency: 150.000000
Static Phase Shift - degrees: 0
Duty Trim Options
● Rising
○ Falling
Delay Multiplier: 0

CLKOS
☑ Enable
☐ Bypass
☐ Clock Divider
Desired Frequency: 150
Tolerance %: 0.0
Divider: 3
Actual Frequency: 150.000000
Static Phase Shift - degrees: 90
Duty Trim Options
● Rising
○ Falling
Delay Multiplier: 0

CLKOS2
☑ Enable
☐ Bypass
☐ Clock Divider
Desired Frequency: 37.5
Tolerance %: 0.0
Divider: 12
Actual Frequency: 37.500000
Static Phase Shift - degrees: 0

CLKOS3
☑ Enable
☐ Bypass
☐ Clock Divider
Desired Frequency: 75
Tolerance %: 0.0
Divider: 6
Actual Frequency: 75.000000
Static Phase Shift - degrees: 0

Figure 3.3. IPExpress Configuration Page for pll_pix2byte.ipx:

4. BYTE_PACKETIZER Module Description:

The Byte_Packetizer module converts pixels to one to four bytes depending on the number of MIPI data lanes defined. The input interface to the module is the pixel data bus. Pixel data is formatted from MSB to LSB for RAW and YUV data types. The RGB data type is formatted as R, then G, then B from MSB to LSB respectively. Additional, input ports include the byte clock and CRC clock. The virtual channel number and word count are also available as interface ports. These ports are clocked into a register on the rising edge and falling edges of FV as well as the rising edge of LV and appended to the Packet Header. By default, the reference design controls the Virtual Channel and Word Count ports through VC and WC parameters in top.v. However, the user can dynamically control VC and WC if desired.

Parameters for the BYTE_PACKETIZER include word_width (bus width of the pixel bus), lane_width (number of byte lanes), dt (data type), crc16 enable. Different NGOs in the */NGO/* folder are called depending on the mode defined.

Within the module the pixel data is converted to bytes. If the data is going to be a long packet, identified by LV (Line Valid), the CRC checksum will be calculated over the data and appended to the end of the long packet. Also appended to the data stream in this module is the Packet Header for all packet types.

The number of horizontal pixels the LV (Line Valid) is high should correlate to an integer multiple of the number of bytes used at the output. It is recommend that active lines be truncated or extended to meet this criteria. This will ensure proper readout of all pixels and a correct checksum calculation.

To ensure that the input pixel data is an integer multiple of the output byte data, the following equation can be used. The LV must be held for an integer number of byte clocks. If "number of byte clocks" does not calculate to an integer value, adjust the number of pixel clock cycles for which LV is active.

number of byte clocks = [(number of pixels) * (bits per pixel)] / [8 bits * (number of data lanes)]

Output ports for the BYTE_PACKETIZER module include the 8-bit data buses for each lane and an enable signal. The hs_en signal goes active ‘high’ when any short packet or long packet is to be transmitted.

5. LP_HS_DELAY_CNTRL Module Description:

The LP_HS_DELAY_CNTRL module uses the hs_en input from the BYTE_PACKETIZER and adds delays so that it is ready for transmission. There are controllable delay parameters available in the module header. These control the time delay between when the clock lane and data lanes transition from LP to HS mode as well as from HS mode to LP mode. It also controls when the data starts with respect to when it entered LP mode. LP11-LP01-LP00 transitions are also controlled with one byte clock between transitions. This module is open source and available for any user modifications desired in Lattice FPGA devices.

