OpenLDI/FPD-LINK/LVDS Transmitter IP Core - Lattice Radiant Software

User Guide

FPGA-IPUG-02117-1.0

August 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.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
FPD	Flat Panel Display
LVDS	Low Voltage Differential Signaling
LDI	LVDS Display Interface
RTL	Register Transfer Language

1. Introduction

The Lattice Semiconductor OpenLDI/FPD-LINK/LVDS Transmitter Interface IP translates parallel video streams to an LVDS (Low Voltage Differential Signaling) interface for an FPD-Link (Flat Panel Display Link) connection to a display.

The increasing demand for better display technology makes bridging applications popular. FPD-Link is a common application interface. Similar to Channel Link and Camera Link, FPD-Link also uses LVDS interface for the physical layer.

The LVDS standard is commonly used in high-speed differential interface among consumer device, industrial control, medical, and automotive applications. It offers low voltage, low power, and improved signal integrity which are advantages over single-ended technology.

The 7:1 LVDS interface is a popular standard for source asynchronous interfaces, which consist of multiple data bits and clocks. Typically, one channel of 7:1 LVDS interface consists of five LVDS pairs (one clock and four data) depending on the data type it supports.

This document describes the use of the OpenLDI/FPD-LINK/LVDS Transmitter Interface IP and Lattice FPGA technology for LVDS interface applications. The design, which can be applied in multiple configurations, is implemented in Verilog HDL. It can be targeted to CrossLink™-NX and Certus™-NX FPGA devices and implemented using the Lattice Radiant® software Place and Route tool integrated with the Synplify Pro® synthesis tool.

1.1. Quick Facts

Table 1.1 presents a summary of the OpenLDI/FPD-LINK/LVDS Transmitter IP.

Table 1.1. OpenLDI/FPD-LINK/LVDS Transmitter IP Quick Facts

IP Requirements	Supported FPGA Families	CrossLink-NX, Certus-NX
Resource Utilization	Targeted Devices	LIFCL-40, LIFCL-17, LFD2NX-40
	Supported User Interface	Parallel to Native LVDS Interface
	Resources	See Table A.1. Resource Utilization
Design Tool Support	Lattice Implementation	IP Core v1.0.0 – Lattice Radiant software 2.1
	Synthesis	Lattice Synthesis Engine
		Synopsys® Synplify Pro for Lattice
	Simulation	For a list of supported simulators, see the Lattice Radiant Software 2.1 User Guide

1.2. Features

The key features of the OpenLDI/FPD-LINK/LVDS Transmitter IP include:

- Compliant with Open LVDS Display Interface (OpenLDI) v0.95 specifications
- Transmits in OpenLDI unbalanced operating mode format
- Supports RGB888 and RGB666 video formats
- Supports transmitting in Dual Channel Flat Panel Display Link Protocol (7:1 LVDS)
- Supports three to four LVDS data lanes per channel
- Supports 1 or 2 input pixel data per pixel clock
- Supports interfacing up to 7.560 Gb/s

1.3. Conventions

1.3.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.3.2. Signal Names

Signal names that end with:

- *_n* are active low (asserted when value is logic 0)
- *_i* are input signals
- *_o* are output signals

1.3.3. Data Ordering and Data Types

- The most significant bit within the pixel data is the highest index.
- Pixel data order before distribution to LVDS lanes is {Red[MSB:0], Green[MSB:0], Blue[MSB:0]}. One or two pixels may be sent for distribution to LVDS lanes in one pixel clock cycle, depending on the number of Tx channels setting. If there are multiple pixels per clock cycle, the pixel in the lower bits is the first pixel received. For instance, the pixel order for two pixels per clock is {pixel1, pixel0}, where pixel0 is received first and pixel1 is received last.
- Pixel data is transmitted over LVDS lanes according to OpenLDI 18-bit and 24-bit unbalanced operating mode formats.

1.3.4. Attribute Names

Attribute names in this document are formatted in title case and italicized (*Attribute Name*).

2. Functional Descriptions

2.1. Overview

The OpenLDI/FPD-LINK/LVDS Transmitter Interface IP converts pixel data into a standard OpenLDI serial video interface domain. The input interface for the design consists of the RGB control signals, pixel clock, and up to two pixel data per pixel clock. Output interface consists of a data bus, vertical and horizontal sync flags, a data enable and a clock in OpenLDI (LVDS7:1) interface format, and optional debug signal/s.

The functional block diagram of OpenLDI/FPD-LINK/LVDS Transmitter IP Core is shown in Figure 2.1. The dashed lines in the figure are optional components/signals, which means they may not be available in the IP when disabled in the attribute.

