

AN 044: Aligning LVDS Clock and Data Signals

AN044-v2.1 June 2025 www.efinixinc.com

Copyright © 2025. All rights reserved. Efinix, the Efinix logo, the Titanium logo, the Topaz logo, Quantum, Trion, and Efinity are trademarks of Efinix, Inc. All other trademarks and service marks are the property of their respective owners. All specifications subject to change without notice.

Contents

Introduction	3
Titanium, Topaz, and Trion LVDS Feature Summary	3
Hardware Design	4
Understanding LVDS Serialization	4
Edge-Alignment and Center-Alignment	5
Trion FPGAs	6
Edge-Aligned Signals	
TX Edge-Aligned	0
RX Edge-Aligned	8
Center-Aligned Signals	0
TX Center-Aligned	
RX Center-Aligned	
7:1 Serialization Examples	13
	4 -
	15
Edge-Aligned Configuration	. 15
IX Edge-Aligned	. 15
KX Edge-Aligned	. 10
TX Center Aligned with Three Cleaks	. 17
TX Center-Aligned with Three Clocks	. 17
TX Center-Aligned with Three Clocks and Seft Care Clock Pattern Serializer	.20
PX Center-Aligned with Three Clocks and Solt-Core Clock Fattern Senalizer	.25
Titanium/Topaz 7:1 Serialization Examples	.23
Summary	.27
Summary	.20
Delay Calibration	31
LVDS Delay Modes	. 31
Calibrating with Dynamic Delay (Titanium/Topaz only)	32
Calibrating with Dynamic Phase Alignment (DPA) (Titanium/Tenaz only)	22
DPA Training Pattern Pequirements	30 21
DIA Italining Fallent Requirements	. 54
	.50
Calibrating with Static Delay (Trion/Titanium/Topaz)	38
Revision History	40

Introduction

Titanium, Topaz, and Trion FPGAs support LVDS TX (transmit) and RX (receive) channels for high-speed and noise-tolerant data transmission. This application note describes how to align the clock and data for LVDS interface in different applications.

Titanium, Topaz, and Trion LVDS Feature Summary

The following table summarizes the LVDS interface features in Titanium, Topaz, and Trion FPGAs related to the scope of this application note. Refer to the respective device data sheets for detailed LVDS specifications.

Table 1: Titanium/Topaz and Trion LVDS Feature Summary

Half-rate sends and receives data on both edges of the fastclk, whereas full-rate sends and receives data on the rising edge of the fastclk.

Feature Titanium/Topaz		Trion
Mode (Half-rate / Full-rate)	Half-rate and Full-rate	Half-rate only
Serialization Width	Half-rate (Even serialization width only): 10:1, 8:1, 6:1, 4:1, 2:1 Full-rate (Even and Odd serialization width except 9:1): 10:1, 8:1, 7:1, 6:1, 4:1, 3:1, 2:1	Up to 8:1, 7:1, 6:1, 5:1, 4:1, 3:1, 2:1
Delay Modes	Static Delay, Dynamic Delay, Dynamic Phase Alignment (DPA) (Full-rate only)	Static Delay Only
Dynamic Phase Alignment	Full-rate only: 1000 Mbps (C3, C4, I3, I4) 800 Mbps (C3L, C4L, I3L, I4L)	Not supported
Serialization	Half-rate or full-rate	Half-rate only
Maximum Data Rate per Channel(Speed grade: C4, I3, I4) 1,500 Mbps (Half-rate) 1,000 Mbps (Full-rate)		800 Mbps 600 Mbps (Center-Aligned)

Learn more: Refer to the device data sheet, table "Maximum Toggle Rate" for detailed data rate specifications that vary with speed grade.

Hardware Design

High-speed differential signal quality is very sensitive to the board design. You must consider trace routing, impedance matching, and crosstalk in both the schematic and PCB design when planning a design with high-speed differential signals.

Observe the following design best practices when implementing LVDS data transmission:

- Use LVDS blocks from the same side of the FPGA to minimize skew between the data lanes, and between the clock and data lanes in an LVDS interface.
 - Example: Signals sent from or received at the top-side of the chip should utilize LVDS resources located in the top I/O banks. All clock lanes and data lanes associated with the channel should use the GPIOT PN xx pins.
- Delay matching is required for all PCB wires in an LVDS transmission channel.

Note: Refer to the Package Net Length section of the Efinix **Board Design** site for the net length and propagation delay for your package.

Learn more: Refer to the Interactive Hardware Design Checklist and Guideline (LVDS Pin General Guidelines) for more information about designing hardware with LVDS in Efinix FPGAs:

- Titanium Interactive Hardware Design Checklist and Guidelines
- Topaz Interactive Harware Design Checklist and Guidelines
- Trion Interactive Hardware Design Checklist and Guidelines

You need to have access to download the Efinity software to be able to view and download board design information.

Understanding LVDS Serialization

LVDS signals are serialized differential signals that are normally sent out synchronously with a differential clock signal. Differential pairs that carry data signals are called *data lanes*, while differential pairs that carry clock signals are called *clock lanes*. Data lanes, together with the clock lanes they are synchronized with, create a *channel*. The following diagram shows how the parallel data serialized on the TX is sent over the wires, received, and deserialized back into parallel data on the RX.





(**i**)

Note: The delay block between the buffer and serializer/deserializer in the figure above has been omitted for the sake of simplicity.

With serialization, you send parallel data at width that varies in accordance with the serialization width. Both a parallel clock and a serial clock from a single PLL are needed for a LVDS block when serialization is enabled. In the figure above, the parallel clock and serial clock are named slowclk and fastclk, respectively. Slowclk is a parallel clock used to load parallel data into the serializer register of the LVDS interface. Slowclk is also the same clock domain as the input/output data received/transmitted to the LVDS interface. Fastclk, on the other hand, is a serial clock used to serialize or deserialize data.

Edge-Alignment and Center-Alignment

There are two methods for aligning the clock signal and data signal:

- Edge-alignment: Entails aligning the rising edge of the clock signal to the edge of data signal bit 0.
- Center-alignment: Involves aligning the rising edge of the clock signal to the center of data signal bit 0.

Trion FPGAs

Trion FPGAs only support half-rate LVDS serialization, in which data is sent and received on both the rising and falling edges of the serial clock.

Edge-Aligned Signals

The following sections describe the generation of edge-aligned Tx and Rx signals for Trion FPGAs.