Table 5.1. LP_HS_DELAY_CNTRL Module Parameters

Parameter	Description
LPHS_clk2data_dly	Number of clocks to delay between the MIPI clock lane and MIPI data lanes transitioning from LP to HS mode
LPHS_startofdata_dly	Number of clocks to delay the MIPI data from the LP to HS mode transition
HSLP_data2clk_dly	Number of clocks to delay the HS to LP mode transition between the MIPI data lanes and MIPI clock lane
HSLP_endofdata_dly	Number of clocks to delay the MIPI data from the HS to LP mode transition
sizeofstartcntr	Size for the start timer counter. Number of bits to count LPHS_clk2data_dly+LPHS_startofdata_dly
sizeofendcntr	Size for the end timer counter. Number of bits to count HSLP_data2clk_dly+HSLP_endofdata_dly

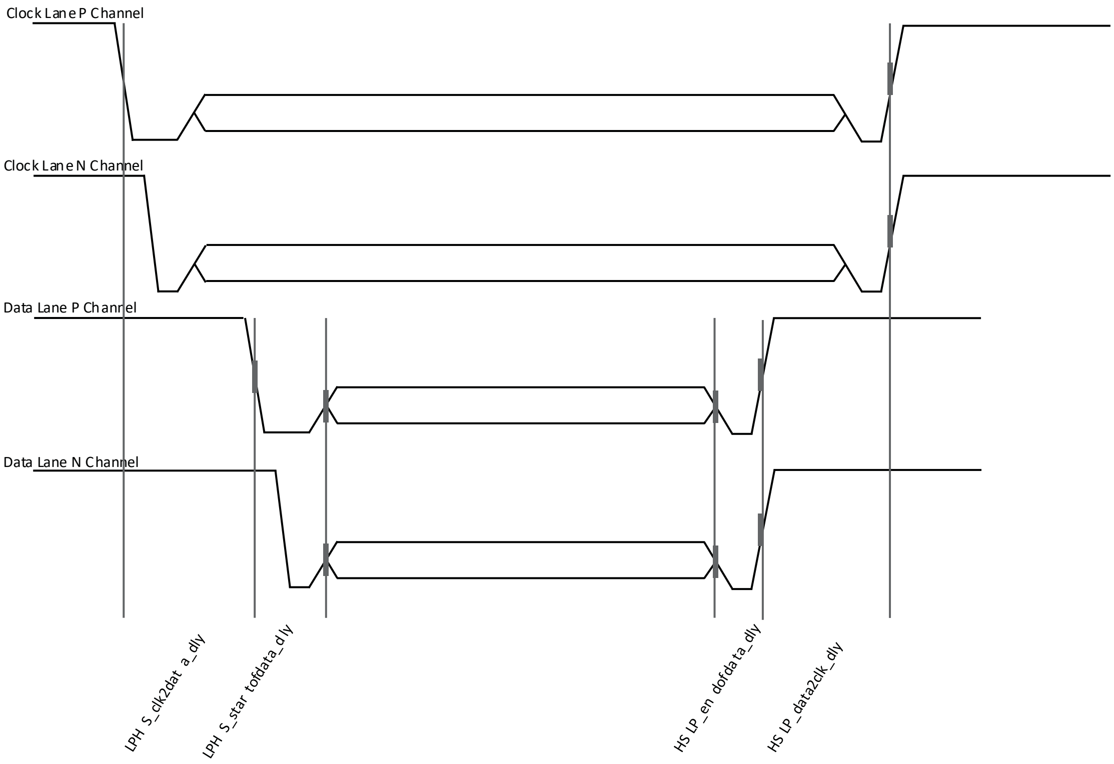


Figure 5.1. Timing Diagram for LP_HS_DELAY_CNTRL Delay Parameters

6. Test Mode and colorbar_gen Module Description:

This reference design also includes a colorbar pattern generator. This allows the user to initially control and drive a display panel with minimal external controls needed from the receiving side. The design is place in test mode setting the top level design parameter testmode = 1. When this is set an additional module colorbar_gen is instantiated at the top level and takes over the all input controls (FV, LV and PIXDATA) with the exception of reset_n and PIXCLK.