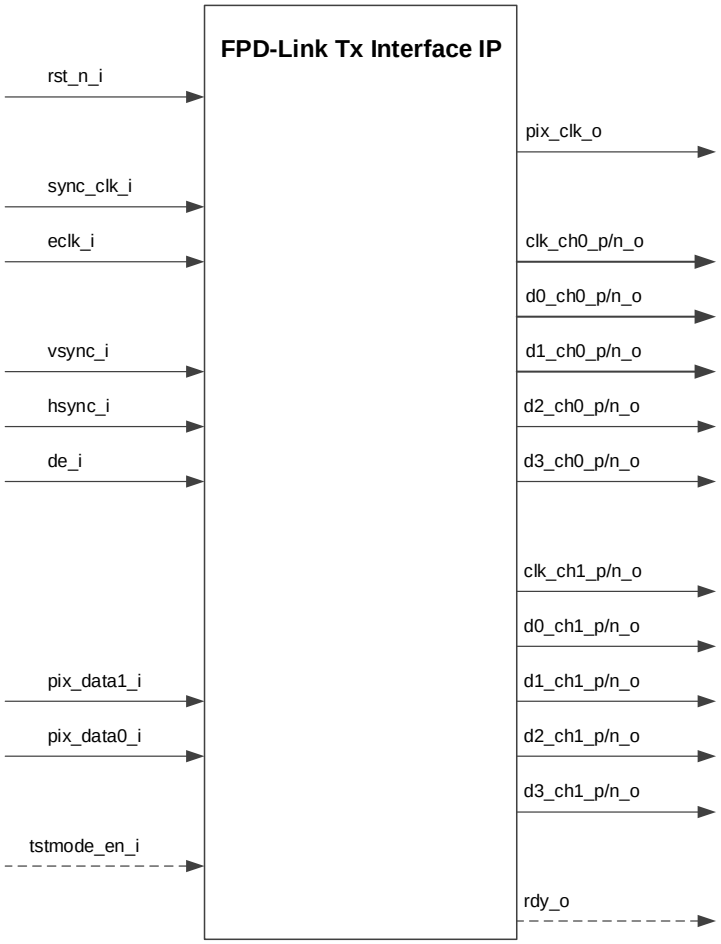

Figure 2.1. Functional Block Diagram

2.2. Signal Description

Table 2.1. OpenLDI/FPD-LINK/LVDS Transmitter IP Core Signal Description

Port Name	I/O	Width	Description
Clock and Reset			
rst_n_i	In	1	Asynchronous active low system reset 0 – System on reset
sync_clk_i	In	1	Clock for GDDR_SYNC Must be slower than all the clocks in the system.
eclk_i	In	1	Edge clock for LVDS side
Pixel Domain Interface			
pix_clk_o	Out	1	Output pixel clock.
pix_data0_i	In	*Number of TX Lanes* * 6	Input pixel data 0 Bus width depends on the data type selected.
pix_data1_i[1,2]	In	*Number of TX Lanes* * 6	Input pixel data 1 Bus width depends on the data type selected.
de_i	In	1	Input data enable for parallel interface
hsync_i	In	1	Input horizontal sync for parallel interface
vsync_i	In	1	Input vertical sync for parallel interface
OpenLDI/FPD-LINK Interface			
clk_ch0_p_o	Out	1	Positive LVDS output clock to LVDS7:1 Tx channel 0
clk_ch0_n_o	Out	1	Negative LVDS output clock to LVDS7:1 Tx channel 0 Complement of clk_ch0_p_o Not available in netlist because this is automatically set to complement by the synthesis tool.
clk_ch1_p_o	Out	1	Positive LVDS output clock to LVDS7:1 Tx channel 1
clk_ch1_n_o	Out	1	Negative LVDS output clock to LVDS7:1 Tx channel 1 Complement of clk_ch1_p_o Not available in netlist because this is automatically set to complement by the synthesis tool.
d0_ch0_p_o	Out	1	Positive LVDS output data lane 0 to LVDS7:1 Tx channel 0
d0_ch0_n_o	Out	1	Negative LVDS output data lane 0 to LVDS7:1 Tx channel 0 Complement of d0_ch0_p_o Not available in netlist because this is automatically set to complement by the synthesis tool.
d1_ch0_p_o	Out	1	Positive LVDS output data lane 1 to LVDS7:1 Tx channel 0
d1_ch0_n_o	Out	1	Negative LVDS output data lane 1 to LVDS7:1 Tx channel 0 Complement of d1_ch0_p_o. Not available in netlist because this is automatically set to complement by the synthesis tool.
d2_ch0_p_o	Out	1	Positive LVDS output data lane 2 to LVDS7:1 Tx channel 0.
d2_ch0_n_o	Out	1	Negative LVDS output data lane 2 to LVDS7:1 Tx channel 0 Complement of d2_ch0_p_o Not available in netlist because this is automatically set to complement by the synthesis tool.
d3_ch0_p_o[3]	Out	1	Positive LVDS output data lane 3 to LVDS7:1 Tx channel 0
d3_ch0_n_o[3]	Out	1	Negative LVDS output data lane 3 to LVDS7:1 Tx channel 0 Complement of d3_ch0_p_o Not available in netlist because this is automatically set to complement by the synthesis tool.
d0_ch1_p_o[4]	Out	1	Positive LVDS output data lane 0 to LVDS7:1 Tx channel 1

Port Name	I/O	Width	Description
d0_ch1_n_o[4]	Out	1	Negative LVDS output data lane 0 to LVDS7:1 Tx channel 1 Complement of d0_ch1_p_o Not available in netlist because this is automatically set to complement by the synthesis tool.
d1_ch1_p_o[4]	Out	1	Positive LVDS output data lane 1 to LVDS7:1 Tx channel 1
d1_ch1_n_o[4]	Out	1	Negative LVDS output data lane 1 to LVDS7:1 Tx channel 1 Complement of d1_ch1_p_o Not available in netlist because this is automatically set to complement by the synthesis tool.
d2_ch1_p_o[4]	Out	1	Positive LVDS output data lane 2 to LVDS7:1 Tx channel 1
d2_ch1_n_o[4]	Out	1	Negative LVDS output data lane 2 to LVDS7:1 Tx channel 1 Complement of d2_ch1_p_o Not available in netlist because this is automatically set to complement by the synthesis tool.
d3_ch1_p_o[3,4]	Out	1	Positive LVDS output data lane 3 to LVDS7:1 Tx channel 1
d3_ch1_n_o[3,4]	Out	1	Negative LVDS output data lane 3 to LVDS7:1 Tx channel 1 Complement of d3_ch1_p_o Not available in netlist because this is automatically set to complement by the synthesis tool.
Miscellaneous			
tstmode_en_i[5]	In	1	Enable or disable test mode. 1 – Enable test mode 0 – Disable test mode
rdy_o[6]	Out	1	1 – Indicates that DDR synchronization is already done. 0 – DDR synchronization is not yet started or still in progress.

Notes:

1. Available only when *Number of Input Pixels per Clock* is more than 1.
2. These pixel data are transmitted by channel 1 when dual channel in selected.
3. Available only for *Data Type* == RGB888.
4. LVDS channel 1 output ports are not available when single LVDS channel is selected.
5. This is in beta mode and is not tested in the initial release. See the Debug Mode section for details.
6. Available if *Enable Miscellaneous signals* is checked.

2.3. Attribute Summary

The configurable attributes of the OpenLDI/FPD-LINK/LVDS Transmitter IP Core are shown in Table 2.2 and are described in Table 2.3. The attributes can be configured through the IP Catalog's Module/IP wizard of the Lattice Radiant software.

Table 2.2. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes
Transmitter			
Number of TX Channels	1,2	1	—
TX Interface	LVDS	LVDS	—
Number of TX Lanes	3,4	4	3 if *Data Type* == RGB666; 4 if *Data Type* == RGB888;
TX Gear	7	7	—
Number of Input Pixels per Clock	1, 2	1	Calculated based on the formula: *Number of Input Pixels per Clock* == *Number of TX Channels*
Video Packet			
Data Type	RGB888, RGB666	RGB888	—
Clock			
TX Total Aggregate Bandwidth (Mbps)	210 - 7560	3780	Calculated based on the formula: *TX Line Rate per lane * Number of TX Lanes * Number of TX Channels*
TX Line Rate per lane (Mbps)	70-945	945	—
Pixel Clock Frequency (MHz)	10-135	135	Calculated based on the formula: *TX Line Rate per lane / TX Gear*
LVDS Output Clock Frequency (MHz)	10-135	135	Calculated based on the formula: *Pixel Clock Frequency* * (*TX Gear/7)*
LVDS ECLK Frequency (MHz)	35-472.5	472.5	Calculated based on the formula: *LVDS Output Clock Frequency* * 3.5
Sync Clock Frequency (MHz)	10-135	10	Must be: ≤ *Pixel Clock Frequency*
Miscellaneous			
Enable Miscellaneous signals	Checked, Unchecked	Unchecked	—
Enable Test Mode	Checked, Unchecked	Unchecked	—
Test mode expected data in Hex format	18'h00000 - 24'hFFFFFF	0	Editable when *Enable Test Mode* is checked

Table 2.3. Attributes Description

Attribute	Dependency on Other Attributes
Transmitter	
Number of TX Channels	Specify how many LVDS links are used
TX Interface	Specify the I/O interface
Number of TX Lanes	Specify number of data lanes per Tx link
TX Gear	Specify what DDR71 gearing is used
Number of Input Pixels per Clock	Specify the number of input pixels per clock
Video Packet	
Data Type	Specify the data type
Clock	
TX Total Aggregate Bandwidth (Mbps)	Tx total line rate
TX Line Rate per lane (Mbps)	Target line rate per lane
Pixel Clock Frequency (MHz)	Pixel clock
LVDS Output Clock Frequency (MHz)	LVDS clock
LVDS ECLK Frequency (MHz)	LVDS edge clock
Sync Clock Frequency (MHz)	Sync clock Must be equal or slower than the slowest clock in the system.
Miscellaneous	
Enable Miscellaneous signals	Enable miscellaneous signals
Enable Test Mode	Enable test mode When enabled, Tx automatically generate test data inside the system through *Test mode expected data in Hex format* and transmit them.
Test mode expected data in Hex format	Test data value used when *Enable Test Mode* is checked

2.4. Interface and Timing Diagrams

Figure 2.2 shows the timing diagram of parallel video input interface. It follows the standard parallel video interface protocol with VSYNC, HSYNC, Data Enable, and pixel data, all clocked by pixel clock. The number of pixels per pixel clock depends on the *Number of TX Channels* and *TX Gear* selected.

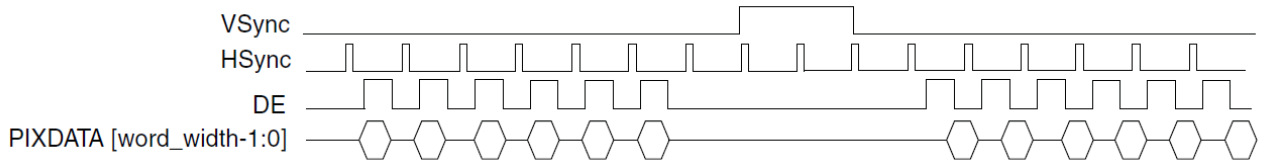

Figure 2.2. Parallel Video Input Interface Timing Diagram

Figure 2.3 shows the pixel data RGB mapping.