TX Edge-Aligned

The following clocks from a single PLL source must be provided to the LVDS interface for edge-aligned signal generation:

- tx slowclk—Parallel clock LVDS TX blocks, both clock lane and data lanes.
- tx_fastclk—Serial clock for LVDS TX blocks, both clock lane and data lanes.

The frequency, phase, and tx slowclk and tx fastclk PLL settings are as follows:

- Data rate must be equal to or greater than the maximum toggle rate as stated in the data sheet.
- tx slowclk frequency = data rate / serialization width.
- tx_fastclk frequency = tx_slowclk frequency × (serialization width ÷ 2).
- tx_fastclk phase = 90° shifted with respect to tx_slowclk.
- tx_slowclk as the core feedback clock for the PLL.

The tx_fastclk and tx_slowclk relationship is given by the following equations:

- Half-rate:
 - tx fastclk = tx slowclk × serialization rate \div 2.
- Full-rate (Titanium only):
 - tx_fastclk = tx_slowclk × serialization rate.

Example:

Scenario:

- Trion Device: T120
- I/O Standard: LVDS
- Serialization Width: 8-to-1
- Data Rate: 800 Mbps

Therefore:

- tx_slowclk frequency = 800 Mbps ÷ 8 = 100 MHz
- tx fastclk frequency = 100 MHz \times (8 ÷ 2) = 400 MHz
- tx slowclk phase = 0°
- tx fastclk phase = 90°
- tx slowclk as core feedback clock signal for PLL

Figure 2: Trion TX Edge-Aligned PLL Settings Example

Serialization Width: 8-to-1; Data Rate: 800 Mbps



Note: The output divider for PLL must be either div4 or div8 when you set the clock output phase to 90°, 180°, or 270°. If you set the PLL in manual mode, then the output divider settings must be corrected to fulfill the constraint check. When using core or local feedback mode, set the F_{VCO} to above 1.6 GHz to maximize data valid window.

Figure 3: Trion TX Edge-Aligned Waveform

Serialization Width: 8-to-1

i

PLL Slow Clock 0.0°		
PLL Fast Clock 90.0°		
Core Data OUT[7:0]	a[7:0]	b[7:0]
Data Lane TXP	a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7] b[0] b[1] b[2] b[3] b[4] b[5] b[6] b[7]
Clock Lane TXP		

RX Edge-Aligned

Note: Core feedback mode is strongly recommended for PLL of LVDS RX to minimize variations of clock network delay among devices.

The following clocks from a single PLL source need to be provided to the LVDS interface for receiving edge-aligned signal:

- rx_slowclk—Parallel clock for LVDS RX blocks, data lanes.
- rx fastclk—Serial clock for LVDS RX blocks, data lanes.

The requirements for frequency, phase, and rx_slowclk and rx_fastclk settings in the PLL setup are as follows:

- Data rate = RX clock lane refclk frequency × serialization width.
- Data rate must be equal to or greater than the maximum toggle rate stated in the data sheet.
- rx_slowclk frequency = RX_clock_lane_refclk frequency.
- rx_slowclk phase = 0°.
- rx_fastclk frequency = rx_slowclk frequency × (deserialization width ÷ 2).
- rx fastclk phase = 90°.
- rx_slowclk as core feedback clock for the PLL.

Figure 4: Trion RX Edge-Aligned PLL Settings Example

Serialization Width: 8-to-1; Data Rate: 800 Mbps

PLL Clock Calculator				
PLL Clock Calculator	Manual Mode: D	Feedback Mode:	Core 🖣	• ?
		Clock 0 Frequency 100.0000 MHz Phase: 0*	Actual: 100 MHz rx_slowclk Feedback ~	
Input Frequency 100.0000 MHz Input Pin		400.0000 MHz Phase: 90°	400 MHz rx_fastclk Feedback	
	-0	Clock 2 Frequency MHz	Actual: - MHz	
pI_rx_RSTN		Locked Pin	pll_rx_LOCKED	
		Cancel	Finish	

Figure 5: Trion RX Edge-Aligned Waveform



Center-Aligned Signals

The following sections describe the generation of center-aligned Tx and Rx signals for Trion FPGAs, as well as providing examples of 7:1 serialization.

TX Center-Aligned

The following clocks from a single PLL source need to be provided to the LVDS TX interface for center-aligned signal generation:

- tx slowclk—Parallel clock for LVDS TX blocks, both clock lane and data lanes.
- tx fastclk clk—Serial clock for LVDS TX blocks, clock lane.
- tx flastclk data—Serial clock for LVDS TX blocks, data lanes.

The requirements for frequency, phase, and tx_slowclk, tx_fastclk_clk, and tx fastclk data settings in the PLL setup are as follows:

- data rate must be equal to or greater than the maximum toggle rate (600 Mbps) for centeraligned configuration on Trion FPGAs.
- tx slowclk frequency = data rate ÷ serialization width
- tx slowclk phase = 0°
- tx_fastclk_clk frequency = tx_slowclk frequency × (serialization width ÷ 2)
- tx_fastclk_data frequency = tx_slowclk frequency × (serialization width ÷ 2)
- tx_fastclk_clk phase = 135°
- tx_fastclk_data phase = 45°
- tx slowclk as core feedback clock for the PLL

Note: Trion FPGAs do not support center-aligned configuration when the serialization width is 3, and the maximum data rate is 600 Mbps on Trion FPGAs with center-aligned configuration.

Figure 6: Trion TX Center-Aligned PLL Settings Example

Serialization Width: 8-to-1; Data Rate: 600 Mbps

i



Figure 7: Trion TX Center-Aligned Waveform

Serialization Width: 8-to-1 pll_slowclk 0.0° pll_fastclk_data 45.0° pll_fastclk 135.0° core_data_OUT[7:0] data_lane_TXP clk_lane_TXP

RX Center-Aligned

The connection type for the LVDS RX clock lane is required to set to pll clkin.

The following clocks from a single PLL source need to be provided to the LVDS RX interface to receive the center-aligned signal:

- rx slowclk—Parallel clock for LVDS RX blocks, data lanes.
- rx_fastclk—Serial clock for LVDS RX blocks, data lanes.

