LVDS IP

User Guide

(UG042008, V1.5)

(24.04.2024)

Revisions History

Document Revisions

Version	Date of Release	Revisions	Applicable IP and Corresponding Versions
V1.5	24.04.2024	Initial release.	V1.5

IP Revisions

IP Version	Date of Release	Revisions
V1.5	24.04.2024	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
APB	Advanced Peripheral Bus
DDR	Double Data Rate
HR	High Range
HP	High Performance
IOB	Input/Output Buffer
LVDS	Low Voltage Differential Signal
PLL	Phase Lock Loop
SoPC	System on Programmable Chip
UI	User Interface
IPC	IP Compiler
PDS	Pango Design Suite
UCE	User Constraint Editor

Related Documentation

The following documentation is related to this manual:

1. ***Pango_Design_Suite_Quick_Start_Tutorial***
2. ***Pango_Design_Suite_User_Guide***
3. ***IP_Compiler_User_Guide***
4. ***Simulation_User_Guide***
5. ***User_Constraint_Editor_User_Guide***
6. ***Physical_Constraint_Editor_User_Guide***
7. ***Route_Constraint_Editor_User_Guide***
8. ***UG040007_Logos2 Family Product GTPs User Guide***
9. ***UG040006_Logos2 Family FPGAs Input/Output Interface (IO) User Guide***
10. ***UG040004_Logos2 Family FPGAs Clock Resources (Clock) User Guide***
11. ***UG041001_PLL_IP***
12. ***UG040012_Logos2 Single Board Hardware Design Guide***
13. ***UG050007_Titan2 Family Product GTPs User Guide***

14. UG050006_Titan2 Family FPGAs Input/Output Interface (IO) User Guide

15. UG050004_Titan2 Family FPGAs Clock Resources (Clock) User Guide

16. UG050012_Titan2 Single Board Hardware Design Guide

17. UG100007_Kosmo2 Family Product GTPs User Guide

18. UG100006_Kosmo2 Family SoPCs Input/Output Interface (IO) User Guide

19. UG100004_Kosmo2 Family SoPCs Clock Resources (Clock) User Guide

Table of Contents

Tables

Figures

Chapter 1 Preface

This chapter describes the scope, structure, and writing standards of this manual to help users quickly find the information they need.

1.1 Introduction of the Manual

This manual is a user guide for the LVDS IP product launched by Pango Microsystems. The content of this manual primarily includes the IP user guide and related information. This manual helps users quickly understand the features and usage of LVDS IP.

1.2 Writing Standards of the Manual

Table 1-1 Description of Writing Standards

Text	**Rules**
Attention	If users ignore the attention contents, they may suffer adverse consequences or fail to operate successfully due to incorrect actions.
Description	Instructions and tips provided for users.
Recommendation	Recommended settings and instructions for users.

Chapter 2 IP User Guide

This chapter provides a guide on the use of LVDS IP, including an introduction to IP, IP block diagram, IP generation process, Example Design, IP interface description, typical applications, instructions and considerations, and IP debugging methods. More details on the design process can be found in the following PDS help documentation.

- "***Pango_Design_Suite_Quick_Start_Tutorial***"
- "***Pango_Design_Suite_User_Guide***"
- "***IP_Compiler_User_Guide***"
- "***Simulation_User_Guide***"

2.1 IP Introduction

The LVDS IP is a general-purpose parallel interface IP launched by Pango Microsystems, using FPGA product IO resources and clock network. It is designed for multi-channel high-speed data transmission between FPGA and peripheral devices. The LVDS IP can configure and generate the IP module using the IPC (IP Compiler) tool within the PDS (Pango Design Suite).

2.1.1 Key Features

The key features of the LVDS IP are as follows:

- Supports four operating modes: "TX," "RX," "TX_RX," and "TX_RX_LOOP;"
- Supports 1~20 data transfer channels[1];
- Supports two data transfer methods: SDR and DDR;
- HR bank supports a maximum transmission rate of 1250 Mbps;
- HP bank[2] supports a maximum transmission rate of 1500 Mbps;

1 The number of interface channels should be properly configured according to the chip package type.

2 Titan2 and Kosmo2 family devices support HP bank configuration.

- Supports multiple serialization factors (2~8, 10, 14);
- Supports the use of on-chip PLL resources;
- Supports delay adjustment for data paths and clock paths;
- Supports word boundary alignment on the receiver;
- Timing training reference solutions for the receiver[3];
- The transmitter supports sending the channel-associated clock;
- Supports interface initialization;
- Supports interface drive types and IO level standards configuration;
- Supports interface resource statistics.

2.1.2 Applicable Devices and Packages

Table 2-1 LVDS IP Applicable Devices and Package

Applicable Devices	**Supported Packages**
PG2L50H	ALL
PG2L200H	ALL
PG2T390HX	ALL
PG2T70H	ALL
PG2K400	ALL
PG2T160H	ALL

2.2 IP Block Diagram

The system block diagram of the LVDS IP is shown in Figure 2-1. The functional modules are mainly divided into the transmission module (tx_top), receive module (rx_top), reset sequence (rst_sequence), and bit order adjustment module (aligner).

3 Available only when the "Work Mode" parameter is set to "TX_RX_LOOP".

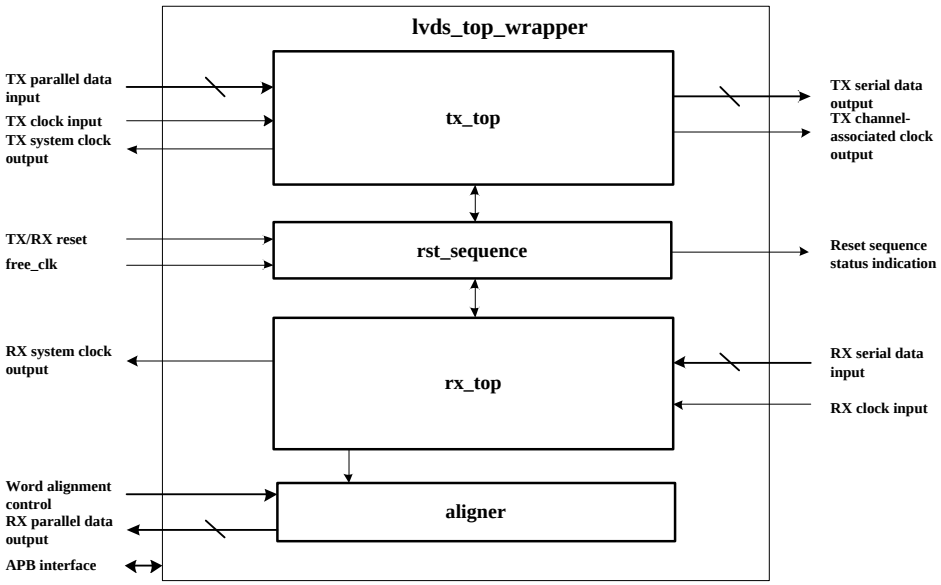


Figure 2-1 LVDS IP System Block Diagram

2.2.1 Module Description

2.2.1.1 LVDS Transmission Module

The tx_top module converts the parallel data from the FPGA core into serial data and then drives the output through the IOB using a specific level standard. The transmission module outputs the transmitter system clock (tx_sysclk) for user utilization. For source synchronous applications, the interface supports sending either the channel-associated clock or the channel-associated divided clock.

2.2.1.2 LVDS Receive Module

The rx_top module first reliably receives off-chip high-speed serial data from IOB, then processes the data through serial-to-parallel conversion and outputs the parallel data to the FPGA core. The receive module outputs the receiver system clock rx_sysclk for user utilization. The IP reference design provides user-selectable automatic timing training and word alignment functions. For detailed instructions, please refer to "2.4 Example Design".

2.2.1.3 Reset Sequence

The rst_sequence module is used to control the reset and initialization of various hard core modules in the interface. Effective data transmission and reception can only be performed by the user after the interface initialization is completed.

2.2.1.4 Bit Order Adjustment Module

The aligner module achieves bit order adjustment logic based on fixed data sequence comparison principles. By sending shift pulses (bitslip), the bit order of parallel data can be adjusted, achieving word boundary alignment on the receiver end.

2.2.2 Operating Modes

The LVDS IP can realise four operating modes by integrating the aforementioned functional modules: Transmission (TX), Reception (RX), Duplex Operation (TX_RX), and Loopback (TX_RX_LOOP). Users can configure other functional parameters through the IPC. For detailed information, please refer to "2.3.1.2 IP Parameter Configuration".

2.2.2.1 TX Mode

The IP top level includes the LVDS transmit module and reset sequence.

2.2.2.2 RX Mode

The IP top level includes LVDS receive module and reset sequence. When the word boundary alignment function is enabled, it also includes the bit order adjustment module.