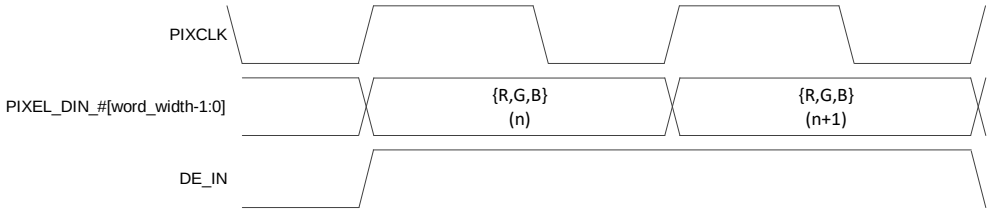

Figure 2.3. Input Pixel Data RGB Arrangement

Figure 2.4 shows the timing of LVDS7:1 output interface. There is a 2-bit offset between the rising edge of LVDS clock and the word boundary. Each word is 7-bit long.

DATAOUT0, DATAOUT1, DATAOUT2, and DATAOUT3 are the data lanes. CLOCKOUT is the LVDS clock lane. For every 7-bit data packet, LSB is the first output serial data to the receiver. A processor sends parallel video packet data to the FPGA chip. Each data lane is serialized using DDR primitive. One channel of LVDS7:1 transmitter has a maximum of five lanes. Each channel consists of one LVDS clock pair and four LVDS data pairs (RGB888) or three LVDS data pairs (RGB666).

The clock lane is generated by feeding constant 1100011. The clock is edge-aligned against data. The clock runs at 1/7th of the data rate, as per the standard for LVDS7:1 interface. Seven bits of data are transmitted in one LVDS clock cycle. The default mode for the LVDS operating system is Unbalanced, as this is commonly used. The maximum supported data rate per lane for LVDS is 945 Mb/s. Maximum of two LVDS7:1 channels can be used. When dual channel is selected, additional data lanes are activated.

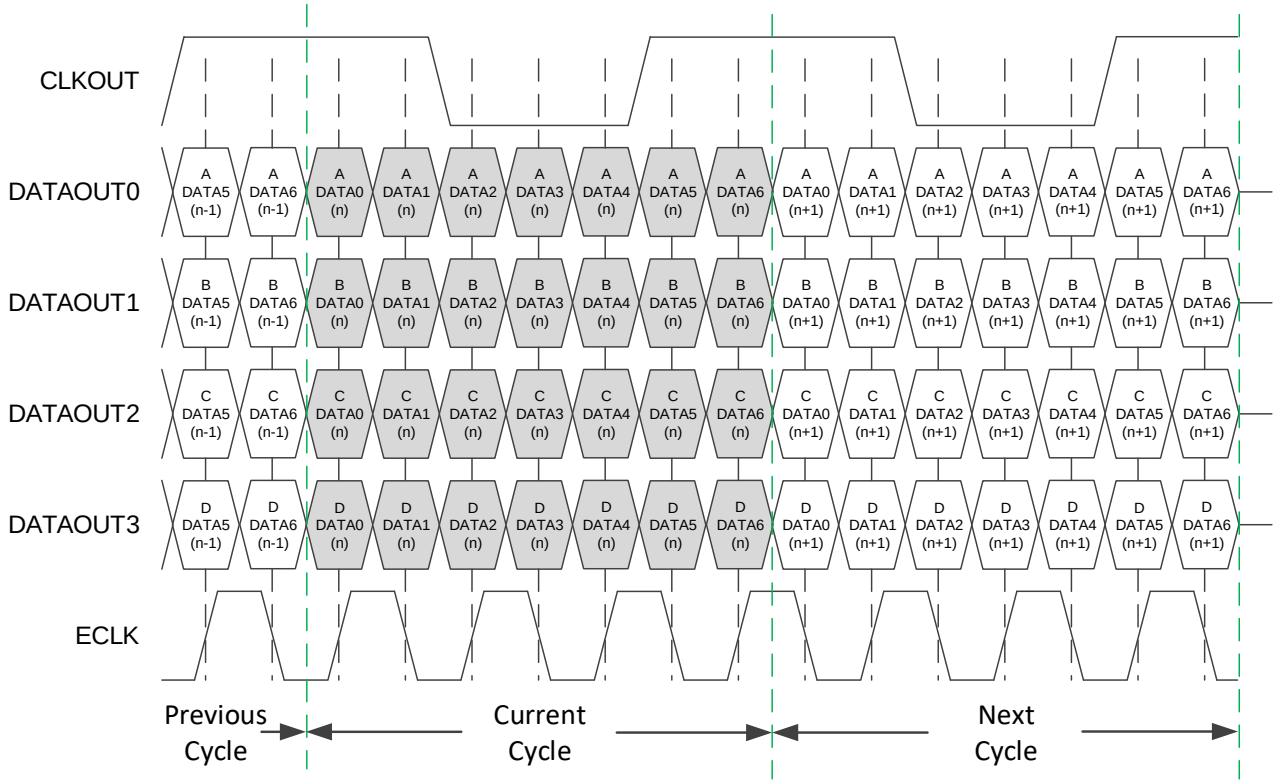

Figure 2.4. OpenLDI/FPD-LINK/LVDS Output Bus Waveform

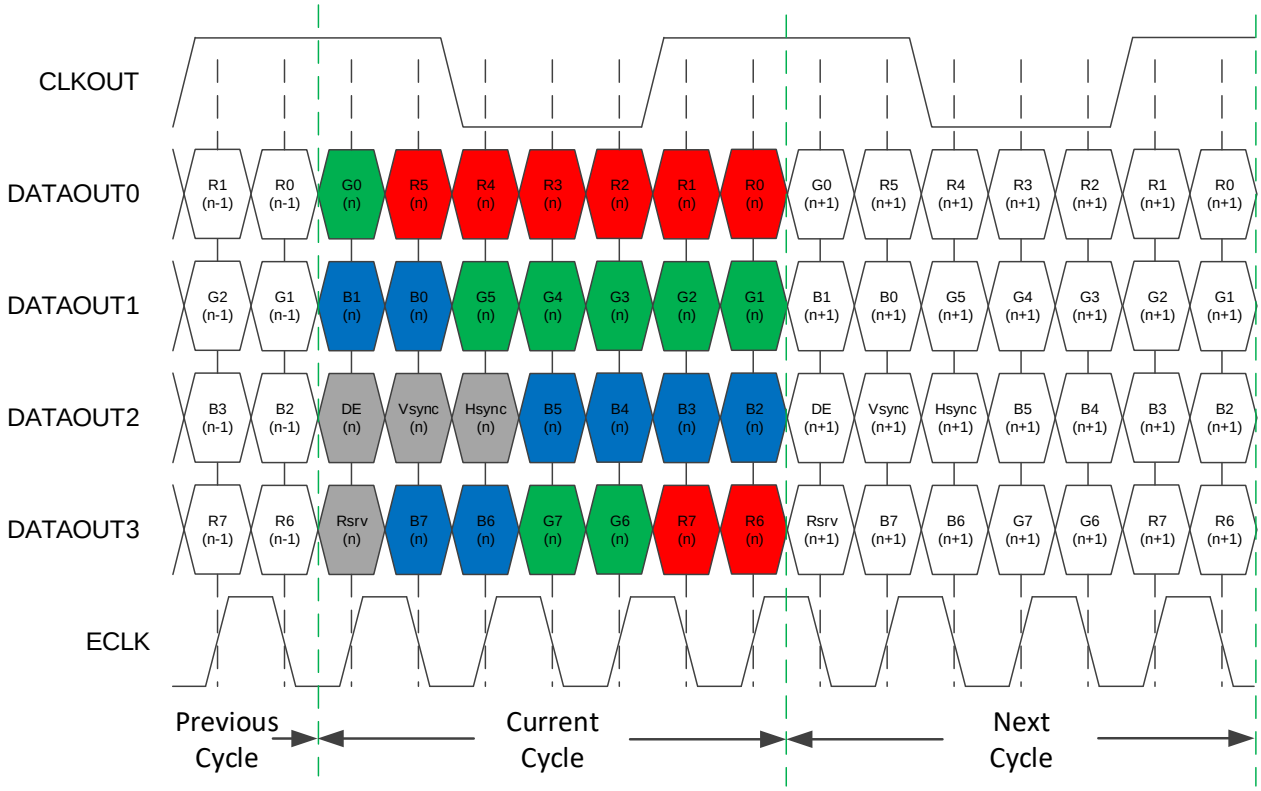

Figure 2.5. Single Channel OpenLDI/FPD-LINK/LVDS Output Bus Waveform for RGB888 Format