The requirements for frequency, phase, and rx_slowclk and rx_fastclk settings in the PLL setup are as follows:

- Data rate = RX clock lane refclk frequency × serialization width
- Data rate must be equal to or greater than the maximum toggle rate stated in the data sheet
- rx slowclk frequency = RX clock lane refclk frequency
- rx_slowclk phase= 0°
- rx_fastclk frequency = rx_slowclk frequency × (deserialization width ÷ 2)
- rx fastclk phase = 90°
- rx fastclk as core feedback clock signal for the PLL

Figure 8: Trion RX Center-Aligned PLL Settings Example

Serialization Width: 8-to-1; Data Rate: 600 Mbps

PLL Clock Calculator Manual Mode: Feedback Mode: Core • (
	r Manual Mode: Deedback Mode: Core 🗸 😗
Input Frequency T5.0000 MHz T5.0000 MHZ T	ency MHz TPINO MT2 MT2 MT2 MT2 MT2 MT2 MT2 MT2
Cancel Finish	Cancel Finish

Figure 9: Trion RX Center-Aligned Waveform

Serialization Width: 8-to-1				
clk_lane_RXP				
 data_lane_RXP د[7] a[0] a[1] a[2] a[3] a[4]	a[5] a[6] a[7] <mark>b[0]</mark>	b[1] b[2] b[3] b[4]	b[5] b[6] b[7]
pll_slowclk 0.0°				
pll_fastclk 90.0° (PLL Feedback) –				
core_data_IN[7:0]	c[7:0]		a[7:0]	b[7:0]

7:1 Serialization Examples

When receiving data from Trion LVDS TX, the input clock signal is 1110000 for one data set. You have to set up the clock and PLL depending on the data alignment as described previously.

Figure 10: Trion RX Edge-Aligned Waveform (Data Input from Trion LVDS TX)



Be aware that many of the sensors available on the market send out edge-aligned LVDS TX data with a clock pattern of 1100011 (i.e., shifted clock pattern), which is different to the Trion LVDS RX clock-data lane's alignment. Therefore, each 7-bit word received on the Trion LVDS RX interface is in {<current data>[4:0], <previous data>[6:5]} shifted format. Therefore, you need an extra register stage to align the bits in each word to data [6:0] in your RTL design.

Table 2: Trion LVDS TX Serialization Configuration Summary

Serialization Width	Half/ Full- Rate	Edge-Align	Center-Align
8	Half	slowclk freqequency = data	slowclk frequency = data
7	Half	rate ÷ serialization width rate ÷ serial slowclk phase = 0° slowclk phase = 0° fastclk frequency = slowclk fastclk data frequency × (serialization width ÷ 2)	rate \div serialization width slowclk phase = 0°
6	Half		fastclk data frequency = slowclk
5	Half		frequency × (serialization width ÷ 2)
4	Half	fastclk phase = 90°	fastclk data phase = 45°
3	Half	feedback clock signal	fastclk_clk frequency = fastclk data frequency
2	Half		fastclk_clk phase = 135°
			slowclk as PLL core feedback clock signal

- slowclk—Shared parallel clock for LVDS TX clock lane and data lanes
- fastclk-Shared serial clock for LVDS TX clock lanes and data lanes
- fastclk data—Serial clock for LVDS TX data lanes
- fastclk_clk—Serial clock for LVDS TX clock lane

Serialization Width	Half/ Full- Rate	Edge-Align	Center-Align
8	Half	slowclk frequency = RX_refclk frequency	slowclk frequency = RX_refclk frequency
7	Half	slowclk phase = 0°	slowclk phase = 0°
6	Half	fastclk frequency = slowclk	fastclk frequency = slowclk
5	Half	frequency × (serialization width ÷ 2)	frequency × (serialization width ÷ 2)
4	Half	fastclk phase = 90°	fastclk phase = 90° fastclk as PLL core
3	Half	feedback clock signal	feedback clock signal
2	Half		

Table 3: Trion LVDS RX Deserialization Configuration Summary

• slowclk—Parallel clock for LVDS RX data lanes

• fastclk—Serial clock for LVDS RX data lanes

Titanium/Topaz FPGAs

i

Titanium/TopazFPGAs support LVDS serialization at both half-rate and full-rate. In half-rate mode, data is sent/received on both the rising edge and falling edge of the serial clock. In full-rate mode, data is sent/received on the rising edge of the serial clock only. Titanium/Topaz FPGAs support full-rate mode for both even and odd serialization widths (even serialization width is 10:1, 8:1, 6:1, 4:1, 2:1; odd serialization width is 7:1, 5:1, 3:1), while half-rate mode for even serialization width only.

Note: Neither full-rate nor half-rate are supported where the serialization width is 9:1.

Edge-Aligned Configuration

TX Edge-Aligned

The following clocks from a single PLL source must be provided to the LVDS interface for edge-aligned signal generation:

- pll_slowclk—Parallel clock LVDS TX blocks, both clock lane and data lanes
- pll_fastclk—Serial clock for LVDS TX blocks, both clock lane and data lanes

The frequency, phase, and pll_slowclk and pll_fastclk settings are as follows:

- Data rate must be less than or equal to the maximum toggle rate as stated in the data sheet, depending upon the device speed grade and choice of half or full rate
- pll slowclk frequency = data rate / serialization width
- pll slowclk phase = 0°
- pll_fastclk frequency:
 - Half-rate:

```
pll_fastclk frequency = pll_slowclk frequency × (serialization width ÷ 2)
```

— Full-rate:

pll fastclk frequency = pll slowclk frequency × serialization width

- pll_fastclk phase:
 - Half-rate: 90°
 - Full-rate: 135°
- pll_slowclk as core feedback clock signal for the PLL

Figure 11: Titanium/Topaz TX, Half-Rate, Edge-Aligned Waveform



Figure 12: Titanium/Topaz TX, Full-Rate, Edge-Aligned Waveform



Figure 14: Titanium/Topaz RX, Full-Rate, Edge-Aligned Waveform

Deserialization: 1-to-8

pll fastclk 90.0°

clk_lane_RXP				
data_lane_RXP	(a[0](a[1](a[2])(a[3](a[4])(a[5])(a[6])(a[7])	b[0]	(b[4] b[5] b[6] b[7]	
pll_slowclk 0.0° (pll feedback) pll_fastclk 180.0°				
core_data IN[7:0]	()(a	[7:0]	<mark>b[7:</mark> 0]

Center-Alignment Configuration

While the HSIO of Titanium/Topaz FPGAs supports much higher data rates than Trion FPGAs, traditional 45° and 135° methods may not be sufficiently reliable for large-scale production due to minor variations in skew. To address this issue, we have developed three robust configuration methods for serializing center-aligned data, ensuring consistent performance and reliability across all Titanium/Topaz FPGAs. These methods have been thoroughly tested to guarantee the highest quality and reliability, providing you with confidence in the performance of Titanium/Topaz FPGAs.