2.2.2.3 TX_RX Mode

The IP top level includes the LVDS transmit module, LVDS receive module, and reset sequence. When the word boundary alignment function is enabled, it also includes the bit order adjustment module.

2.2.2.4 TX_RX_LOOP Mode

The IP top level includes the LVDS transmit module, LVDS receive module, and reset sequence. In this mode, word boundary alignment and timing training functions are enabled by default, and data loopback is required to facilitate users to quickly understand the interface and complete basic interface debugging.

2.3 IP Generation Process

2.3.1 Module Instantiation

With the IPC tool, users can customise the configuration of LVDS IP and instantiate the required IP modules. For detailed instructions on using the IPC tool, please refer to "***IP_Compiler_User_Guide***".

The main steps for instantiating the LVDS IP module are described as follows.

2.3.1.1 Selecting IP

Open IPC and click File > Update in the main window to open the Update IP dialog box, where you add the corresponding version of the IP model.

After selecting the device type, the Catalog interface displays the loaded IP models. Select the corresponding version of LVDS under the "System/LVDS" directory. The IP selection path is shown in Figure 2-2. Then set the Pathname and Instance Name on the right side of the page. The project instantiation interface is shown in Figure 2-3.

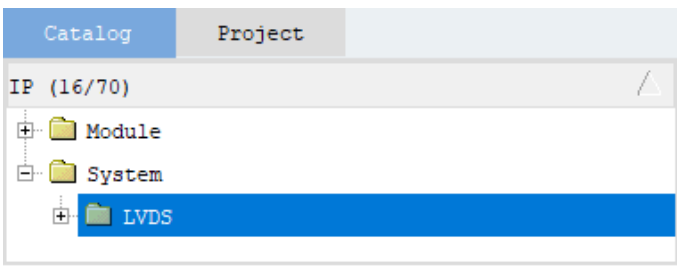


Figure 2-2 LVDS IP Selection Path

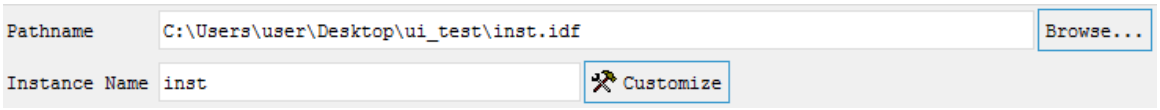


Figure 2-3 Project Instantiation Interface

Attention:

PG2L50H device: The software version must be 2022.1 or above;

PG2L200H device: The software version must be 2022.2 or above;

PG2T390HX device: The software version must be 2023.1 or above;

PG2T70H device: The software version must be 2023.2 or above;

PG2K400 device: The software version must be 2023.2 or above;

PG2T160H device: The software version must be 2024.1 or above.

2.3.1.2 IP Parameter Configuration

After selecting the IP, click <Customize> to enter the LVDS IP parameter configuration interface. The left Symbol is the interface block diagram, as shown in Figure 2-4; the IP Parameter Configuration window is shown on the right side.

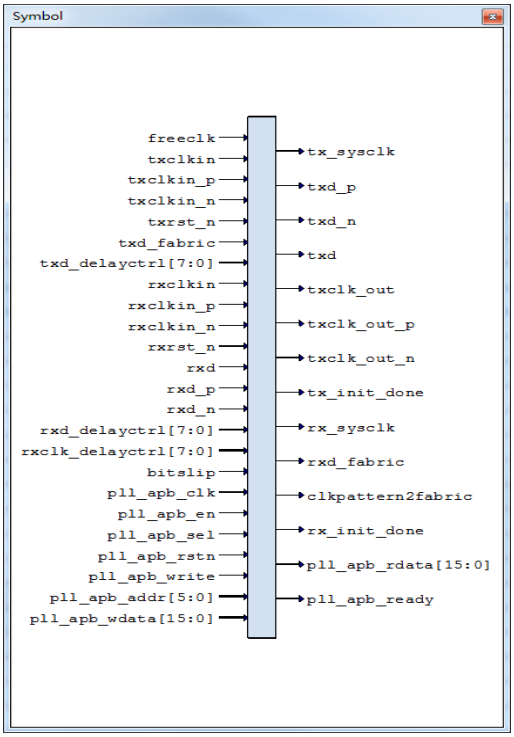


Figure 2-4 Block Diagram of the LVDS IP Interface

IP parameter configuration consists of three pages: Interface Configurations, Pin Settings, and Interface Summary. The configuration steps for LVDS IP are described as follows.

2.3.1.2.1 Interface Configurations Page

Interface Configurations is the basic configuration page for IP, as shown in Figure 2-5. Please refer to Table 2-2 for detailed parameters.

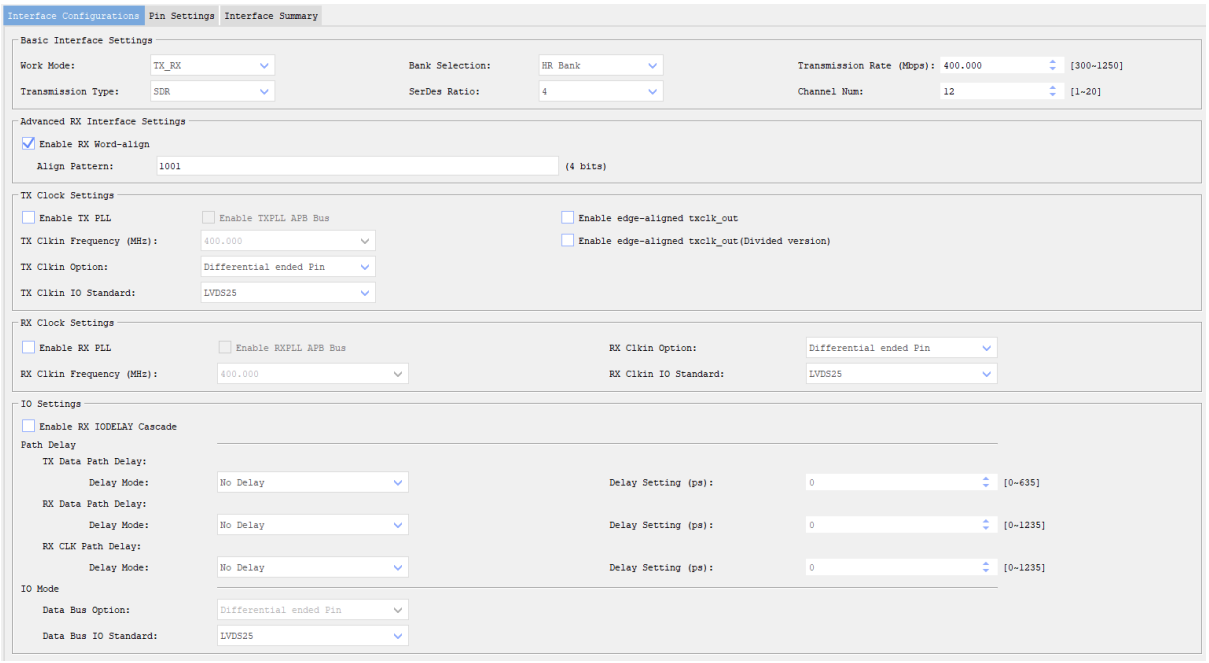


Figure 2-5 Interface Configurations Page

Table 2-2 Descriptions of Configuration Parameters on the Interface Configurations Page

Option Domain	Parameter/Configuration Options	Parameter Description	Default Value
Basic Interface Settings	Work Mode	LVDS interface operating mode selection: 1) TX: Transmit mode; 2) RX: Receive mode; 3) TX_RX: Bidirectional mode; 4) TX_RX_LOOP: Loopback mode.	TX_RX
	Bank Selection	IO Bank selection: HR Bank: Wide voltage range IO bank; HP Bank: High-performance IO bank.	HR Bank
	Transmission Rate (Mbps)	Interface data transfer rate configuration: HR Bank: 300~1,250 Mbps; HP Bank: 300~1,500 Mbps.	400
	Transmission Type	Data transfer type selection: 1) SDR: Data transmission on a single clock edge; 2) DDR: Data transmission on both edges of the clock. Note: Only DDR is supported when transmission rate is greater than 800 Mbps.	SDR
	SerDes Ratio	Serdes serial-to-parallel conversion factor selection: 1) SDR transmission type supports: 2, 3, 4, 5, 6, 7, 8; 2) DDR transmission type supports: 2, 4, 6, 8, 10, 14.	4
	Channel Num	Data channel number configuration; The range is 1 to 20.	12

Option Domain	Parameter/Configuration Options	Parameter Description	Default Value
Advanced RX Interface Settings[4]	Enable RX Training[5]	The timing training function is enabled in the Example Design; For more information on timing training, please refer to "2.4.2.6 Training Module".	Cleared
	Clock Pattern[6]	Clock pattern setting used for timing training.	0011
	Enable RX Word-align[7]	Enable word boundary alignment function at the receiver.	Selected
	Align Pattern[8]	Data pattern configuration used for word boundary alignment; Notes: The data pattern configured by RX should be consistent with the transmitter's data pattern. When configuring data patterns, avoid creating a new pattern that is identical to the original pattern after shifting the sequence n times, where 1≤n< SerDes Ratio.	1001
TX Clocking Settings	Enable TX PLL	Configuration for TX PLL enable: Selected: The required high-speed serial clock and low-speed parallel clock for the interface are generated by the GPLL. Cleared: The required high-speed serial clock for the interface is sourced from IO input, and the low-speed parallel clock is generated by clock propagation division.	Cleared
	Enable TX PLL APB Bus[9]	Enable TX PLL APB configuration bus options.	Cleared
	Enable edge-aligned txclk_out	TX high-speed channel-associated clock enable option.	Cleared
	Enable edge-aligned txclk_out(Divided version)	TX channel-associated divided clock enable option.	Cleared
	TX Clkin Frequency (MHz)	TX input clock frequency configuration;	400.000

4 When "Work Mode" is not "TX-RX-LOOP" mode, "Enable RX Training]" and "Clock Pattern" are hidden by default and are cleared.