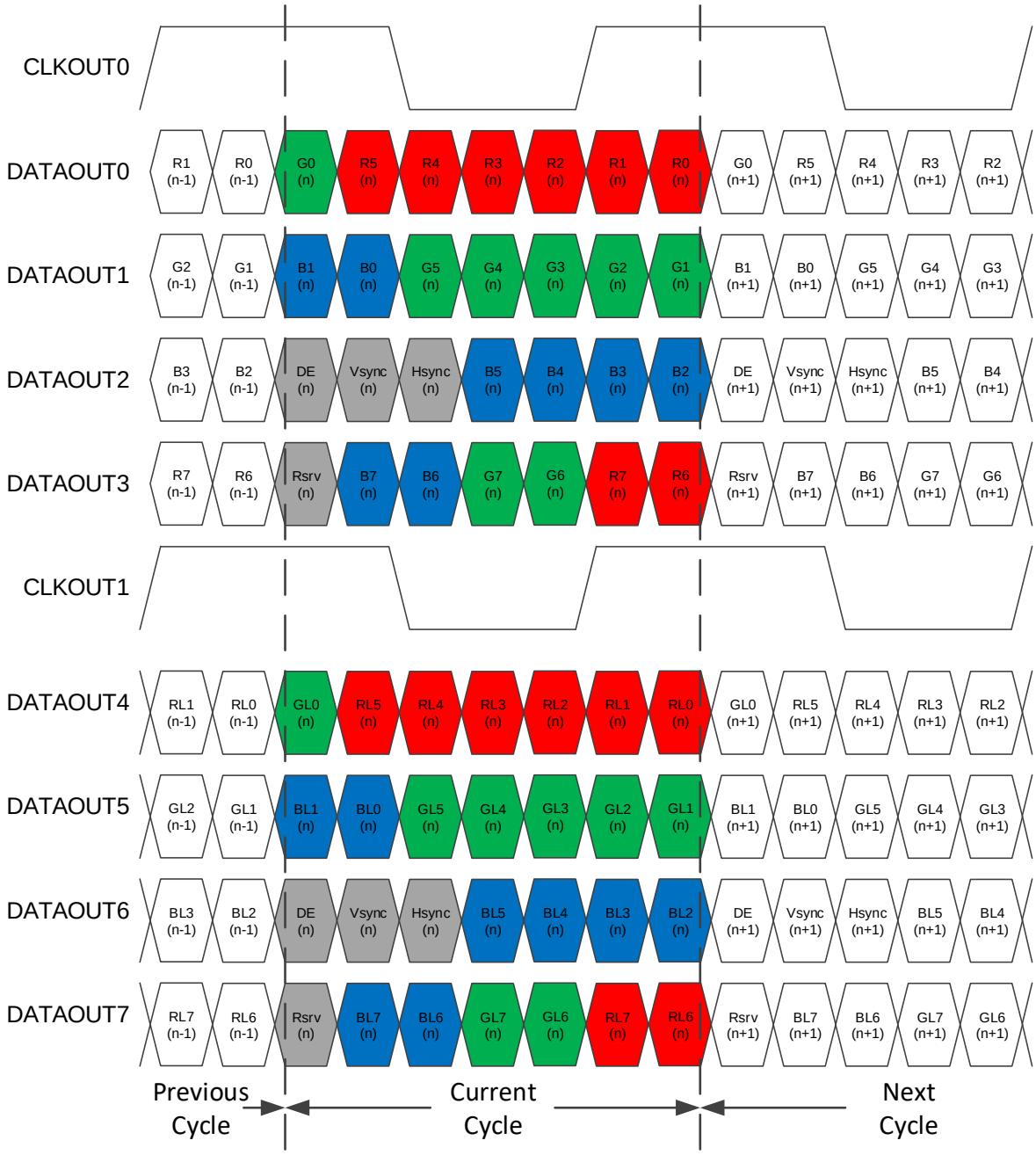

Figure 2.6. Dual Channel OpenLDI/FPD-LINK/LVDS Output Bus Waveform for RGB888 Format

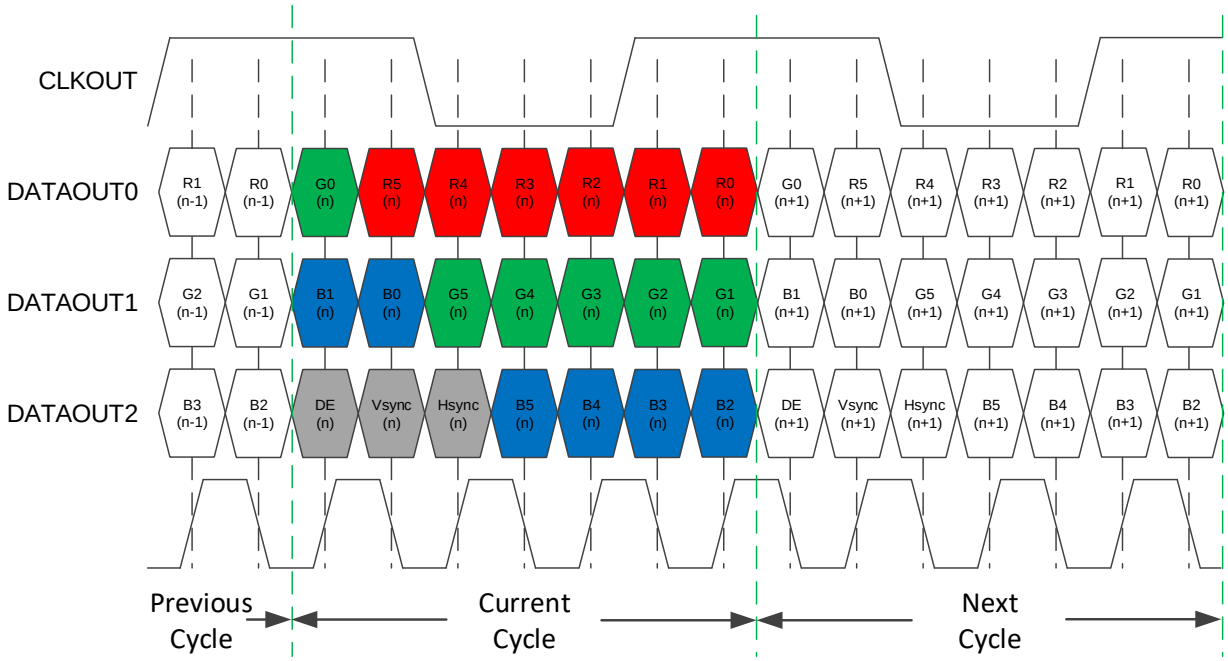

Figure 2.7. Single Channel OpenLDI/FPD-LINK/LVDS Output Bus Waveform for RGB666 Format