TX Center-Aligned with Three Clocks

In this configuration, the LVDS TX block used as a **clock lane uses half-rate serialization** and a serialization width of 4 for the serializer clock pattern. The pll_slowclk is shared for parallel clocks on both clock and data lanes. This configuration can save a global clock buffer resource. This configuration method only supports half-rate serialization of 10, 8, and 6; or full-rate serialization of 10, 8, and 6.

A serialization width of four for the LVDS TX block for the clock lane is required.

The following clocks from a single PLL need to be provided to the LVDS TX interface for center-aligned signal generation:

- pll slowclk—Parallel clock for LVDS TX blocks, both clock lane and data lanes
- pll fastclk data—Serial clock for LVDS TX blocks, data lanes
- pll_fastclk_clk—Serial clock for LVDS TX block, clock lane

The requirement of frequency, phase, and settings for pll_slowclk, pll fastclk data, and pll fastclk clk are given as follows:

- Data rate must be less than or equal to the maximum toggle rate (follows device datasheet, depending on device's speed grade and half-/full-rate choice)
- pll_slowclk frequency = data rate / serialization width
- pll_slowclk phase = 0°
- pll_fastclk_data frequency:
 - Half-rate:

pll_fastclk_data frequency = pll_slowclk frequency× (serialization
width ÷ 2)

— Full-rate:

 $\verb"pll_fastclk_data\ frequency = \verb"pll_slowclk" frequency \times serialization width$

- pll fastclk data phase:
 - Half-rate: 90°
 - Full-rate: 135°
- pll_fastclk_clk frequency = pll_slowclk frequency × 2

(as clock lane use half-rate with serialization width of four, pll_slowclk is shared together with data lanes as parallel clock)

- pll fastclk clk phase =
 - Half-rate: $(2 \times 360) \div$ serialization width
 - Full-rate: $(7 \times 360) \div (4 \times \text{serialization width})$
 - pll slowclk as core feedback clock signal for the PLL

Note: If the PLL clock calculator cannot find a solution for the clockout frequencies and required phase combinations, you can add an extra clockout with a frequency set to an integer of multiple PLL reference clock frequencies, and with phase 0° as the PLL core feedback clock. This can provide more clock frequency output options. Otherwise, you must choose an oscillator with an oscillating frequency appropriate to your target data rate requirements.

Example:

target PLL clkout settings:

(*PLL clock calculator cannot find this solution)

- refclk: 50 MHz
- clkout0: 187.5 MHz, 0°, (feedback)
- clkout1: 750 MHz, 90°

workaround PLL clkout settings:

- refclk: 50 MHz
- clkout0: 187.5 MHz, 0°
- clkout1: 750 MHz, 90°
- clkout2: 50 MHz, 0°, (feedback) [50 MHz = 50×1]

Figure 15: Titanium/Topaz TX, Half-Rate, Center-Aligned Waveform (Three Clock Configuration)



Figure 16: Titanium/Topaz TX, Full-Rate, Center-Aligned Waveform (Three Clock Configuration)



core_data_OUT[7:0] is user data from core logic, while peri_clock_OUT[3:0] is a constant pattern provided by the interface when the LVDS TX block is set to reference clock output mode.

Example:

Scenario:

- Titanium Ti60F225
- Speed grade: C4
- I/O standard: LVDS
- Serialization width: 8-to-1, half-rate
- Maximum toggle rate = 1500 Mbps
- Target data rate = 1500 Mbps

Frequencies:

- pll slowclk frequency = $1500 \div 8 = 187.5 \text{ MHz}$
- pll_fastclk_data frequency = $187.5 \times (8 \div 2) = 750 \text{ MHz}$
- pll_fastclk_clk frequency = $187.5 \times 2 = 375 \text{ MHz}$

Phases:

- pll_slowclk phase = 0°
- pll_fastclk_data phase = 90°
- pll fastclk clk phase = $(2 \times 360) \div 8 = 720 \div 8 = 90^{\circ}$

Figure 17: Titanium TX Center-Aligned PLL Settings Example (Three Clock Configuration)

Serialization Width: 8-to-1; Data Rate: 1500 Mbps



TX Center-Aligned with Four Clocks

In this configuration, the PLL generates two clocks for the data lanes and another two clocks for the clock lane. The pair of clocks will serialize the clock pattern and user data such that the serial clock output edge at the center of the data output.

The following clocks from a single PLL need to be provided to the LVDS TX interface for center-aligned signal generation:

- pll slowclk data—Parallel clock for LVDS TX blocks, data lanes
- pll fastclk data-Serial clock for LVDS TX blocks, data lanes
- pll slowclk clk—Parallel clock for LVDS TX block, clock lane
- pll fastclk clk—Serial clock for LVDS TX block, clock lane

The frequency, phase, and settings for pll_slowclk_data, pll_fastclk_data, pll_slowclk_clk, and pll_fastclk_clk are as follows:

- data rate must be less than or equal to the maximum toggle rate (follows device datasheet, depending on device's speed grade and half-/full-rate choice)
- pll_slowclk_data frequency = data rate ÷ serialization width
- pll slowclk data phase = 0°
- pll fastclk data frequency:
- Half-rate:

```
pll_fastclk_data frequency = pll_slowclk_data frequency ×
(serialization width ÷ 2)
```

— Full-rate:

```
pll_fastclk_data frequency = pll_slowclk_data frequency×
serialization width
```

- pll fastclk data phase:
 - Half-rate: 90°
 - Full-rate: 135°
- pll_slowclk_clk frequency = pll_slowclk_data frequency
- pll slowclk clk phase = $360^\circ \div (2 \times \text{serialization width})$
- pll fastclk clk frequency = pll fastclk data frequency
- pll_fastclk_clk phase:
 - Half-rate: 180°
 - Full-rate: 270°
- pll slowclk as core feedback clock signal for the PLL

Note: If the PLL clock calculator cannot find a solution for the clockout frequencies and required phase combinations, you can add an extra clockout with a frequency set to an integer of multiple PLL reference clock frequencies, and with phase 0° as the PLL core feedback clock. This can provide more clock frequency output options. Otherwise, you must choose an oscillator with an oscillating frequency appropriate to your target data rate requirements.

Figure 18: Titanium/Topaz TX, Half-Rate, Center-Aligned Waveform (Four Clock Configuration)



core_data_OUT[7:0] is user data from core logic, while peri_clock_OUT[3:0] is a constant pattern provided by the interface when the LVDS TX block is set to reference clock output mode.