5 When "Enable RX Training" is selected, the RX PLL is enabled by default, and the dynamic delay mode of the RX Data/CLK Path is also enabled by default.

6 Configuration is not supported if "Enable RX Training" is cleared.

7 In DDR mode, word alignment is not supported when the serialization factor is configured to 2.

8 Configuration is not supported if "Enable RX Word-align" is cleared.

9 Configuration is not supported if "Enable TX PLL" is cleared.

Option Domain	Parameter/Configuration Options	Parameter Description	Default Value
	TX Clkin Option	TX input clock IO transmission type selection: 1) Single ended Pin: single-end clock input; 2) Differential ended Pin: differential clock input.	Differential ended Pin
	TX Clkin IO Standard	TX input clock IO level standard selection: Single-end levels: SSTL18_I, SSTL15_I, LVCMOS25, LVCMOS18, LVCMOS15; Differential levels: LVDS25, MINI-LVDS; Note: When the TX input clock frequency is higher than 300 MHz, options LVCMOS25, LVCMOS18, LVCMOS15 cannot be configured.	LVDS25
	TX Clkout Option[10]	TX output channel-associated clock transmission type selection: 1) Single ended Pin: single-end clock output; 2) Differential ended Pin: differential clock output;	Differential ended Pin
	TX Clkout IO Standard[10]	TX output clock IO level standard selection: Single-end levels: SSTL18_I, SSTL15_I; Differential levels: LVDS25, MINI-LVDS;	LVDS25
RX Clocking Options	Enable RX PLL	Configuration for RX PLL enable: Selected: The high-speed serial clock and low-speed parallel clock for the interface are generated by the PPLL; Cleared: The high-speed serial clock of the interface is sourced from IO input, and the low-speed parallel clock is generated by clock propagation division.	Cleared
	Enable RX PLL APB Bus[11]	Enable RX PLL APB configuration bus.	Cleared
	RX Clkin Frequency (MHz)	RX input clock frequency configuration;	400.000
	RX Clkin Option	RX input clock transmission type selection: 1) Single ended Pin: single-end clock input; 2) Differential ended Pin: differential clock input.	Differential ended Pin
	RX Clkin IO Standard	RX input clock IO level standard selection: Single-end levels: SSTL18_I, SSTL15_I; Differential levels: LVDS25, MINI-LVDS.	LVDS25

10 Configuration is not supported if the TX channel-associated clock output is disabled.

11 Configuration is not supported if "Enable RX PLL" is cleared.

Option Domain	Parameter/Configuration Options		Parameter Description	Default Value
IO Settings	Enable RX IODELAY Cascade		Set whether to enable RX IODELAY cascading: Cleared: The adjustable delay range of RX signal path is 0~1235 ps; Selected: The adjustable delay range of RX signal path is 0~2,470 ps.	Cleared
	TX Data Path	Delay Mode	TX data path delay mode setting: 1) No Delay: no additional path delay; 2) Fixed Delay: fixed path delay; 3) Dynamic Delay: dynamically controls the path delay.	No Delay
		Delay Setting(ps)	Set the TX data path delay, in ps, in the Fixed Delay mode; Note: It can be configured only in the Fixed Delay mode.	0
	TX Clkout Path	Delay Mode[12]	TX channel-associated clock output path delay mode settings: 1) No Delay: no additional path delay; 2) Fixed Delay: fixed path delay.	No Delay
		Delay Setting(ps)[12]	Set the TX channel-associated clock output path delay value, in ps, in the Fixed Delay mode; Note: It can be configured only in the Fixed Delay mode.	0
	RX Data Path	Delay Mode	RX data path delay mode setting: 1) No Delay: no additional path delay; 2) Fixed Delay: fixed path delay; 3) Dynamic Delay: dynamically controls the path delay.	No Delay
		Delay Setting(ps)	Set the RX data path delay, in ps, in the Fixed Delay mode; Note: It can be configured only in the Fixed Delay mode.	0
		Control Mode	RX data path delay control settings: Common control: The path delay levels of all RX data channels are adjusted together. Per channel control: The path delay levels of each RX data channel can be adjusted individually. Note: This option can be configured only in the Dynamic Delay mode.	Common control
	RX CLK Path	Delay Mode	RX input clock path delay mode settings: 1) No Delay: no additional path delay; 2) Fixed Delay: fixed path delay; 3) Dynamic Delay: dynamically controls the path delay.	No Delay
		Delay Setting(ps)	Set the RX input clock path delay, in ps, in the Fixed Delay mode; Note: It can be configured only in the Fixed Delay mode.	0

12 Configuration is not supported if the TX channel-associated clock output is disabled.

Option Domain	Parameter/Configuration Options	Parameter Description	Default Value
	Data Bus Option	Data transfer type selection: 1) Single-ended Pin: Single-ended data signal transmission; 2) Differential ended Pin: Differential data signal transmission.	Differential ended Pin
	Data Bus IO Standard	Level standard selection for data transfer: Differential levels: LVDS25, MINI-LVDS.	LVDS25
	IDDR Align Setting[13]	Input DDR data alignment type settings: 1) DDR1TO2_SAME_EDGE; 2) DDR1TO2_OPPOSITE_EDGE; 3) DDR1TO2_SAME_PIPELINED.	DDR1TO2_SAME_EDGE
	ODDR Align Setting[13]	Output DDR data alignment type settings: 1) DDR2TO1_SAME_EDGE; 2) DDR2TO1_OPPOSITE_EDGE.	DDR2TO1_SAME_EDGE

2.3.1.2.2 Pin Settings Page

The Pin Settings page is used to configure the pin positions for the data bus signals and clock signals of the interface. The page is shown in Figure 2-6. For configuration parameter details, please refer to Table 2-3.

13 It can be configured only when "Transmission Type" is DDR and "SerDes Ratio" is 2. For detailed information about data alignment types, please refer to "***UG040006_Logos2 Family FPGAs Input/Output Interface (IO) User Guide***", "***UG050006_Titan2 Family FPGAs Input/Output Interface (IO) User Guide***", and "***UG100006_Kosmo2 Family SoPCs Input/Output Interface (IO) User Guide***".