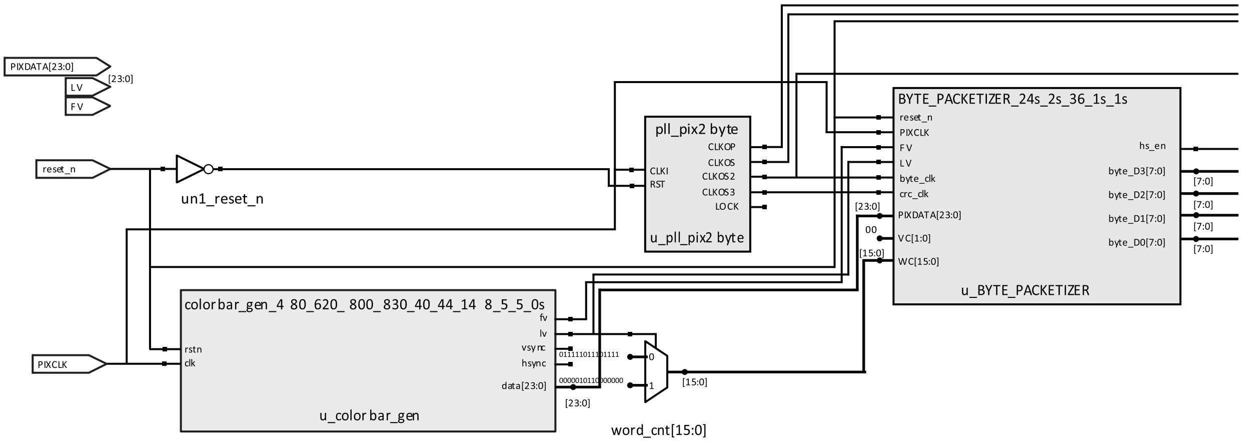


Figure 6.1. RTL Block Diagram of colorbar_gen Instantiation When testmode=1:

7. Packaged Design

The Parallel to MIPI CSI-2 TX Bridge Reference Design is available for Lattice MachXO2TM devices. The reference design immediately available on latticesemi.com is configured for RAW10, 2-lane mode. Other designs are available through the bridge request form. The packaged design contains a Lattice Diamond project within the *\impl\ folder configured for the MachXO2 device. Verilog source is contained within the *\rtl\ folder. The Verilog test bench is contained within the tb folder. The simulation folder contains an Aldec Active-HDL project. It is recommended that users access the active HDL Simulation environment through the Lattice Diamond Software and the simulation setup script contained within the project. For details on how to access the design simulation environment see the Functional Simulation section of this document.

```
◢ rd1183
    docs
  ◢ project
    ◢ ecp5
        verilog
    ◢ xo2
        verilog
    ◢ xo3l
        verilog
  ◢ simulation
    ◢ ecp5
      ▷ verilog
    ◢ xo2
      ▷ verilog
    ◢ xo3l
      ▷ verilog
  ◢ source
    ◢ verilog
      ◢ ecp5
        ▷ clarity
          ngo
      ◢ xo2
          ipexpress
          ngo
      ◢ xo3l
          ipexpress
          ngo
  ◢ testbench
      verilog
```

Figure 7.1. Packaged Design Directory Structure

8. Functional Simulation

The simulation environment and testbench Parallel2CSI2_tb_*.v instantiates the top level design module. The top level design inputs are driven with a generated pattern from the colorbar_gen module.