Example:

Scenario:

- Titanium Ti60F225
- Speed grade: C4
- I/O standard: LVDS
- Serialization width: 8-to-1, half-rate
- Maximum toggle rate = 1500 Mbps
- Target data rate = 1500 Mbps

Frequencies:

- pll slowclk data frequency = $1500 \div 8 = 187.5 \text{ MHz}$
- pll fastclk data frequency = $187.5 \times (8 \div 2) = 750 \text{ MHz}$
- pll slow clk frequency = pll_slowclk_data frequency = 187.5 MHz
- pll fastclk clk frequency = pll fast clk data frequency = 750 MHz

Phases:

- pll_slowclk_data phase = 0°
- pll fastclk data phase = 90°
- pll slowclk clk phase = 360 ÷ (2 × 8) = 360 ÷ 16 = 22.5°
- pll fastclk clk phase = 180°

Figure 20: Titanium TX Center-Aligned PLL Settings Example (Four Clock Configuration)

Serialization Width: 8-to-1; Data Rate: 1500 Mbps



In the figure above, CLKOUT4 acts as the PLL's feedback clock signal from core. Depending on the device (e.g., Ti60), CLKOUT4 may be limited to driving only the top and bottom peripheral blocks. To choose CLKOUT4 as a feedback clock signal, select the reference clock pin from the Top/Bottom, for example in Ti60F225, PLL_TL0, pick GPIOT_P_00 from External Clock 1 as the reference clock source instead of using GPIOL_P_18 from External Clock 0.

TX Center-Aligned with Three Clocks and Soft-Core Clock Pattern Serializer

In this configuration, the soft-core clock pattern serializer in user RTL generates the clock pattern for clock lane LVDS TX block with serialization width of two. Configure the mode of the clock lane LVDS TX block as **serial data output** in the interface designer. LVDS TX with a serialization width of two uses both the rising and falling edges of the parallel clock for serialization, just like in the double-data-rate method; therefore, only one clock signal is required when using a serialization width of two. The soft-core clock pattern serializer registers a looping data pattern for clock pattern generation; therefore, around 20 XLR will be needed for a clock lane.

The following clocks from a single PLL need to be provided to the LVDS TX interface for center-aligned signal generation:

- pll slowclk data—Parallel clock for LVDS TX blocks, data lanes
- pll fastclk data—Serial clock for LVDS TX blocks, data lanes
- pll slowclk clk—Parallel clock for LVDS TX block, clock lane

The frequency, phase, and settings for pll_slowclk_data, pll_fastclk_data, pll_slowclk_clk, and pll_fastclk_clk are as follows:

- Data rate must be less than or equal to the maximum toggle rate (follows device datasheet, depending on device's speed grade and half-/full-rate choice)
- pll_slowclk_data frequency = data rate ÷ serialization width
- pll_slowclk_data phase = 0°
- pll_fastclk_data frequency:
 - Half-rate: pll_fastclk_data frequency = pll_slowclk_data frequency × (serialization width ÷ 2)
 - Full-rate: pll_fastclk_data frequency = pll_slowclk_data frequency× serialization width
- pll_fastclk_data phase:
 - Half-rate: 90°
 - Full-rate: 135°
- pll_slowclk_clk frequency:
 - Half-rate:
 - pll_slowclk_clk frequency = pll_fastclk_data frequency
 - Full-rate:

pll slowclk clk frequency= pll fastclk data frequency ÷ 2

- pll slowclk clk phase = 157.5°
- pll_locked signal as active low asynchronous reset signal for the soft-core pattern serializer
- pll_slowclk_data as core feedback clock signal for the PLL

Note: If the PLL clock calculator cannot find a solution for the clockout frequencies and required phase combinations, you can add an extra clockout with a frequency set to an integer of multiple PLL reference clock frequencies, and with phase 0° as the PLL core feedback clock. This can provide more clock frequency output options. Otherwise, you must choose an oscillator with an oscillating frequency appropriate to your target data rate requirements.





Figure 22: Titanium/Topaz TX, Full-Rate, Center-Aligned Waveform (Three Clock Configuration with Soft-Core Clock Pattern Serialization)



core_data_OUT[7:0] is user data from core logic, while core_clk_OUT[7:0] is a repeating pattern provided by the soft-core clock pattern serializer to generate a clock pattern on the LVDS TX clock lane configured with serial_data_output mode, a serialization width of two, and the Enable Half Rate Serialization option checked.

Soft-Core Clock Pattern Serializer

Verilog RTL of soft-core clock pattern serializer for serialization width eight and its usage are attached below to serialize center-aligned clock pattern using LVDS TX with serialization width two (mode as serial data output). The initial pattern of register r_ring 8'b00011110 makes the bit [0] of serial data center-aligned on positive edge of clock lane. You may change the initial pattern of register r_ring to make the other bit of serial data to instead of bit [0] center-aligned on positive edge of clock lane depends on the application.

Usage:

```
clk_pattern_serializer clk_pattern_serializer_u0 (
    .i_sclk_clk (tx_slowclk_clk),
    .i_pll_locked (i_tx_pll_locked),
    .o_clk_pattern (tx_ch0_clk_TX_DATA)
    );
```

RX Center-Aligned

The connection type for the LVDS RX clock lane must be set to pll_clkin.

The following clocks from a single PLL need to be provided to the LVDS TX interface to receive center-aligned signals:

- pll slowclk—Parallel clock for LVDS RX blocks, data lanes
- pll_fastclk—Serial clock for LVDS RX blocks, data lanes

The requirement of frequency, phase, and settings for pll_slowclk, pll_fastclk in the PLL setup are given as follows:

- Data rate = RX clock lane refclk frequency × deserialization width
- Data rate must be less than or equal to the maximum toggle rate (follows device datasheet, depending on device's speed grade and half-/full-rate choice)
- pll_slowclk frequency = RX_clock_lane_refclk frequency
- pll slowclk phase = 0°
- pll fastclk frequency:
 - Half-rate:

pll_fastclk frequency = pll_slowclk frequency × (deserialization width ÷
2)

— Full-rate:

pll_fastclk frequency = pll_slowclk frequency × deserialization width

- pll_fastclk_phase
- Half-rate: 90°
- Full-rate: 180°
- pll_fastclk as core feedback clock signal for the PLL
- If the above PLL configuration cannot be obtained for full-rate, an extra clockout as the core feedback clock is required. The feedback clock must be in phase with

pll_fastclk and its frequency is an integer multiple of RX_clock_lane_refclk frequency.