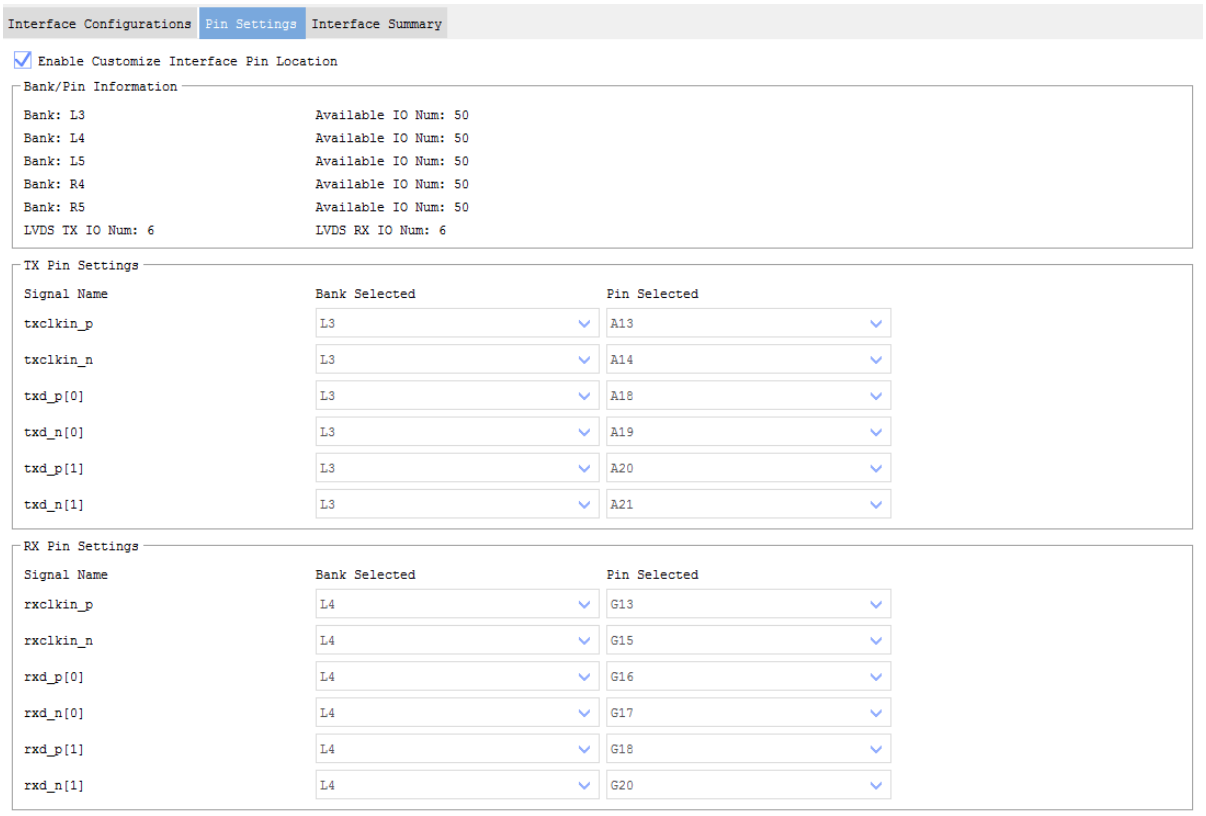


Figure 2-6 Pin Settings Page

Table 2-3 Pin Settings Parameter Description

Option Domain	**Parameter/Configuration Options**	**Parameter Description**	**Default Value**
Pin Settings	Enable Customize Interface Pin Location	Enable the configuration of interface pin positions: Enable: The user needs to allocate the interface pins; Disable: IP automatically allocates pins as examples, but it cannot be directly used in projects.	Cleared
Bank/Pin Information	Bank	List of IO banks available for the interface under the current package.	-
	Available IO Num	Number of general-purpose IOs in each bank under the current package.	-
	LVDS TX IO Num	Number of IOs required by LVDS interface TX.	-
	LVDS RX IO Num	Number of IOs required by LVDS interface RX.	-
TX Pin Settings	Bank Selected	Select the IO bank where the TX pin is located.	-
	Pin Selected	Select the pin corresponding to the TX signal.	-
RX Pin Settings	Bank Selected	Select the IO bank where the RX pin is located.	-
	Pin Selected	Select the pin corresponding to the RX signal.	-

Note: The "-" in the table indicates that the IP automatically sets the initial value based on the device and package.

Pin configuration information is stored as textual constraints in the <project_path>/pnr/core_only/*.fdc constraint files.

Descirption:

Users need to select the appropriate IO bank based on the IO resource information provided in the Bank/Pin Information to avoid insufficient IO resources.

Users can skip the Pin Settings configuration page and set the interface pins in the UCE.

2.3.1.2.3 Summary Page

The Summary page outlines the interface configuration information and some resource statistics, and users cannot configure parameters on this page. The page is shown in Figure 2-7, and the parameter description is shown in Table 2-4.

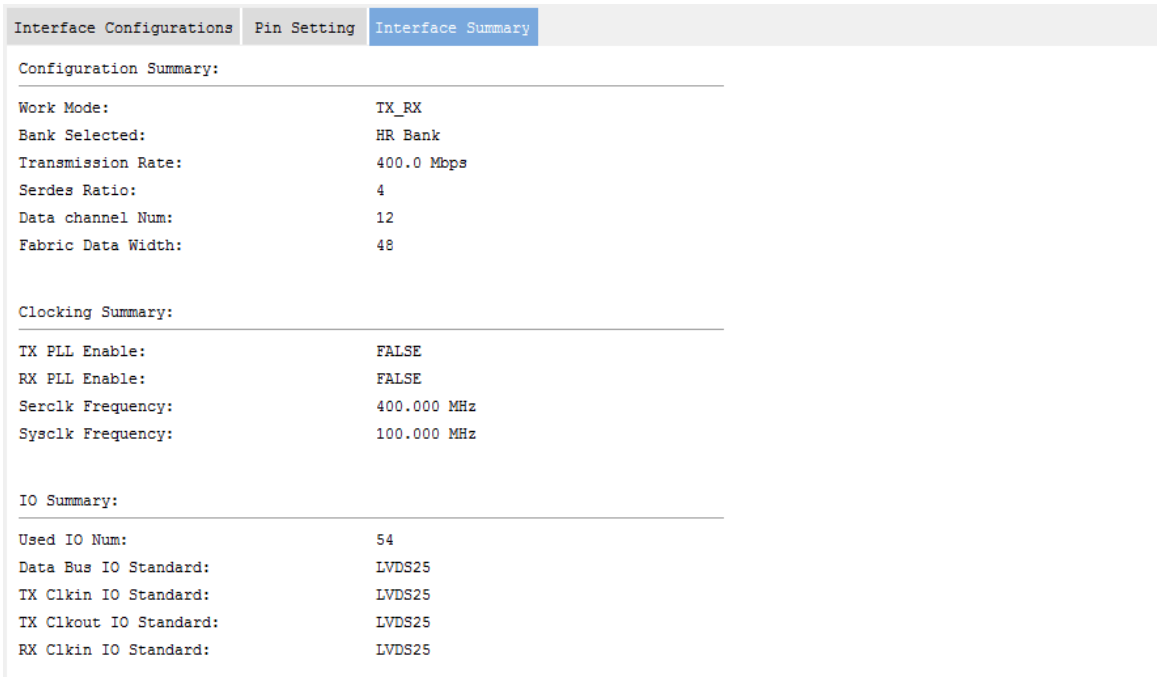


Figure 2-7 Summary Page

Table 2-4 Summary Parameter Description

Information Area	Parameter	Description
Configuration Summary	Work Mode	LVDS interface operating mode
	Bank Selected	Selected IO bank type
	Transmission Rate	Interface transmission rate
	Serdes Ratio	Serialization factor
	Data Channel Num	Number of interface data channels
	Fabric Data Width	Interface parallel data width
Clocking Summary	TX PLL Enable	TX PLL enable
	RX PLL Enable	RX PLL enable
	Serclk Frequency	High-speed serial clock frequency
	Sysclk Frequency	Low-speed parallel clock frequency
IO Summary	Used IO Num	Total number of IOs used in the interface
	Data Bus IO Standard	Data bus IO level standard
	TX Clkin IO Standard	TX input clock IO level standard
	TX Clkout IO Standard[14]	TX channel-associated clock IO level standard
	RX Clkin IO Standard	RX input clock IO level standard

Note: The parameters on the Summary page are refreshed and displayed according to user configuration.

2.3.1.3 Generating IP

Upon completion of parameter configuration, click the <Generate> button in the upper left corner to generate LVDS IP codes corresponding to user-specific settings. The information report interface for IP generation is shown in Figure 2-8.

Figure 2-8 LVDS IP Generation Report Interface

14 This option is not displayed when the TX channel-associated clock function is disabled.

Upon successful IP generation, the files indicated in the following table will be output under the project path specified in Figure 2-3.

Table 2-5 Output Files after LVDS IP Generation

Directories/Files	Subdirectories/Files	Description
--/example_design/	--bench	Contains IP simulation top-level files.
	--rtl	Contains Example Design and files for interface simulation.
--/pnr/	--core_only	Contains the project file .pds for IP core and the pin constraint file. fdc.
	--example_design	Contains the project file .pds for IP Example Design and the pin constraint file .fdc.
--/rtl/	--ipsxe_lvds_xxx	RTL code file generated by IP.
	--ipsxe_lvds_rst	The RTL code directory of the IP's reset sequence.
--/sim/	--modelsim	IP simulation directory. sim.bat is the modelsim simulation batch script, lvds_top_filelist.f is the simulation file list, and lvds_top_*.do are the do files used for simulation and loading waveforms.
--/sim_lib/	--	The directory of the encryption files required for IP. Not for LVDS IP.
<instance_name>.idf	--	Configuration file of IP.
<instance_name>.v	--	Top-level .v file of IP.
readme.txt	--	File directory and hierarchy description.

2.3.2 Constraint Configuration

For the specific configuration method of constraint files, please refer to the relevant help documents in the PDS installation path: "***User_Constraint_Editor_User_Guide***", "***Physical_Constraint_Editor_User_Guide***", "***Route_Constraint_Editor_User_Guide***".