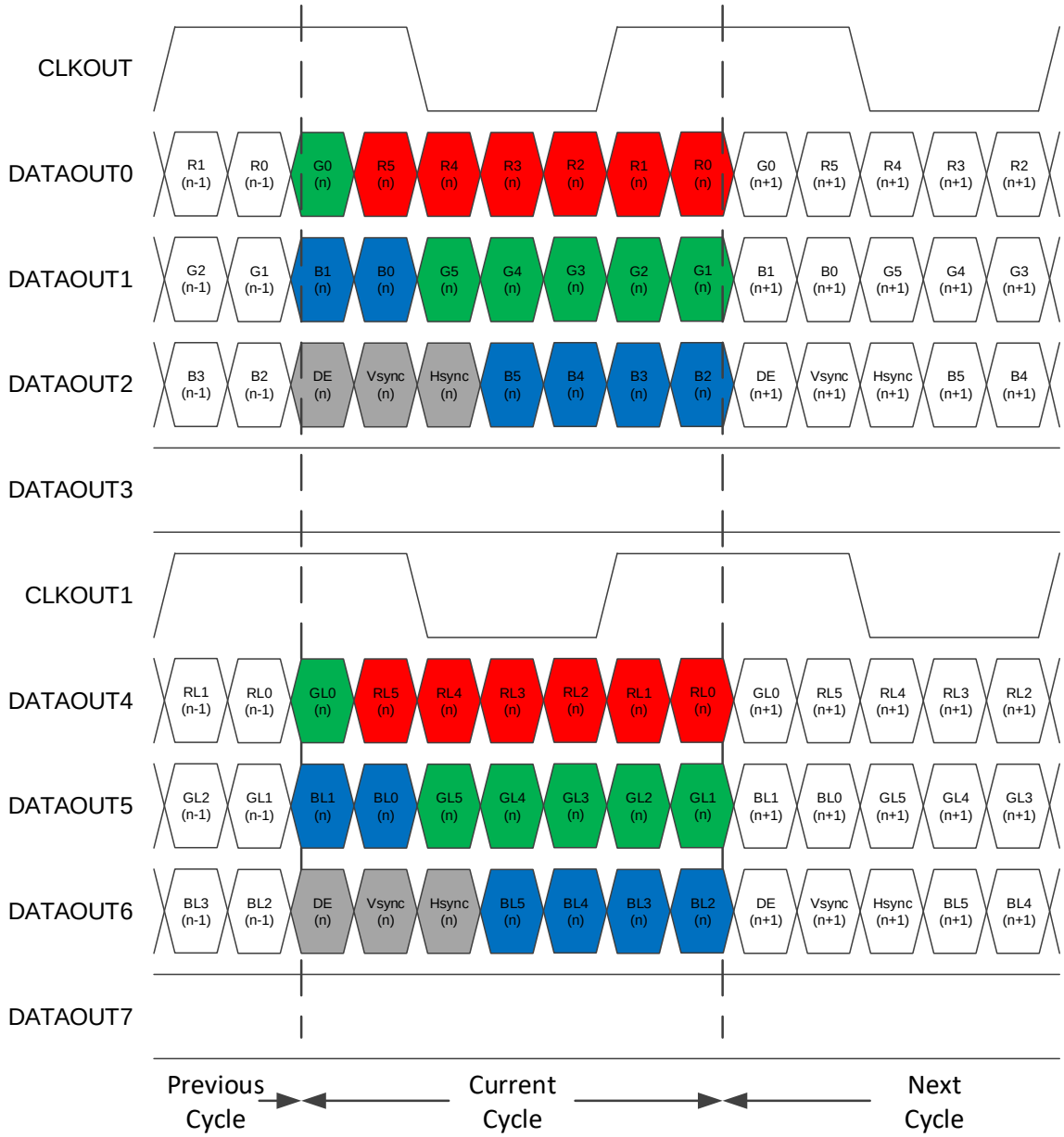

Figure 2.8. Dual Channel OpenLDI/FPD-LINK/LVDS Output Bus Waveform for RGB666 Format

Table 2.4. Input Pixel Data Summary

Number of FPD-Link Channels	Gear	No. of Input Pixel Data	Pixel Clock
1	7	1	LVDS Clock
2	7	2	LVDS Clock

The general arrangement of how pixel data are mapped based on configuration is shown in Figure 2.9.

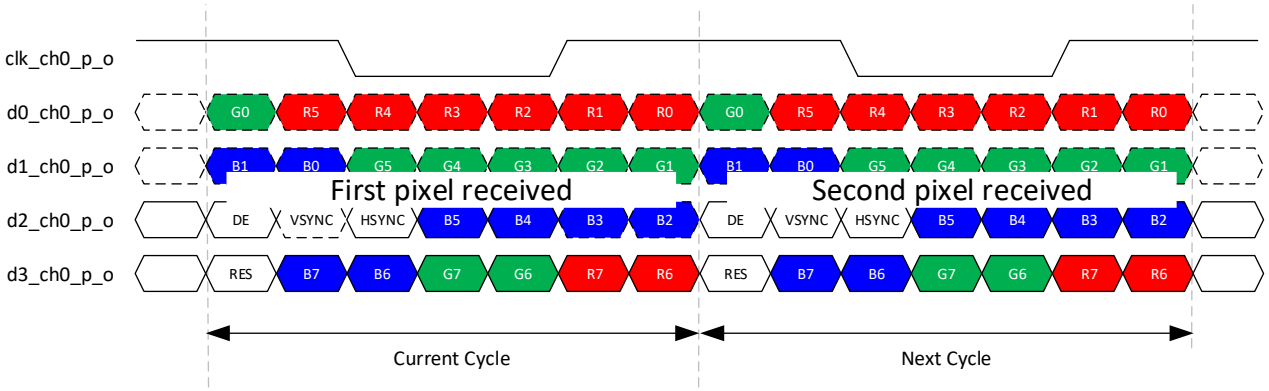

Figure 2.9. Output Pixel Data Arrangement for Single Channel OpenLDI/FPD-LINK/LVDS

2.5. Clock, Reset and Initialization

Active low reset is used in the design with synchronous release. This is the system reset input connected to LVDS7:1 Tx module.

Follow this initialization and reset sequence:

1. Assert active low system reset for at least three clock cycles of the slowest clock (sync clock). It is expected that input clock is already stable after reset. Clock synchronization is started immediately after the release of system reset.
2. If *Enable Miscellaneous signals* is checked, wait for rdy_o to be asserted. The rdy_o is used to indicate that the LVDS7:1 Tx clock synchronization is done. Only when rdy_o is asserted can valid data be sampled and correctly transmitted by the FPD-Link IP.
   If *Enable Miscellaneous signals* is unchecked, wait for some time before sending the valid data to give time for Tx clock synchronization to complete.

2.6. Clock Domains and Clock Domain Crossing

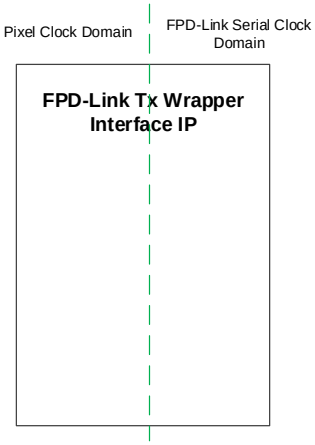

Figure 2.10. Clock Domain Crossing Block Diagram

The general formula for computing the required clocks of the IP:

Tx line rate (total) $= TX\ line\ rate\ (per\ lane) * no.of\ TX\ lane * no.of\ Tx\ Channel$