Serialization: 8-to-1					
clk_lane_RXP					
data_lane_RXP	a[0] a[1] a[2] a[3	3] a[4] a[5] a[6]	a[7] <mark>b[0]</mark> b[1]	b[2] b[3] b[4] b[5]	b[6] b[7]
pll_slowclk 0°					
pll_slowclk 90° (pll feedback)					
core_data_IN[7:0])			a[7:0]	<mark>b[7:</mark> 0]
Figure 24: Titaniu	ım/Topaz RX, Ful	l-Rate, Center-	Aligned Wa	aveform	
Serialization: 8-to-1					
clk_lane_RXP					
data_lane_RXP	a[0] a[1] a[2] a[3] a[4] a[5] a[6]	a[7] <mark>b[0]</mark> b[1]) b[2] b[3] b[4] b[5]	b[6] b[7]
pll_slowclk 0°					
pll_slowclk 180° (pll feedback)					
core_data_IN[7:0]				a[7:0]	[<mark>b[7:</mark> 0]

Figure 23: Titanium/Topaz RX, Half-Rate, Center-Aligned Waveform

Titanium/Topaz 7:1 Serialization Examples

Unlike Trion FPGAs that serialize/deserialize using odd width data at half-rate, Titanium/Topaz FPGAs serialize/deserialize odd width data only at full-rate. The clocks and PLL must be set up depending on the data alignment as described previously. For centeralign configuration, Titanium/Topaz devices only support four clock configuration and three clock configuration with a soft-core clock pattern serializer.





Summary

Serialization	Half/	TX Configuration				
Width	Full Rate			Center-Align		
		Edge-Align	Three Clocks	Four Clocks	Three Clocks with Soft-Core Pattern Serializer	
10	Half	slowclk_freq	slowclk frequency	slowclk_data	slowclk_data	
8	Half	= data rate / serialization width	= data rate ÷ serialization width	trequency = data rate ÷	trequency = data rate ÷	
6	Half	slowclk phase = 0° fastclk frequency =	slowclk phase = 0° fastclk_data	serialization width slowclk_data phase = 0°	serialization width slowclk_data phase = 0°	
		slowclk frequency × (serialization width ÷ 2) fastclk phase = 90°	frequency = slowclk frequency × (serialization width ÷ 2) fastclk_data phase = 90°	fastclk_data frequency = slowclk_data frequency × (serialization width ÷ 2)	fastclk_data frequency = slowclk_data frequency × (serialization width ÷ 2)	
			(Clock Lane uses half-rate, serialization	fastclk_data phase = 90°	fastclk_data phase = 90°	
			width: 4) fastclk_clk frequency = slowclk frequency × 2 fastclk_clk phase = (2 × 360) ÷ serialization width	slowclk_clk frequency = slowclk_data frequency slowclk_clk phase = 360 ÷ (2 × serialization width) fastclk_clk	(Clock Lane uses half-rate, serialization width: 2) slowclk_clk frequency = slowclk_data frequency × 2 slowclk_clk phase = 180°	
4	Half		Not Supported	frequency = fastclk_data frequency fastclk_clk phase = 180°		

Table 4: Titanium/Topaz LVDS_TX Serialization Configuration Summary

Serialization	Half/	TX Configuration			
Width	Full Rate			Center-Align	
		Edge-Align	Three Clocks	Four Clocks	Three Clocks with Soft-Core Pattern Serializer
<u>10</u> 8	Full	slowclk frequency = data rate ÷ serialization width slowclk phase = 0° fastclk frequency = slowclk frequency × serialization width fastclk phase = 135°	slowclk frequency = data rate ÷ serialization slowclk phase = 0° fastclk_data frequency = slowclk frequency × serialization width fastclk_data phase = 135° (Clock Lane uses half-rate, serialization width: 4) fastclk_clk frequency = slowclk frequency × 2 fastclk_clk phase = (7 × 360) ÷ (4 × serialization width)	slowclk_data frequency = data rate ÷ serialization width slowclk_data phase = 0° fastclk_data frequency = slowclk_data frequency × serialization width fastclk_data phase = 135° slowclk_clk frequency = slowclk_data frequency = slowclk_data frequency = slowclk_clk frequency = slowclk_clk phase = 360 ÷ (2 × serialization width) fastclk_clk	slowclk_data frequency = data rate ÷ serialization width slowclk_data phase = 0° fastclk_data frequency = slowclk_data frequency × serialization width fastclk_data phase = 135° (Clock Lane uses half-rate, serialization width: 2) slowclk_clk frequency = slowclk_data frequency × 2 slowclk_clk
7	Full	-	Not Supported	fastclk_data frequency	phase = 157.5°
6	Full		Same as full- rate, serialization width 10 and 8	fastclk_clk phase = 270°	
5	Full]	Not Supported]	
4	Full	-	Not Supported	-	
3	Full	_	Not Supported	-	
2	Half	slowclk frequency = data rate ÷ 2 slowclk phase = 0°	slowclk_	data frequency = data lowclk_data phase = (a rate ÷ 2)°
		(Serialization width two: Serial clock left blank)	slowclk_clk frequency = data rate ÷ 2 slowclk_clk phase = 90° (Serialization width two: Serial clock left blank)		

(i) (i)

Note: All TX configuration uses slowclk/slowclk_data (0°) as PLL core feedback clock signal.

Note: Clock lanes and data lanes use the same serialization width and half-/full-rate configuration if not otherwise specified.

- slowclk: shared parallel clock for LVDS TX clock lane and data lanes
- fastclk: shared serial clock for LVDS TX clock lane and data lanes
- slowclk clk: parallel clock for LVDS TX clock lane
- fastclk clk: serial clock for LVDS TX clock lane
- slowclk data: parallel clock for LVDS TX data lanes
- fastclk data: serial clock for LVDS TX data lanes