2.3.3 Simulation Runs

Attention:

When users perform system simulation using the generated IP design, it is required that the transmitter side and the receiver side use the same source clock or source synchronous clock; otherwise, simulation failures may occur due to clock frequency offset or simulator error accumulation.

Ensure the setup and hold times for RX data sampling.

Under the Windows operating system, after the IP is generated, simulation can be run by double-clicking the sim.bat file in the "<project_path>/sim/modelsim" path.

The simulation of LVDS IP is based on the test bench of Example Design. For detailed information about Example Design, please refer to "2.4 Example Design".

For more details about the PDS simulation functions and third-party simulation tools, please consult the related help documents in the PDS installation path: "***Pango_Design_Suite_User Guide***", "***Simulation_User_Guide***".

2.3.4 Synthesis and Placement/Routing

The specific usage of PDS synthesis tools and placement/routing tools can be found in the help documents within the PDS installation path.

Attention:

Example Design project files .pds and pin constraint files .fdc generated by the IP are located in the "/pnr/example_design" directory, and physical constraints need to be modified according to the actual devices and PCB trace routing. For details, please refer to "2.8 Descriptions and Considerations".

2.3.5 Resources Utilization

Table 2-6 Typical Resource Utilization Values for LVDS IP Based on Applicable Devices

Device	Operating Mode (DDR 14:1)	Number of Channels	Word Boundary Alignment	Typical Resource Utilization Values					
				LUT	FF	GPLL	PPLL	USCM	IO
PG2L50H	TX	20	NA	86	70	1	0	1	44
	RX	20	Enabled	1020	641	0	1	1	42
	TX_RX	20	Enabled	1094	710	1	1	1	86
	TX_RX_LOOP	20	Enabled	1094	710	1	1	1	86
PG2L200H	TX	20	NA	83	71	1	0	1	44
	RX	20	Enabled	1016	644	0	1	1	42
	TX_RX	20	Enabled	1087	715	1	1	1	86
	TX_RX_LOOP	20	Enabled	1087	715	1	1	1	86

Device	Operating Mode (DDR 14:1)	Number of Channels	Word Boundary Alignment	Typical Resource Utilization Values					
				LUT	FF	GPLL	PPLL	USCM	IO
PG2T390HX	TX	20	NA	70	63	1	0	1	44
	RX	20	Enabled	1002	637	0	1	1	42
	TX_RX	20	Enabled	1080	708	1	1	1	86
	TX_RX_LOOP	20	Enabled	1083	707	1	1	1	86
PG2K400	TX	20	NA	83	67	1	0	1	44
	RX	20	Enabled	1094	640	0	1	1	42
	TX_RX	20	Enabled	1169	707	1	1	1	86
	TX_RX_LOOP	20	Enabled	1169	707	1	1	1	86
PG2T70H	TX	20	NA	72	67	1	0	1	44
	RX	20	Enabled	1081	637	0	1	1	42
	TX_RX	20	Enabled	1152	704	1	1	1	86
	TX_RX_LOOP	20	Enabled	1129	711	1	1	1	86
PG2T160H	TX	20	NA	69	67	1	0	1	44
	RX	20	Enabled	996	637	0	1	1	42
	TX_RX	20	Enabled	1070	704	1	1	1	86
	TX_RX_LOOP	20	Enabled	1071	711	1	1	1	86

Note: NA indicates that this function is not applicable.

2.4 Example Design

This section mainly introduces the Example Design scheme of LVDS IP. This scheme includes a data generation module and a data check module for testing the basic functions of the interface. Example Design provides an LVDS timing training module and a word boundary alignment state machine, which enables receive timing adjustment and data word boundary alignment, respectively. Users can instantiate them separately according to their needs in actual applications.

2.4.1 Design Block Diagram

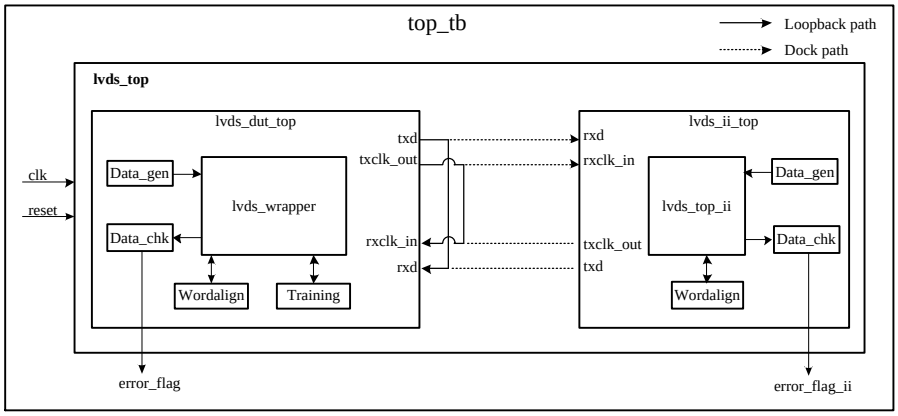


Figure 2-9 Example Design Structure Block Diagram

Example Design is divided into two modes based on the interface types: dock mode and loopback mode, and the design block diagram is shown in Figure 2-9.

2.4.1.1 Dock Mode

When the IP operating mode is configured as TX, RX, and TX_RX, the IP automatically generates a mirror test interface that is only used for simulation, facilitating dock simulation for users.

2.4.1.2 Loopback Mod

When the IP operating mode is configured as TX_RX_LOOP, the data from the interface transmitter will loop back to the receiver. In this mode, the Example Design includes word alignment logic and timing training logic by default, facilitating testing.

2.4.2 Module Description

2.4.2.1 top_tb

Test bench used for interface simulation.

2.4.2.2 lvds_top

The top level of interface simulation.

- When the IP operating mode is configured as TX, RX, and TX_RX, the lvds_dut_top module and the lvds_ii_top module perform transmit-receive docking tests.
- When the IP operating mode is configured as TX_RX_LOOP, the lvds_dut_top module performs data loopback tests.

2.4.2.3 lvds_dut_top

The reference design provided by the IP includes the top level of interface (lvds_wrapper) generated by user configuration, the data generation module (Data_gen), the data check module (Data_chk), the word boundary alignment module (Word-align), and the timing training module (Training).

2.4.2.4 lvds_wrapper

The interface top level configured and generated by the user through the interface.

2.4.2.5 lvds_top_ii

A docking test module used to achieve basic data transmit/receive functions. This module is used only for interface simulation, not for actual design purposes.

2.4.2.6 Training Module

Attention:

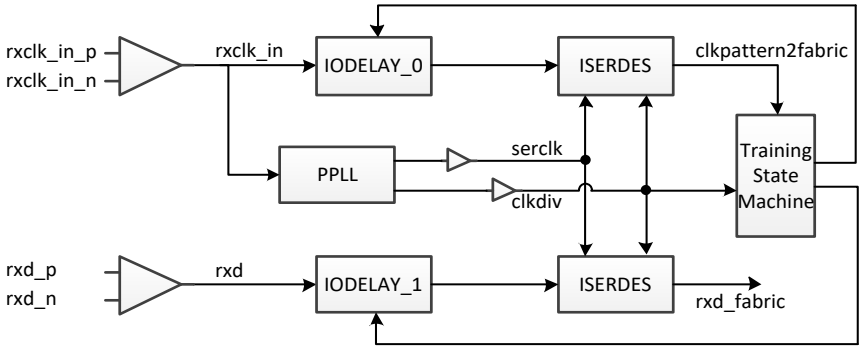
This Training module can only be applied in scenarios where the parameter "Work Mode" is set to "TX_RX_LOOP".

In the source synchronous interface, to ensure that the channel-associated clock and data have the same transmission delay, the PCB design requires equal trace lengths. Additionally, packaging delay differences should be considered, and simulation estimation is recommended. For detailed PCB design, please refer to "***UG040012_Logos2 Single Board Hardware Design Guide***" or "***UG050012_Titan2 Single Board Hardware Design Guide***" based on the selected device type.

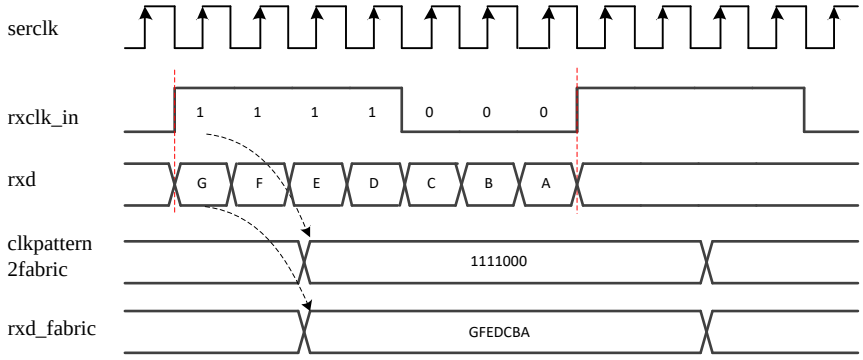
The Training module provides the timing training logic for the receiver, and can automatically adjust the phase between the sampling clock and the data window at the receiver.