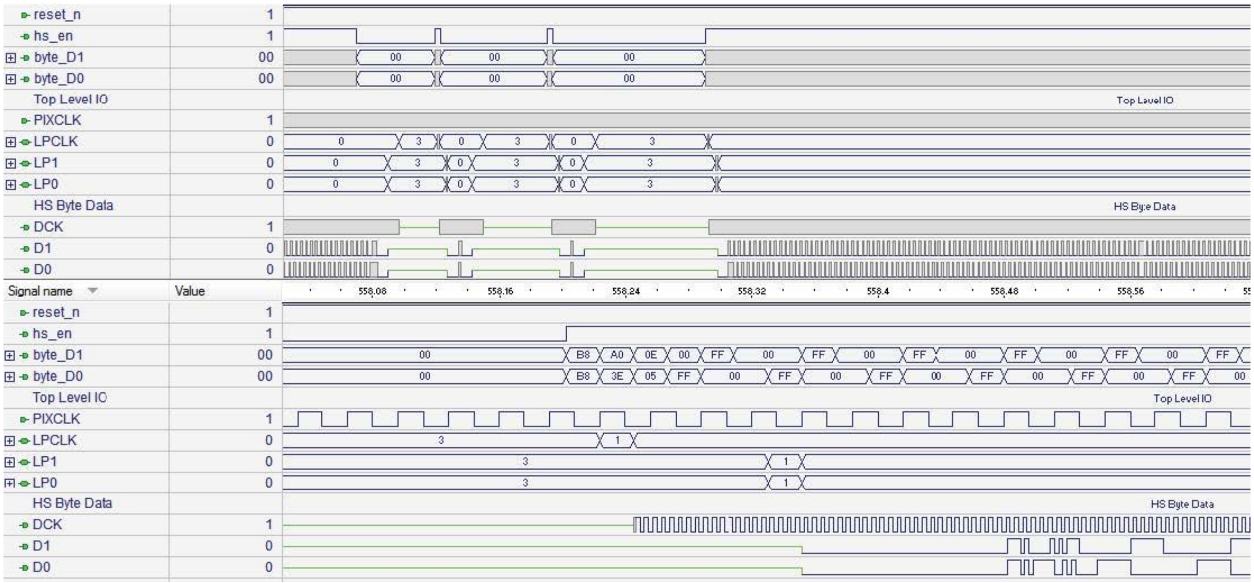


Figure 8.1. Simulation Waveforms

The simulation environment can be accessed by double clicking on the <name>.spf script file in Lattice Diamond from the file list. After clicking OK, Aldec ActiveHDL opens to the pop-up windows. Compile the project and initialize the simulation. Add signals to the waveform viewer that are desired to be viewer and run the simulation.

9. External Resistor Network Implementation for D-PHY TX

As described in the Lattice MIPI D-PHY Reference IP (FPGA-RD-02040) documentation, an external resistor network is needed to accommodate the LP and HS mode transitioning on the same signal pairs as well as the lower 200 mV common mode voltage during HS clock and data transfers. The resistor network needed for MIPI TX implementations is provided below.

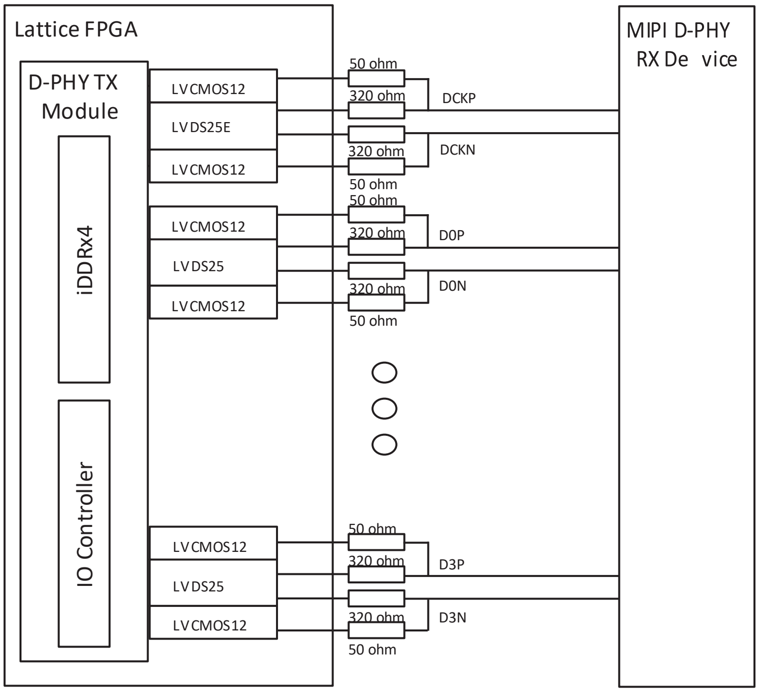


Figure 9.1. Unidirectional Transmit HS Mode and Bidirectional LP Mode Interface Implementation

10. Device Pinout and Bank Voltage Requirements

Choosing a proper pinout to interface with another D-PHY device is essential to meet functional and timing requirements.