Pixclk (Pixel Clock) $= \frac{TX\ line\ rate\ (per\ lane)}{TX\ gear}$

Tx LVDS Output Clock $= pixclk * \frac{TX\ gear}{7}$

Tx LVDS ECLK $= pixclk * \frac{TX\ gear}{2}$

Number of Pixels per Pixel Clock $= \frac{TX\ gear}{7} * no.of\ Tx\ Channel$

2.7. Module Description

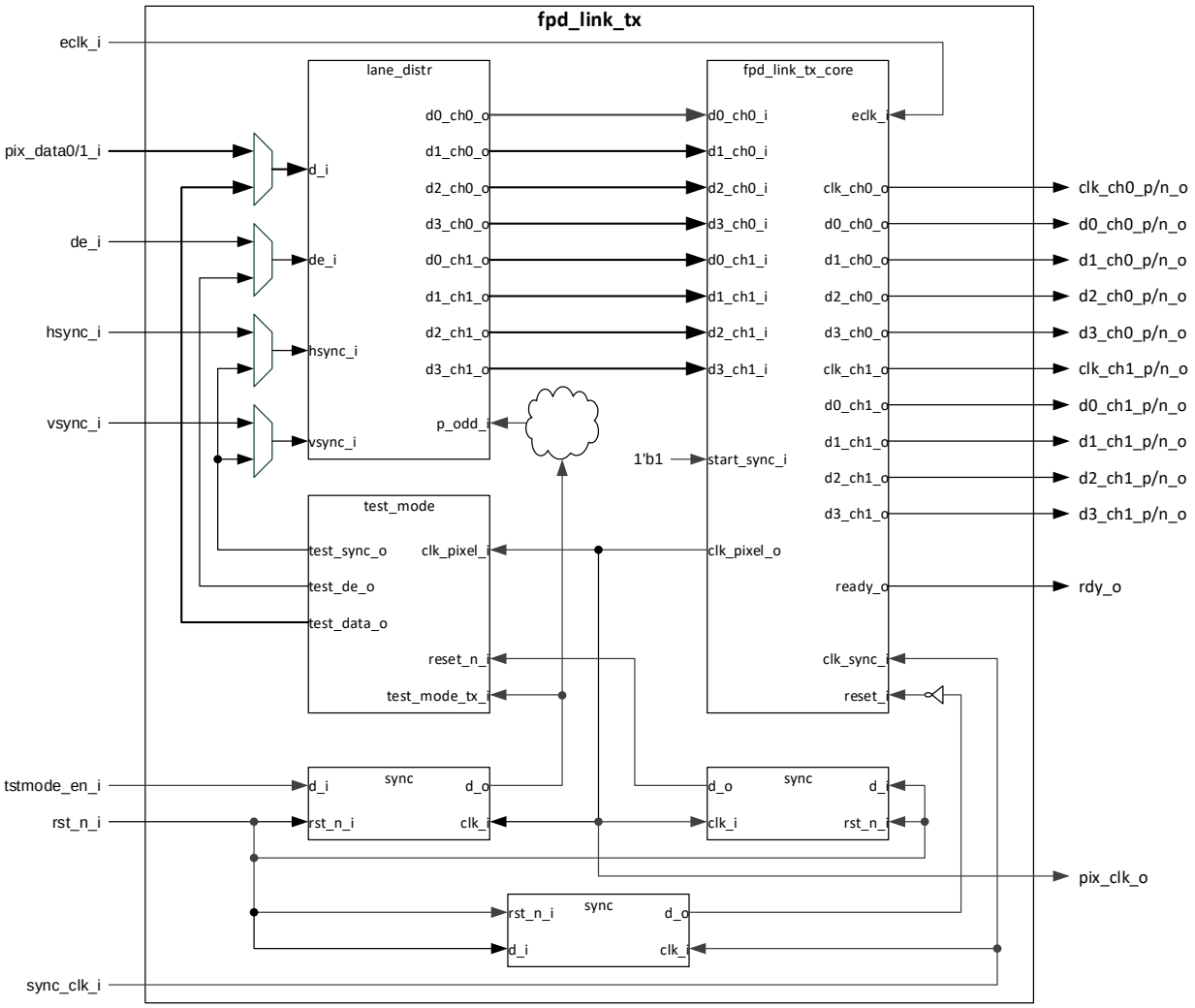

Figure 2.11. FPD-Link Tx Block Diagram

The fpd_link_tx.v module instantiates fpd_link_tx_core, test_mode, lane_distr, and synchronizer modules. The fpd_link_tx_core module is the core module which does parallel to serial conversion. The test_mode is used for automatic generation of pixel data inside the system during debug mode. lane_distr is used for distributing pixel data to different LVDS lanes for OpenLDI unbalanced format. Synchronizers are two-level synchronizers used to sync the system reset and test mode enable signal into different clock domains before it is used in the system.

2.7.1. FPD-LINK Tx Core

The fpd_link_tx_core module instantiates GDDR_SYNC, lvds_oddr, and lvds_clk_tree modules. The GDDR_SYNC module is required to initialize and synchronize DDR clock, and tolerate the large skew between stop and reset of the DDR components.

Parallel data are fed to the I/O logic DDR71 register in the lvds_oddr module. This module is used to convert the incoming parallel data into serial format. lvds_clk_tree is used to generate the clocks needed by the system. The SCLK is used as the output pixel clock of DDR71 IP.

2.7.2. Lane Distribution Module

lane_distr module is used for distributing pixel data to different LVDS lanes for Openldi unbalanced format.

2.7.3. Test Mode Tx Module

The test_mode_tx is used to automatically generate test data inside the system through *Test mode expected data in Hex format*, and transmit them. Data comparison should be done outside the design. See Debug Mode section for details on how to enable test mode.

2.7.4. Synchronizer Module

Synchronizer is a two-level synchronizer used to sync the input data into a different clock domain. In the design, this is used to synchronize the system reset into different clock domains before it is used in the system.

2.8. Debug Mode

This debug feature is used for automatic transmission of test data that is configured using *Test mode expected data in Hex format*. The test mode module drives channel 0 and channel 1 (if enabled) with predefined test data. Data comparison should be done outside the design.

To enable test mode:

1. Make sure *Enable Test Mode* is checked and *Test mode expected data in Hex format* is configured during IP generation. Pre-defined data is equivalent to {R[7:0],G[7:0],B[7:0]}.
2. Drive tstmode_en_i to 1'b1.
3. Perform sequence as specified in Clock, Reset and Initialization section.
4. Monitor data transmission over LVDS lanes (test data is sent continuously with DE = high, VSYNC = HSYNC = RSVD = 0).

3. IP Generation and Evaluation

This section provides information on how to generate the OpenLDI/FPD-LINK/LVDS Transmitter IP Core using the Lattice Radiant software and how to run simulation and synthesis. For more details on the Lattice Radiant software, please refer to Lattice Radiant Software 2.1 User Guide.

3.1. Licensing the IP

An IP core-specific license string is required enable full use of the OpenLDI/FPD-LINK/LVDS Transmitter IP Core in a complete, top-level design.

When the IP Core is used in CrossLink-NX and Certus-NX devices, you can fully evaluate the IP core through functional simulation and implementation (synthesis, map, place and route) without an IP license string. This IP core supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP core, which operate in hardware for a limited time (approximately four hours) without requiring an IP license string. See the Hardware Evaluation section for further details. However, a license string is required to enable timing simulation and to generate bitstream file that does not include the hardware evaluation timeout limitation.

3.2. Generation and Synthesis

The Lattice Radiant software allows you to customize and generate modules and IPs and integrate them into the device's architecture. The procedure for generating the OpenLDI/FPD-LINK/LVDS Transmitter IP Core in Lattice Radiant software is described below.

To generate the OpenLDI/FPD-LINK/LVDS Transmitter IP Core:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click **FPD-Link Transmitter** under the **IP, Audio_Video_Image_Processing** category. The **Module/IP Block Wizard** opens as shown in Figure 3.1.
3. Enter values in the **Instance name** and **Create in** fields. Click **Next**.

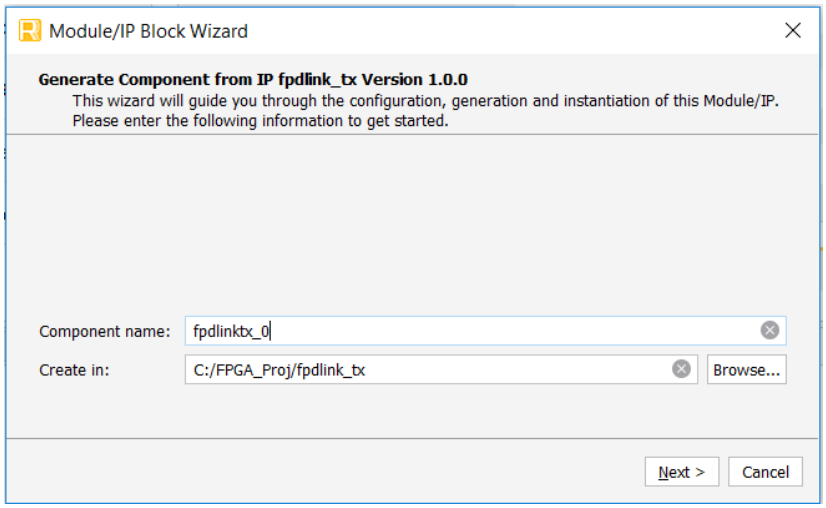

Figure 3.1. Module/IP Block Wizard

4. In the module's dialog box of the **Module/IP Block Wizard** window, customize the selected OpenLDI/FPD-LINK/LVDS Transmitter IP Core using drop-down menus and check boxes. As a sample configuration, see Figure 3.2. For configuration options, see the Attribute Summary section.