The block diagram of scheme and timing diagram for the timing training at the receiver are shown in Figure 2-10, and the functions of each hardware primitive module in the diagram can be referred to in "***UG040007_Logos2 Family Product GTPs User Guide***", "***UG050007_Titan2 Family Product GTPs User Guide***" or "***UG100007_Kosmo2 Family Product GTPs User Guide***". The channel-associated division clock rxclk_in is multiplied by the PPLL to produce the serclk for high-speed sampling and the parallel clock named clkdiv. Under the condition that PCB trace lengths are consistent, theoretically, the channel-associated division clock rxclk_in and the transmit data rxd have the same phase relationship. The Training module uses the phase of rxclk_in as a reference for timing training, which mainly includes three steps:

- The state machine adjusts the delay levels of IODELAY_0 in the rxclk_in transmission path and monitors changes in the clkpattern2fabric values to find a stable sampling window and boundary.
- Once the stable sampling window boundary is determined, the state machine calculates the levels of IODELAY-0 corresponding to the optimal sampling point.
- Reload the value of delay level of IODELAY obtained from the training process into IODELAY_0 and IODELAY_1, which respectively in the clock path and data path, to ensure the stability of data sampling and complete the interface timing training.



(a) Block diagram of scheme



(b) Training timing in 1:7 mode

Figure 2-10 Simplified Diagram of Training Implementation

2.4.2.7 Word-align Module

Word-align module serves as a word boundary alignment state machine by sending shift control pulses to the bit order adjustment module (aligner) to enable word boundary alignment.

2.4.2.7.1 Data Pattern Implementation

When using Data Pattern to achieve word boundary alignment, the user needs to configure a specific data alignment sequence in the "Align Pattern" option on the IP configuration interface. During word boundary alignment, TX needs to continuously transmit the Data Pattern until the word boundary alignment is complete.

The Word-align module monitors the received parallel data and compares it with the Data Pattern. If

there is a mismatch, it generates bitslip pulses to adjust the data boundary. The Aligner module adjusts the bit order of the parallel data in a specific direction based on the received bitslip pulses. When the alignment flag signal align_done is pulled high, it indicates that word boundary alignment is completed.

2.4.2.7.2 Clk Pattern Implementation

When the serialization factor is 7, the Word-align module uses the Data Pattern for word boundary alignment by default, but the Clk Pattern can also be used. At this time, the "Align Pattern" in the IP configuration interface can only be set to "1111000" or "0001111," and the user needs to modify the parameter "PATTERN_SELECT" in *_dut_top.v to switch the implementation method.

The timing diagram for Clk Pattern to achieve word boundary alignment is shown in Figure 2-11. Because the channel-associated division clock and the serial data maintain definite boundary information, at the receiver, the channel-associated division clock can be deserialized into a clock sequence (clkpattern2fabric). The data channel and clock channel are adjusted by the same bitslip pulse to align the parallel data bit order. When clkpattern2fabric at the receiver is adjusted to "1111000," the clock channel boundary is aligned, and accordingly, the data also completes word boundary alignment.

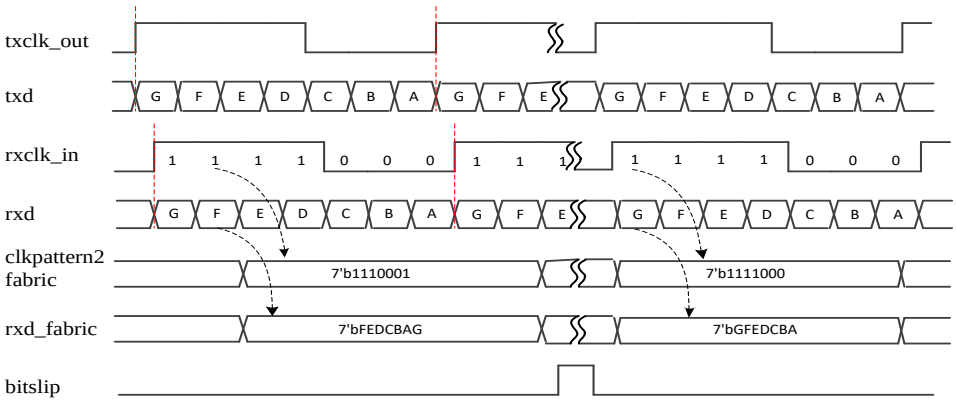


Figure 2-11 Word Alignment Timing for a Serialization Factor of 7

2.4.2.7.3 bitslip Timing

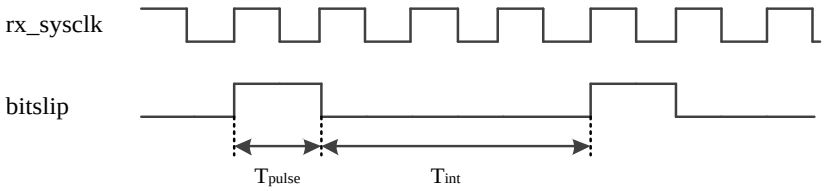


Figure 2-12 Word Alignment Control Pulse Timing

The bitslip timing of the shift control pulse of the bit order adjustment module (aligner) is shown in Figure 2-12. Users can design their own word boundary alignment logic according to the timing requirements to meet different application needs.

- The bitslip pulse width Tpulse is greater than or equal to one RX system clock cycle;
- The interval Tint between two adjacent bitslip pulses is greater than three RX system clock cycles.

2.4.2.8 data_gen Module

The data_gen module sends parallel data with the corresponding data width according to the serialization factor and the number of channels. The transmit data patterns include pseudo-random code, accumulated code, and fixed code. The transmit data patterns can be adjusted by modifying parameters, with the default being the pseudo-random code.

- When the Word-align function is enabled, the data_gen module first sends a Data Pattern for interface word boundary alignment. After the word alignment flag align_done is pulled high, it begins to send valid data;
- When the Word-align function is disabled, the data_gen module sends valid data immediately after the interface initialization is completed.

2.4.2.9 data_chk Module

The data_chk module checks the receive data. The data verification pattern can be adjusted by modifying parameters and must be consistent with the transmit data pattern on the transmitter side.

- When the Word-align function is enabled, valid data is not checked before the word alignment flag align_done is pulled high. After the word boundary alignment is complete and data reception is stable, the data_chk module starts to check the received data and outputs the error status;
- When the Word-align function is disabled, the data_chk module starts to check the receive data and outputs the error status immediately after the interface initialization is completed.

2.4.3 Instance Simulation

Example Design supports basic interface simulation. The detailed simulation test process is shown in Figure 2-13. Taking TX_RX_LOOP mode as an example, the interface simulation mainly includes the following steps:

- After the system reset is completed, the LVDS IP starts the initialization process.
- Wait for the interface initialization to complete, that is when the transmitter-side initialization completion flag tx_init_done becomes 1 and the receiver-side initialization completion flag rx_init_done becomes 1.
- The Training module performs timing training, and the status signal train_done being pulled high marks the completion of timing training on the receiver.
- The data_gen module sends the Align pattern, and the Word-align module performs word boundary alignment.
- After completing word boundary alignment, the data_gen module begins to send valid data.
- After the data reception is stable, data_chk performs verification on the data and outputs the error status indicator.

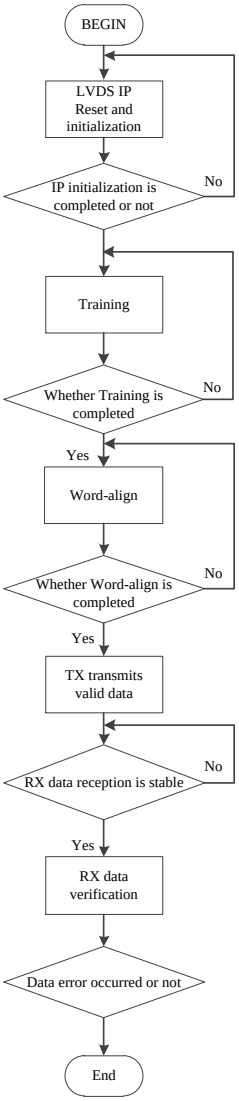


Figure 2-13 Example Design Test Flowchart

In TX, RX, and TX_RX modes, the IP interface performs docking simulation with the automatically generated mirror interface, and the simulation steps are consistent with those in the TX_RX_LOOP mode.

Attention:

Do not directly use the Example Design generated by the IP for running flow and on-board testing. Constrain pins according to the actual pin connections of the single board, then proceed with running flow and on-board testing.

2.5 Descriptions of IP Interfaces

This section provides the LVDS IP related interface instructions and timing descriptions.