The following are rules for choosing a proper pinout on MachXO2 devices:

Bank 0 should be used for HS outputs (DCK, D0, D1, D2, D3) with the TX D-PHY IP since these pins utilize oDDRx4 gearbox primitives.

- The VCCIO voltage for banks 0 should be 2.5 V.
- The HS input clock (DCK) for the RX DPHY IP should use an edge clock on bank 2.
- The HS data signals (D0, D1, D2, D3) for the RX and TX DPHY IP's should only use A/B IO pairs.
- LP signals (LPCLK, LP0, LP1, LP2, LP3) for RX and TX DPHY IP's can use any other bank.
- The VCCIO voltage for the bank containing LP signals (LPCLK, LP0, LP1, LP2, LP3) should be 1.2 V.
- When in doubt, run the pinout through Lattice Diamond software can check for errors.

With the rules mentioned above a recommend pinout is provided for the most common packages chosen for this IP. For the MachXO2 the cs132bga is the most common package. The pinouts chosen below are pin compatible with MachXO2-1200, MachXO2-2000 and MachXO2-4000 devices.

Table 10.1. Recommended TX Pinout and Package

Signal	MachXO2 1200/2000/4000 cs132bga Package	
DCK_p	Bank 0	A7
DCK_n		B7
D0_p		B5
D0_n		C6
D1_p		A2
D1_n		B3
D2_p		A10
D2_n		C11
D3_p		C12
D3_n		A12
LPCLK [1]	Bank 1	E12
LPCLK [0]		E14
LP0 [1]		E13
LP0 [0]		F12
LP1 [1]		F13
LP1 [0]		F14
LP2 [1]		G12
LP2 [0]		G14
LP3 [1]		G13
LP3 [0]		H12

Table 10.2. TX IO Timing

Device Family	Speed Grade -4 @ 262Mhz		Speed Grade -5 @ 315Mhz		Speed Grade -6 @ 378Mhz	
MachXO2^{TM}	Data Valid Before Clock (ps)	Data Valid After Clock (ps)	Data Valid Before Clock (ps)	Data Valid After Clock (ps)	Data Valid Before Clock (ps)	Data Valid After Clock (ps)
	710	710	570	570	455	455

Table 10.3. TX Maximum Operating Frequencies by Configuration

Device Family	Configuration	Speed Grade -4 (MHz)		Speed Grade -5 (MHz)		Speed Grade -6 (MHz)		Speed Grade -7 (MHz)		Speed Grade -8 (MHz)	
		PIXCLK	byte_clk	PIXCLK	byte_clk	PIXCLK	byte_clk	PIXCLK	byte_clk	PIXCLK	byte_clk
ECP5^{TM}	RAW10, 2 Data Lane (LP+HS)- LSE	—	—	—	—	205.888	127.097	244.439	152.23	267.451	187.547
	RAW10, 2 Data Lanes (LP+HS)- Syn	—	—	—	—	220.459	135.722	258.732	149.903	291.29	140.905
MachXO2	RAW10, 1 Data Lane (LP+HS)	150	97.9	164.7	110	182.5	103.3	164.7	110	182.5	103.3
	RAW10, 2 Data Lanes (LP+HS)	150	92.5	164.7	98.367	182.5	109.158	164.7	98.367	182.5	109.158
	RAW10, 2 Data Lanes (LP+HS) - LSE	150.015	97.704	164.69	97.733	182.5	107.365	164.69	97.733	182.5	107.365
	RAW10, 4 Data Lanes (LP+HS)	150	94.9	164.7	100	182.5	113.2	164.7	100	182.5	113.2
MachXO3L	RAW10, 2 Data Lanes (LP+HS) - LSE	—	—	164.69	106.326	182.548	99.118	164.69	106.326	182.548	99.118
	RAW10, 2 Data Lanes (LP+HS) - Syn	—	—	164.69	102.881	182.548	108.542	164.69	102.881	182.548	108.542

Note: The maximum operating frequencies were obtained by post P&R timing analysis. They do not correlate to clocking ratios (obtained from PLL clock equations) used for proper design operation.