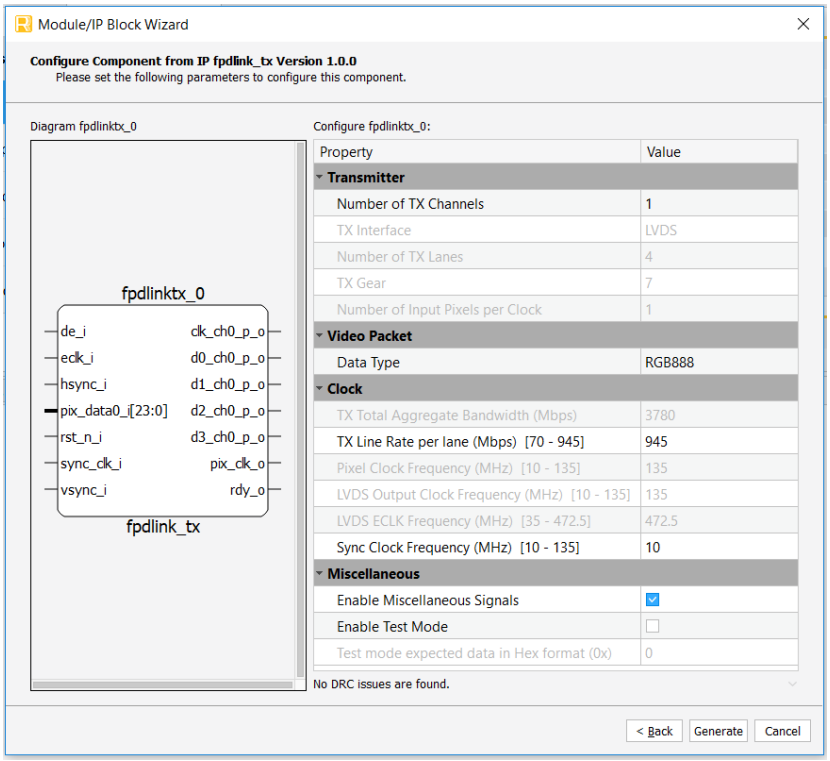

Figure 3.2. Configure User Interface of selected OpenLDI/FPD-LINK/LVDS Transmitter IP Core

5. Click **Generate**. The **Check Generating Result** dialog box opens, showing design block messages and results. See Figure 3.2.

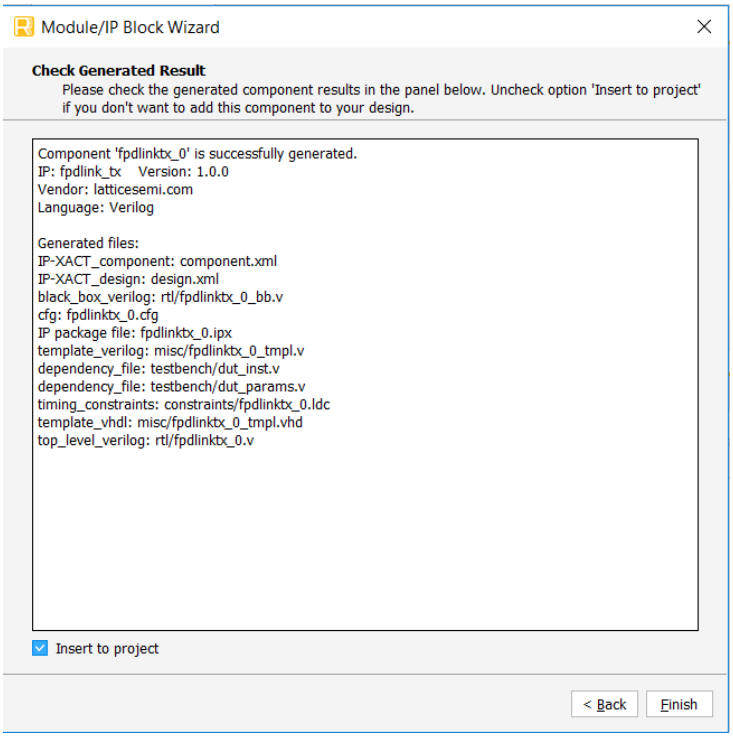

Figure 3.3. Check Generating Result

6. Click the **Finish** button. All the generated files are placed under the directory paths in the **Create in** and **Instance name** fields shown in Figure 3.1.

The generated OpenLDI/FPD-LINK/LVDS Transmitter IP Core package includes the black box (<Instance Name>_bb.v) and instance templates (<Instance Name>_tmpl.v/vhd) that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file (<Instance Name>.v) that can be used as an instantiation template for the IP core is also provided. You may also use this top-level reference as the starting template for the top-level for their complete design. The generated files are listed in Table 3.1.

Table 3.1. Generated File List

Attribute	Description
<Instance Name>.ipx	This file contains the information on the files associated to the generated IP.
<Instance Name>.cfg	This file contains the parameter values used in IP configuration.
component.xml	Contains the ipxact:component information of the IP.
design.xml	Documents the configuration parameters of the IP in IP-XACT 2014 format.
rtl/<Instance Name>.v	This file provides an example RTL top file that instantiates the IP core.
rtl/<Instance Name>_bb.v	This file provides the synthesis black box.
misc/<Instance Name>_tmpl.v misc /<Instance Name>_tmpl.vhd	These files provide instance templates for the IP core.

3.2.1. Required Post-Synthesis Constraints

The OpenLDI/FPD-LINK/LVDS Transmitter IP Core has several clocks. These clocks need to be constrained in the post-synthesis constraint file of the Lattice Radiant project.

In the generated IP, there is a constraint file (*.ldc) that is generated based on user settings.

To constrain the clocks:

1. Copy the contents of the *.ldc file to the active Post Synthesis Constraint File of the Lattice Radiant project.
2. Uncomment *set_false_path* and modify the path/s based on the generated netlist.

3.3. Running Functional Simulation

Running functional simulation can be performed after the IP is generated. The following steps can be performed.

1. To run simulation, add the top level testbench file *tb_top.v* in the project as a simulation file. Click the **File** tab and select **Add** in the drop down menu.
2. Click **Existing Simulation File** and select the *<Component name>/testbench/tb_top.v* file.
3. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in Figure 3.4.

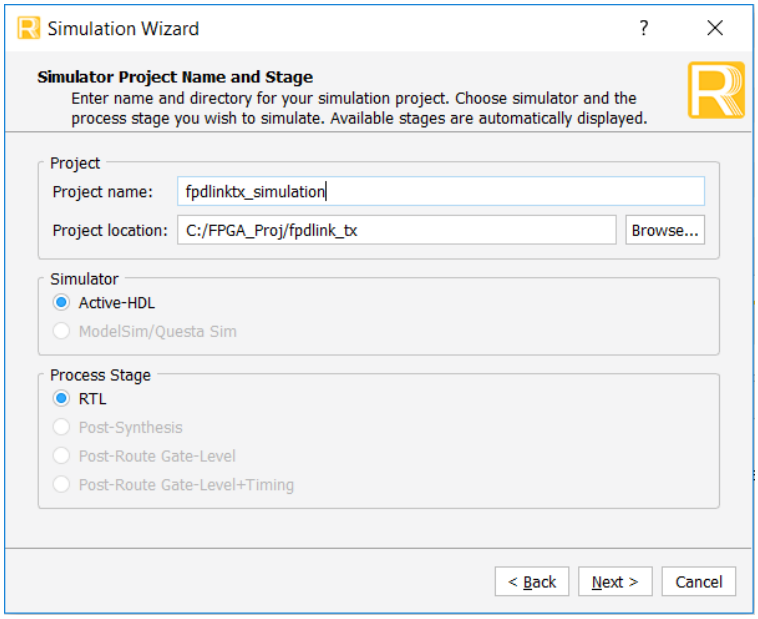

Figure 3.4. Simulation Wizard

4. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 3.5.