2.5.1 Reset Sequence Interface Description

Table 2-7 Reset Sequence Interface List

Port	I/O	Bit width	Description
free_clk	I	1	The clock input of the reset sequence module must have a frequency lower than 100 MHz.
txrst_n	I	1	Interface TX reset signal; 0: Reset; 1: Reset release.
rxrst_n	I	1	Interface RX reset signal; 0: Reset; 1: Reset release.
tx_init_done	O	1	Flag signal for the completion of TX initialization of interface.
rx_init_done	O	1	Flag signal for the completion of RX initialization of interface.

2.5.2 TX Interface Description

Table 2-8 TX Interface List

Port	I/O	Bit width	Description
txclk_in	I	1	Single-ended clock input.
txclk_in_p	I	1	Differential clock P-side input.
txclk_in_n	I	1	Differential clock N-side input.
txd_fabric	I	lane_num *ratio	Parallel data input.
txd_delayctrl	I	8	The control signal for the IODELAY delay levels in the TX data path, which is a binary code; where [7:2] is for coarse delay adjustment, and [1:0] is for fine delay adjustment. For delay configuration, please refer to "2.8.1.3 Feedback Delay Configuration".
txd _p	O	lane_num	Differential data P-side output.
txd _n	O	lane_num	Differential data N-side output.
txclk_out	O	1	Differential channel-associated clock single-end output.
txclk_out_p	O	1	Differential channel-associated clock P-side output.
txclk_out_n	O	1	Differential channel-associated clock N-side output.
tx_sysclk	O	1	TX system clock, used to drive user logic.

Note: "lane_num" refers to the number of channels, and "ratio" refers to the serialization factor.

2.5.3 RX Interface Description

Table 2-9 RX Interface List

Port	I/O	Bit width	Description
rxd	I	lane_num	Single-ended serial data input.
rxd_p	I	lane_num	Differential serial data P-side input.
rxd_n	I	lane_num	Differential serial data N-side input.
rxclkin	I	1	Single-ended channel-associated clock input.
rxclkin_p	I	1	Differential channel-associated clock P-side input.
rxclkin_n	I	1	Differential channel-associated clock N-side input.
rxd_fabric	O	lane_num*ratio	Parallel data output.
rxd_delayctrl	I	8 or 8*lane_num	The control signal for the IODELAY delay levels in the RX data path, which is a binary code; When the delay for each channel is adjusted individually: the delay level for the i-th channel corresponds to rxd_delayctrl[8*i+7:8*i]; When the delay for each channel is adjusted collectively: the delay level for all channels is rxd_delayctrl[7:0]; In the 8-bit delay level control signal of each channel, [7:2] is for coarse delay adjustment and [1:0] is for fine delay adjustment. For delay configuration, please refer to "2.8.1.3 Feedback Delay Configuration".
rxclk_delayctrl	I	8	The control signal for the IODELAY delay levels in the RX clock path, which is a binary code; where [7:2] is for coarse delay adjustment, and [1:0] is for fine delay adjustment. For delay configuration, please refer to "2.8.1.3 Feedback Delay Configuration".
bitslip	I	1	The bit-shift impulse signal is used for word boundary alignment.
rxd_fabric	O	lane_num*ratio	Parallel data output.
clkpattern2fabric	O	ratio	Taking the channel-associated clock as a data sequence, the derived clock sequence is obtained through deserialization.
rx_sysclk	O	1	RX system clock, used to drive user logic.

Note: "lane_num" refers to the number of channels, and "ratio" refers to the serialization factor.

2.5.4 APB Interface Description

When enabling the PLL for the LVDS IP, the user logic can be used to dynamically configure the operating parameters of both the TX PLL and RX PLL through the APB interface.

Table 2-10 APB Interface List

Port	I/O	Bit width	Description
pll_apb_clk	I	1	APB clock signal.
pll_apb_en	I	1	APB bus enable signal: 0: Disabled; 1: Enabled.
pll_apb_sel	I	1	APB selection signal. 0: Indicates that the slave device is selected; 1: Indicates that the slave device is not selected.
pll_apb_rstn	I	1	APB reset signal: 0: Reset; 1: Reset release.
pll_apb_write	I	1	APB interface read and write control signal: 0: Read operation; 1: Write operation.
pll_apb_ready	O	1	APB interface handshake signal: 0: Indicates that a normal bus cycle has not been completed; 1: Indicates the end of a normal bus cycle.
pll_apb_addr	I	5	APB interface address signal.
pll_apb_rdata	O	16	APB interface read data output signal
pll_apb_wdata	I	16	APB interface write data input signal.

The mapping relationship between the IP APB interface address and the TX PLL and RX PLL addresses is shown in Table 2-11. For detailed information on the PLL APB interface timing and PLL APB register, please refer to "***UG040004_Logos2 Family FPGAs Clock Resources (Clock) User Guide***", "***UG050004_Titan2 Family FPGAs Clock Resources (Clock) User Guide***", and "***UG100004_Kosmo2 Family SoPCs Clock Resources (Clock) User Guide***".

Table 2-11 APB Address Mapping Relationship List

LVDS APB_ADDR	TX PLL	RX PLL
apb_addr[6]	0	1
apb_addr[5:1]	txpll_addr[4:0]	rxpll_addr[4:0]
apb_addr[0]	Reserved	Reserved

2.6 IP Register Description

None.

2.7 Typical Applications

For typical applications of LVDS IP, please refer to "2.4 Example Design".

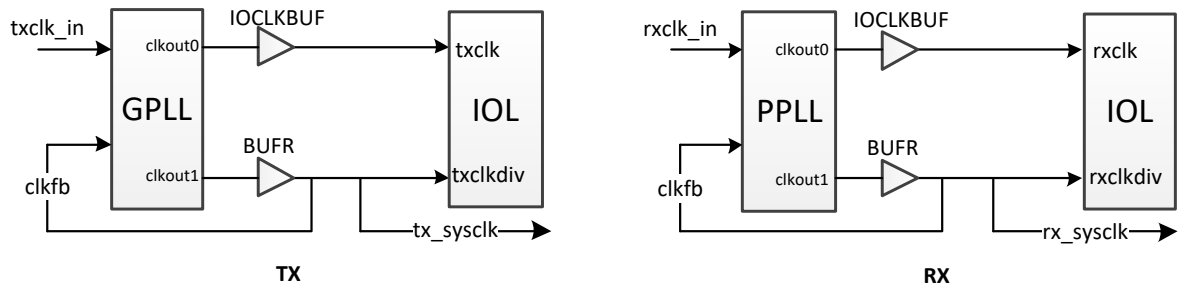
2.8 Descriptions and Considerations

2.8.1 Configuration Descriptions

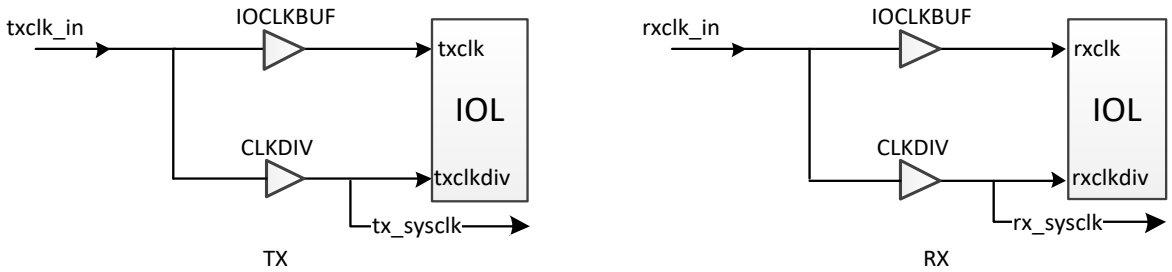
2.8.1.1 PLL Enable

The "Enable TX PLL" and "Enable RX PLL" options allow you to configure whether the PLL is enabled for the TX and RX interfaces, respectively. The clock structures for the two cases are shown in Figure 2-14.

- When the PLL is enabled, the high-speed serial clock and parallel clock for the interface are provided by the PLL.
- When the PLL is disabled, the high-speed serial clock must enter the clock network through a dedicated global clock pin (GMCLK), and the parallel clock is generated by dividing the high-speed clock.



(a) Enable PLL



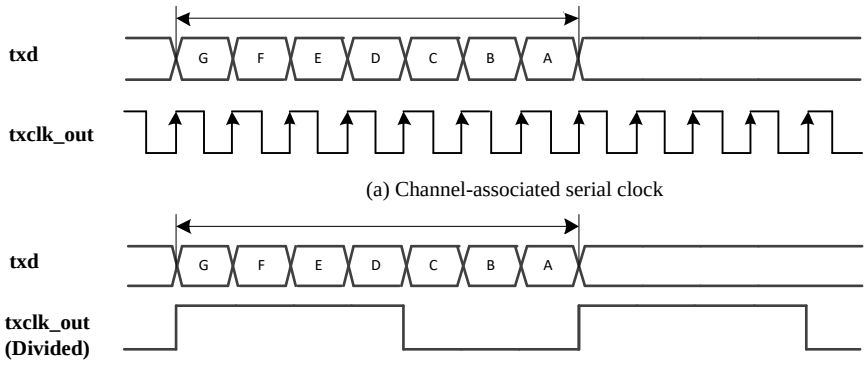
(b) Disable PLL

Figure 2-14 Interface Internal Clock Structure

2.8.1.2 Enable Channel-Associated Clock Output

TX supports transmitting two types of channel-associated clocks, with example timing shown in Figure 2-15.