11. Resource Utilization

The resource utilization tables below represent the device usage in various configurations of the D-PHY IP. Resource utilization was performed on the IP in configurations of 1, 2 and 4 data lanes. For each of these configurations LP mode on the data lanes used was turned on. In addition, HS and LP clock signals were available for each configuration.

Table 11.1. TX Resource Utilization

Device Family	Configuration	Register	LUT	EBR	PLL	Gearbox	Clock Divider
ECP5	RAW10, 2 Data Lanes (LP+HS) - LSE	627	1369	3	2	3	1
	RAW10, 2 Data Lanes (LP+HS) - Syn	542	858	3	2	3	1
MachXO2	RAW10, 1 Data Lanes (LP+HS)	217	299	3	1	2	1
	RAW10, 2 Data Lanes (LP+HS)	272	394	2	1	3	1
	RAW10, 4 Data Lanes (LP+HS)	345	467	2	1	1	1
MachXO3L	RAW10, 2 Data Lanes (LP+HS) - LSE	447	383	2	1	3	1
	RAW10, 2 Data Lanes (LP+HS) - Syn	272	394	2	1	3	1

References

- MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2) V1.01
- MIPI Alliance Specification for D-PHY V1.1

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.6, January 2020

Section	Change Summary
All	• Changed document number from RD1183 to FPGA-RD-02132. • Updated document template.
Disclaimers	Added this section.

Revision 1.5, January 2015

Section	Change Summary
Resource Utilization	• Added support for ECP5 device family. • Updated the Resource Utilization section. Updated Table 11.1, TX Resource Utilization.
Packaged Design	Updated the Packaged Design section. Updated Figure 7.1, Packaged Design Directory Structure.
Device Pinout and Bank Voltage Requirements	Updated the Device Pinout and Bank Voltage Requirements section. Updated Table 10.3, TX Maximum Operating Frequencies by Configuration.
All	Corrected version number on first page footer. Previous version should be 01.4; updated this version to 01.5.

Revision 1.4, April 2014

Section	Change Summary
Resource Utilization	Added support for MachXO3L device family.
Functional Description	Updated Functional Description section. Revised top level design (top.v) main modules.
Functional Simulation	Updated Functional Simulation section. Revised .spf script file name.
Packaged Design	Updated Packaged Design section. Updated Figure 7.1, Packaged Design Directory Structure.
Device Pinout and Bank Voltage Requirements	Updated the Device Pinout and Bank Voltage Requirements section. Updated Table 10.3, TX Maximum Operating Frequencies by Configuration.
Resource Utilization	Updated the Resource Utilization section. Updated Table 9, TX Resource Utilization. Added support for Lattice Diamond 3.1 design software.e

Revision 1.3, March 2014

Section	Change Summary
LP_HS_DELAY_CNTRL Module Description:	Updated Figure 5.1, Timing Diagram for LP_HS_DELAY_CNTRL Delay Parameters.

Revision 1.2, December 2013

Section	Change Summary
BYTE_PACKETIZER Module Description	Updated the BYTE_PACKETIZER Module Description section.

Revision 1.1, August 2013

Section	Change Summary
Device Pinout and Bank Voltage Requirements	Updated Table 10.3, title to TX Maximum Operating Frequencies by Configuration and added footnote.

Revision 1.0, August 2013

Section	Change Summary
All	Initial release.

www.latticesemi.com