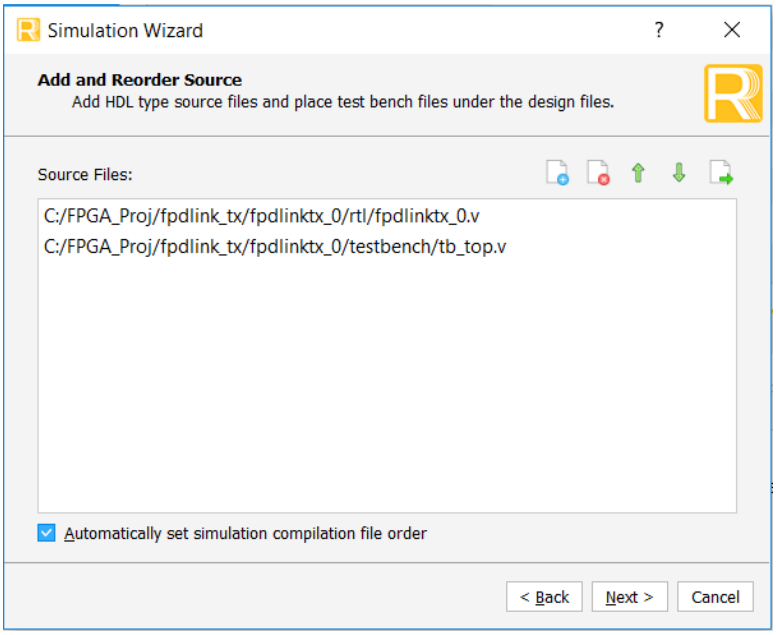

Figure 3.5. Adding and Reordering Source

5. Click **Next**. The **Summary** window is displayed.
6. Click **Finish** to run the simulation.

The result of the simulation in our example is shown in Figure 3.6.

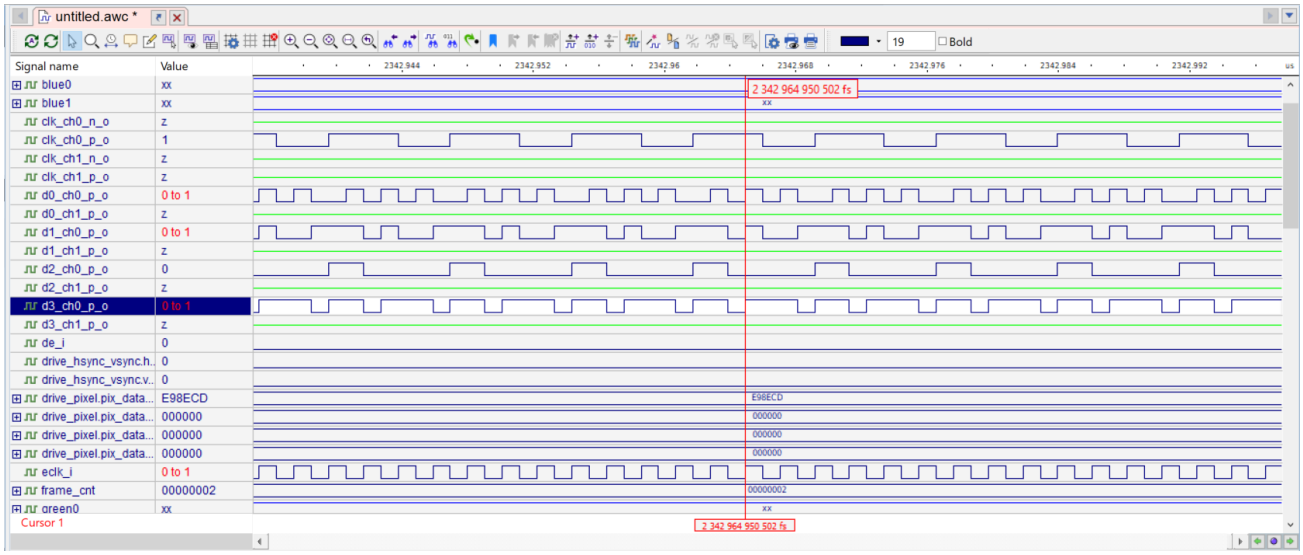

Figure 3.6. Simulation Waveform

Notes:

- It is necessary to follow the procedure above until it is fully automated in the Lattice Radiant software Suite.
- Image resolution of testbench can be configured through parameter file <Component name>/testbench/tb_params.v.
- When simulation finishes successfully, log files are generated for input and output data (input_data0.log and pixel_out0.log for Tx channel 0 and input_data1.log and pixel_out1.log for Tx channel 1, respectively). Compare these log files to check if simulation passes.

3.4. Hardware Evaluation

The OpenLDI/FPD-LINK/LVDS Transmitter IP Core supports Lattice's IP hardware evaluation capability when used with LIFCL and LFD2NX devices. This makes it possible to create versions of the IP core that operate in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. It may also be used to evaluate the core in hardware in user-defined designs. The hardware evaluation capability may be enabled/disabled in the Strategy dialog box. It is enabled by default. To change this setting, go to Project > Active Strategy > LSE/Synplify Pro Settings.

4. Ordering Part Number

The Ordering Part Number (OPN) for this IP Core are the following:

- FPD-TX-CNX-U – FPD-LINK Transmitter for CrossLink-NX - Single Design License
- FPD-TX-CNX-UT – FPD-LINK Transmitter for CrossLink-NX - Site License
- FPD-TX-CTNX-U – FPD-LINK Transmitter for Certus-NX - Single Design License
- FPD-TX-CTNX-UT – FPD-LINK Transmitter for Certus-NX - Site License

Appendix A. Resource Utilization

Table A.1 shows the resource utilization of the OpenLDI/FPD-LINK/LVDS Transmitter Core for the LIFCL-40-9BG400I device using Lattice Synthesis Engine of the Lattice Radiant software. Default configuration is used and some attributes are changed from the default value to show the effect on the resource utilization.

Table A.1. Resource Utilization

Configuration	Clk Fmax (MHz)[1]	Slice Registers	LUTs[2]	EBRs
Default	200	18	40	0
Number of TX Channels = 2, Others = Default	200	18	39	0
Data Type = RGB666, Others = Default	200	18	40	0
Number of TX Channels = 2, *Data Type* = RGB666, Others = Default	200	18	39	0

Notes:

1. Fmax is generated when the FPGA design only contains OpenLDI/FPD-LINK/LVDS Transmitter Core and the target frequency is 135 MHz. These values may be reduced when user logic is added to the FPGA design.
2. The *distributed RAM* utilization is accounted for in the total LUT4s utilization. The actual LUT4 utilization is distribution among *logic*, *distributed RAM*, and *ripple logic*.

Appendix B. Limitations

The following is a known limitation:

- For Dual Channel configuration, odd multiple number of pixels is not supported.

References

For complete information on Lattice Radiant Project-Based Environment, Design Flow, Implementation Flow and Tasks, as well as on the Simulation Flow, see the Lattice Radiant Software 2.1 User Guide.

Technical Support Assistance

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

Revision History

Revision 1.0, August 2020

Section	Change Summary
All	Initial release.

www.latticesemi.com

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Development Software ***category:***

Click to view products by Lattice ***manufacturer:***

Other Similar products are found below :

RAPPID-560XBSW RAPPID-567XFSW DG-ACC-NET-CD SRP004001-01 SW006021-1NH SW163052 SYSWINEV21 Core429-SA SW500006-HPA CWP-BASIC-FL W128E13 CWP-PRO-FL AD-CCES-NODE-1 NT-ZJCAT1-EV4 CWA-BASIC-FL RAPPID-567XKSW CWA-STANDARD-R SW89CN0-ZCC CWA-LS-DVLPR-NL CWA-STANDARD-FL VDSP-21XX-PCFLOAT RAPPID-563XMSW IPS-EMBEDDED SWR-DRD-L-01 SDAWIR-4532-01 MPROG-PRO535E AFLCF-08-LX-CE060-R21 WS02-CFSC1-EV3-UP SYSMAC-STUDIO-EIPCPLR LIB-PL-PC-N-1YR-DISKID LS1043A-SWSP-PRM SW006026-COV 1120270005 1120270006 MIKROBASIC PRO FOR FT90X (USB DONGLE) MIKROC PRO FOR AVR (USB DONGLE LICENSE) MIKROC PRO FOR FT90X (USB DONGLE) MIKROBASIC PRO FOR AVR (USB DONGLE LICEN MIKROBASIC PRO FOR FT90X MIKROC PRO FOR DSPIC30/33 (USB DONGLE LI MIKROC PRO FOR FT90X MIKROC PRO FOR PIC32 (USB DONGLE LICENSE 52202-588 MIKROPASCAL PRO FOR ARM (USB DONGLE LICE MIKROPASCAL PRO FOR FT90X MIKROPASCAL PRO FOR FT90X (USB DONGLE) MIKROPASCAL PRO FOR PIC32 (USB DONGLE LI SW006021-2H SW006023-3 CWP-STANDARD-FL