- When the "Enable edge-aligned txclk_out" option is selected, TX transmits the channel-associated serial clock;
- When the "Enable edge-aligned txclk_out (Divided version)" option is selected, TX transmits the channel-associated parallel clock.



(a) Channel-associated serial clock

(b) Channel-associated parallel clock

Figure 2-15 Timing Diagram of Channel-Associated Clock in 7:1 Mode

2.8.1.3 Feedback Delay Configuration

Allows users to configure the signal delay of the input/output path. The delay configuration types supported by IP under different modes are shown in Table 2-12.

Table 2-12 List of Interface Delay Configuration Types

Interface Orientation	Delay Path	Delay Type	Delay Range[15]
TX	Data transmission path	1) No Delay; 2) Fixed Delay; 3) Dynamic Delay.	Delay range: 0~635 ps.
	Channel-associated clock sending path	1) No Delay; 2) Fixed Delay.	

15 For detailed information, please refer to "***UG040006_Logos2 Family FPGAs Input/Output Interface (IO) User Guide***", "***UG050006_Titan2 Family FPGAs Input/Output Interface (IO) User Guide***", and "***UG100006_Kosmo2 Family SoPCs Input/Output Interface (IO) User Guide***".

Interface Orientation	Delay Path	Delay Type	Delay Range[15]
RX	Data reception path	1) No Delay; 2) Fixed Delay; 3) Dynamic Delay.	Cascading disabled: Delay range is 0~1,235 ps. Cascading enabled: Delay range is 0~2,470 ps.
	Clock input path	1) No Delay; 2) Fixed Delay; 3) Dynamic Delay.	

Note: The receiver supports IODELAY cascading, configured by the "Enable RX IODELAY Cascade" option.

2.8.2 Clock Constraints

LVDS IP has multiple clocks that require constraints, namely free_clk, apb_clk, txclkdiv, and rxclkdiv. Among them, free_clk is the reset sequence reference clock, apb_clk is the APB bus clock of the IP, and txclkdiv and rxclkdiv are the interface system clocks. In applications, clock attributes can be constrained through the PDS software or by editing the clock attributes in the .fdc file.

Users need to pay attention to whether the clock path after PDS routing meets the design intention. Specific constraints can refer to the .fdc file under the "IP/pnr/example design" directory, the clock constraint example is as follows:

create_clock -name {free_clk} [get_ports {free_clk}] -period {10.000} -waveform {0.000 5.000}

create_clock -name {txclk_in} [get_ports {txclkin_p}] -period {10.000} -waveform {0.000 5.000}

create_generated_clock -name {txclk} -source [get_ports {txclkin_p}] [get_nets {u_lvds_inst.TX_INST.u_tx_top.u_txclkgen.txclk}] -master_clock [get_clocks {txclk_in}] –multiply_by {12} -divide_by {3}

create_generated_clock -name {txclkdiv} -source [get_ports {txclkin_p}] [get_nets {u_lvds_inst.TX_INST.u_tx_top.u_txclkgen.txclkdiv}] -master_clock [get_clocks {txclk_in}] -multiply_ by {12} -divide_by {12}

2.8.3 Physical Constraints

In applications, physical constraints need to be applied to GPLL, PPLL, BUFR, and input clock pins based on the actual IO bank used. The physical positions can be constrained through PDS software or by editing the physical positions in .fdc file.

Taking the Example Design in this document as an example, the location constraints of GPLL and BUFR are as follows by editing .fdc file:

define_attribute {i:u_lvds_inst.TX_INST.u_tx_top.u_txclkgen.GPLL_CLKNETWORK.U_GPLL} {PAP_LOC} {GPLL_667_157}

define_attribute {i:u_lvds_inst.TX_INST.u_tx_top.u_txclkgen.GPLL_CLKNETWORK.u_bufr0} {PAP_LOC} {RCKB_663_151}

Without enabling the PLL, the IP requires the input clock of the interface to be constrained to the global clock pin GMCLK. The location constraints of the clock signal can be set by editing the .fdc file as follows:

define_attribute {p:txclkin_p} {PAP_IO_DIRECTION} {INPUT}

define_attribute {p:txclkin_p} {PAP_IO_LOC} {D17}

define_attribute {p:txclkin_n} {PAP_IO_DIRECTION} {INPUT}

define_attribute {p:txclkin_n} {PAP_IO_LOC} {C17}

2.8.4 Power Constraints

When using the SSTL18_I and SSTL15_I level standards, the reference voltage and termination resistor should be configured in the UCE. For IO usage instructions, please refer to the specifications in "***UG040006_Logos2 Family FPGAs Input/Output Interface (IO) User Guide***", "***UG050006_Titan2 Family FPGAs Input/Output Interface (IO) User Guide***", or "***UG100006_Kosmo2 Family SoPCs Input/Output Interface (IO) User Guide***" based on the device. The corresponding .fdc reference constraints for using external reference voltage (VREF0) are given below:

define_attribute { p:port_name } {PAP_IO_DDR_TERM_MODE} {ON}

define_attribute {p:port_name} {PAP_IO_VREF_MODE} {EXTERNAL}

define_attribute {p:port_name} {PAP_IO_VREF_MODE_VALUE} {VREF0}

2.8.5 PLL Usage Instructions

Table 2-13 The Relationships Between IP Parameters and PLL Parameters

PLL Type	IP Parameter	Description
GPLL	TXPLL_IDIV	GPLL input Divider configuration.
	TXPLL_MDIV	GPLL feedback Divider configuration.
	TXPLL_ODIV0	Division parameter of GPLL clkout0; Note: GPLL clkout0 corresponds to the TX serial clock.
	TXPLL_ODIV1	GPLL clkout1 division parameter; Note: GPLL clkout1 corresponds to the TX parallel clock, and it is also used as the external feedback input clock for the GPLL.
	TXPLL_FREQIN	GPLL input clock frequency; Note: Correspond to TX reference clock input frequency.
	TXPLL_BANDWIDTH	GPLL bandwidth selection and configuration.
PPLL	RXPLL_IDIV	PPLL input Divider configuration.
	RXPLL_MDIV	PPLL feedback Divider configuration.
	RXPLL_ODIV0	PPLL clkout0 division parameter; Note: PPLL clkout0 corresponds to the RX serial clock.
	RXPLL_ODIV1	PPLL clkout1 division parameter; Note: PPLL clkout1 corresponds to the RX parallel clock and is also the external feedback input clock for the PPLL.
	RXPLL_FREQIN	PPLL input clock frequency; Correspond to RX reference clock input frequency.
	RXPLL_BANDWIDTH	PPLL bandwidth selection and configuration.

In applications with PLL enabled, it is recommended that users combine PLL IP for parameter configuration.

LVDS IP offers basic PLL parameter configuration functions, allowing for optional PLL parameter configurations based on transmission rates, transmission types, and serialization factors. If users need to enable more frequency point configurations, they can combine it with PLL IP or modify parameters through the APB bus. Table 2-13 A list of correspondence between IP parameters and PLL parameters is provided. For detailed PLL IP information, please refer to "***UG041001_PLL_IP***".

2.9 IP Debugging Methods

When using the Example Design provided by the IP for debug, pay attention to testing some top-level key indicator signals. These can be connected to external LEDs or monitored in other ways to quickly determine the interface's operating status. For descriptions of key indicator signal, please refer to Table 2-14.

Table 2-14 Key Indicator Signal

Port	I/O	Bit width	Description
tx_init_done	O	1	Flag signal for the completion of TX initialization of interface. 0: TX PHY initialization has not completed; 1: TX PHY initialization completed.
rx_init_done	O	1	Flag signal for the completion of RX initialization of interface. 0: RX PHY initialization has not completed; 1: RX PHY initialization completed.
train_done	O	1	Flag signal for the completion of interface timing training. 0: RX timing training has not completed; 1: RX timing training completed.
debug_training[17:12]	O	6	The coarse delay levels of IODELAY in the RX data path, which were obtained from the interface timing training.
align_done	O	1	Flag signal for the completion of interface word boundary alignment. 0: RX has not completed word boundary alignment; 1: RX has completed word boundary alignment.
error_flag	O	1	Interface data error flag signal. 0: Interface is error-free; 1: There is an error code in the interface.

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage license, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its family's products at any time without prior notice.