Table 5: Titanium/Topaz LVDS_RX Deserialization Configuration Summary

Deserialization	Half/	RX Configuration									
vviatn	Full Rate	Edge-Align	Center-Align								
10	Half	slowclk frequency = rx_refclk frequency	slowclk frequency = rx_refclk frequency								
8	Half	slowclk phase = 0°	slowclk phase = 0°								
6	Half	fastclk frequency = slowclk frequency × deserialization width ÷ 2	fastclk frequency = slowclk frequency × deserialization width ÷ 2								
4	Half	fastclk phase = 90°	fastclk phase = 90°								
		slowclk as PLL core feedback clock signal	fastclk as PLL core feedback clock signal								
10	Full	slowclk frequency = rx_refclk frequency	slowclk frequency = rx_refclk frequency								
8	Full	slowclk phase = 0°	slowclk phase = 0°								
7	Full	fastclk frequency = slowclk frequency × deserialization width ÷ 2	fastclk frequency = slowclk frequency × deserialization width ÷ 2								
6	Full	fastclk phase = 180°	fastclk phase = 180°								
5	Full	slowclk as PLL core feedback clock signal	fastclk as PLL core feedback clock signal								
4	Full	leedback clock signal	leedback clock signal								
3	Full										
2	Half	slowclk frequency = rx_refclk frequency	slowclk frequency = rx_refclk frequency								
		slowclk phase = 90°	slowclk phase = 0°								
		feedback_clk frequency = rx_refclk frequency	slowclk as PLL core feedback clock signal								
		feedback_clk phase = 0°									
		feedback_clk as PLL core feedback clock signal									

- slowclk: parallel clock for LVDS RX data lanes
- fastclk: serial clock for LVDS RX data lanes

i

Note: Deserialization width of two, edge-aligned configuration need a feedback clock just for PLL core feedback clock signal to align pll_slowclk positive edge at middle of data. PLL's clockout that selected as feedback will in-phase with the reference clock.

Delay Calibration

There is an optimal window that is considered the center point of a valid data sampling window. The source synchronous clock and data signals should arrive at the deserializer inputs at the same time; however, the optimal window can vary depending upon package LVDS lane position (i.e., top/bottom/left/right) and PCB trace design.

Keep in mind that some skew will likely be present due to slight variations in production. Therefore, Efinix strongly recommends calibrating the delay settings for LVDS clock and data alignment to ensure best performance.

LVDS Delay Modes

There are three types of delay modes in Efinix FPGAs that can be used for further LVDS clock and data alignment after implementing LVDS TX/LVDS RX channels with the chosen data alignment configuration as discussed in previous chapters.

Delay modes:

- Static delay
 - A static delay setting on interface designer, a bitstream for a delay setting.
- Dynamic delay
 - Delay setting can be changed/tuned with user core logic in user mode.
- Dynamic phase alignment (DPA)
 - A hard block on LVDS RX that can automatically adjust delay settings, only applies to LVDS_RX full-rate serialization mode.

Mode	Tri	on	Тој	oaz	Titanium					
	LVDS TX	LVDS RX	LVDS TX	LVDS RX	LVDS TX	LVDS RX				
Static Delay		~	~	~	~	~				
Dynamic Delay				~		~				
DPA				~		~				

Table 6: Available Delay Modes for Titanium/Topaz/Trion FPGAs

Calibrating with Dynamic Delay (Titanium/Topaz only)

On Titanium and Topaz FPGAs, LVDS RX supports dynamic delay mode. The dynamic delay can be applied to both clock and data lanes by enabling the **Dynamic Mode Delay Settings** in the Efinity Interface Designer. Additional control signals are made available for tuning the delay value in user core logic.

A parallel clock (slowclk) from PLL is required to supply the dynamic delay control block. LVDS Dynamic Delay block is pll_slowclk positive edge triggered. The following requirments must be satisfied to assert the signals for the dynamic delay control block:

- Assert RX_DLY_RST on the positive edge of pll_slowclk
- Assert RX_DLY_INC and RX_DLY_ENA on negative edge of pll_slowclk

Dynamic Delay Control Signals	Signal Assertion	Description
RX_DLY_RST	Positive edge of pll_slowclk	Active HIGH reset. Reset the delay step count to 31.
RX_DLY_ENA	Negative edge of pll_slowclk	Active HIGH delay enable. 0: Same as previous delay step count. 1: Enable to increase/ decrease delay step count.
RX_DLY_INC	Negative edge of pll_slowclk	0: Decrease delay step count every clock cycle. 1: Increase delay step count every clock cycle.

Table 7: LVDS Dynamic Delay Block Assertion Requirements

Each lane offers 64 taps (value 0–63), with each tap adding approximately 20 ps of delay. The reset signal (RX_DLY_RST) resets the delay count to half of the maximum count value – step 31. The updated delay takes effect approximately 5 ns after the rising edge of the parallel clock.

Figure 26: Example Dynamic Delay Control Waveform



Calibrating with Dynamic Phase Alignment (DPA) (Titanium/Topaz only)

The dynamic phase alignment (DPA) is a harden block on the HSIO of Titanium/Topaz FPGAs to automatically eliminates skew for clock-to-data channels and data-to-data channels by adjusting a delay chain setting according to training data pattern toggling transitions.

Adjust the following settings in the Efinity interface designer for the LVDS RX / LVDS BIDIR (RX tab) data lanes to enable the DPA function on LVDS:

- Connection Type Choose normal
- Enable Deserialization Check (DPA not support for bypass mode)
- Enable Half-Rate Deserialization Uncheck (DPA only supports full-rate deserialization)
- Delay Mode Choose DPA option

Figure 27: DPA Block Diagram



Table	8:	DPA	Signal	Des	crip	tion
-------	----	-----	--------	-----	------	------

Signal	Direction (from RTL)	Description
DLY_RST	Output	Reset DOA circuit.
DLY_ENA	Output	Enable DPA operation.
DLY_LOCK	Input	1: Indicates DPA process has achieved training lock and data can be passed.
		0: DPA training operation is ongoing.
DLY_DBG [5:0]	Input	Delay step of DPA circuit currently set at <i>n</i> .

DPA Training Pattern Requirements

For training patterns sent from TX, from LSB to MSB-1 (denoted as n here), bit[n] and bit[n+1] cannot be equal more than six slowclk cycles. Any training pattern, including a PBRS pattern, can be used as training pattern as long as it satisfies the requirements.

For example:

```
Deserialization width of 8, the training pattern is static 8'h55 = 8'b01010101.
bit[0] and bit[1] are not equal.
bit[1] and bit[2] are not equal.
bit[2] and bit[3] are not equal.
bit[3] and bit[4] are not equal.
bit[4] and bit[5] are not equal.
bit[6] and bit[6] are not equal.
bit[6] and bit[7] are not equal.
Same for every following slowclk cycle.
No two consecutive bits (bit[n] and bit[n+1]) are equal for more than 6 slowclk cycles,
satisfying the training pattern requirement.
```

Note: If the training pattern requirements are not satisfied, DLY_DBG [5:0] always appear as 16 because the DPA operation cannot be completed. DLY_LOCK can still go HIGH as timeout signal for DPA operation.

Figure 28: DPA Example Waveform 1

i



Figure 29: DPA Example Waveform 2

Consider DPA lock when delay step reaches 0 or 63.



To enable the DPA circuit in HSIO, select **Delay Mode** > **DPA** under **Advanced Settings** in the Interface Designer under LVDS RX Block Editor.

The following steps are needed for DPA training:

- 1. Prepare the toggling changing signal from the transmitter side on the data lanes for the delay training. Refer to the previously described training pattern requirements above.
- 2. Before starting the training (before or during user mode), assert PLL reset and DLY_RST signals. Keep the PLL reset signal asserted for at least 10 ns.
- 3. De-assert the PLL reset signal and monitor the PLL locked signal.
- 4. Apply the DPA training pattern to the LVDS_RX data lanes, wait for the DPA circuit to lock. The DPA may take up to 3000 pll slowclk cycles to lock.
- 5. De-assert the DLY_RST signal and assert DLY_ENA after PLL locked signal goes HIGH and the training pattern is ready.
- 6. After the DLY_LOCK signal goes HIGH, de-assert the DLY_ENA signal and assert LVDS RX reset signal for at least one pll fastclk cycle.

A significant clock-data delay mismatch on PCB traces or data rates below 800 Mbps can result in a rolloff or overflow in the value of DLY_DBG, which in turn can cause a jump directly from 0 to 63, or 63 to 0. Such a rolloff will invalidate the DPA operation. Therefore, you should disable the DPA operation when the DPA is stepped on 0 or 63. Be aware that 0/63 here is already the optimum delay step; therefore, you can assume that DPA is already locked at this point, so raise DLY_LOCK_CTRL as this indicator.

7. DPA training process complete. Data can now be sent from the transmitter.

Note: For data rates below 800 Mbps, applying DPA is not significant as a 63 step delay can only cover 1.26 ns (every delay step increase around 20 ps), which is insufficient to cover a 1 unit interval (UI) larger than 1.25 ns. Therefore, disable the DPA by setting DLY_ENA as LOW if the DPA steps on step 0 or step 63 as DPA cannot at the whole UI covered at lower data rate.

DPA Controller Verilog RTL

DPA controller Verilog RTL is provided in the figure below to help control DPA calibration and RX lane reset.

Figure 30: DPA controller Verilog RTL

```
output [LANE_NUM-1:0]
output [LANE_NUM-1:0]
assign o_dly_rst = {LANE_NUM{~i_pll_locked | ~i_arstn}};
assign o_dly_ena = r_ena;
assign o_all_locked = r_all_locked;
assign o_rx_rst = {LANE_NUM{w_rst}};
assign w_rst = ~r_rx_rst[1] & r_rx_rst[0];
                                          r_ena[i] <= 1'b0;
r_lock[i] <= 1'b0;
end else begin
r_ena[i] <= 1'b1;
r_lock[i] <= 1'b1;
r_lock[i] <= r_lock[i];</pre>
                    assign w_all0[i] = ~|(i_dly_dbg[6*(i+1)-1:6*i]);
assign w_all1[i] = &(i_dly_dbg[6*(i+1)-1:6*i]);
```

Usage:

Concatenate ports of DPA from different RX data lanes, shown in the figure below, the controller reset all RX data lanes together at once after all DPA are locked. The o_all_locked signal indicates DPA operation for all data lanes is done.

Figure 31: Data Lane DPA Operation Complete

dpa ctrl #(.LANE NU	M(4)) dpa ctrl u0 (
	(rx_slowclk_ch0),			
.i_arstn	(i_arstn),			
.i_pll_locked	(i_rx_pll_locked),			
.i_dly_dbg	({rx_ch0_d3_RX_DBG,	rx_ch0_d2_RX_DBG,	rx_ch0_d1_RX_DBG,	<pre>rx_ch0_d0_RX_DBG }),</pre>
.i_dly_lock	({rx_ch0_d3_RX_LOCK,	rx_ch0_d2_RX_LOCK,	rx_ch0_d1_RX_LOCK,	<pre>rx_ch0_d0_RX_LOCK }),</pre>
.o_dly_rst	({rx_ch0_d3_RX_DLY_RST,	rx_ch0_d2_RX_DLY_RST,	rx_ch0_d1_RX_DLY_RST,	<pre>rx_ch0_d0_RX_DLY_RST}),</pre>
.o_dly_ena	({rx_ch0_d3_RX_DLY_ENA,	rx_ch0_d2_RX_DLY_ENA,	rx_ch0_d1_RX_DLY_ENA,	<pre>rx_ch0_d0_RX_DLY_ENA}),</pre>
.o_rx_rst	({rx_ch0_d3_RX_RST,	rx_ch0_d2_RX_RST,	rx_ch0_d1_RX_RST,	<pre>rx_ch0_d0_RX_RST }),</pre>
.o_all_locked	(w_all_dpa_locked));			

Calibrating with Static Delay (Trion/Titanium/Topaz)

To calibrate a static delay process, a known synchronous data pattern is sent via a transmitter to an FPGA, which acts as a receiver. The data pattern consists of a repeating string; for example, deserialization width of 8, cycling 8'h4b, 8'h57, 8'h7c, 8'h3e. The FPGA is loaded with an RTL calibration design that does the word alignment and verifies receipt of the data pattern. Once received data pattern after word aligning is mismatch with the known data pattern, a FAIL signal asserted.

Static Delay Calibration Steps:

- 1. With the calibration design project peripheral design opened in Interface designer, set **Static Mode Delay** setting for clock lane and all data lanes to 0. (If Trion FPGA is used, also turn on the **Enable Delay** option).
- 2. Save the peripheral design, generate Efinity Constraint Files, and then re-generate bitstream. Synthesis, Place and Route compilation steps can be skipped as no changes are made in RTL design files.
- 3. Program FPGA with newly generated bitstream.
- 4. Send out the known synchronous data pattern from transmitter, check any received data pattern mismatch with the known data pattern, record down the checker PASS/FAIL result for all data lanes.
- 5. Repeat step 2 to 4, by sweeping the Static Mode Delay setting from 0 to 63 for all data lanes.
- 6. If two passing window edges are found, select the middle value in the passing window range. For example, with passing window following, the valid range is from 9 to 47; therefore, select the middle value (i.e., 28) for all data lanes.

Formula: middle value = (upper edge - lower edge) \div 2 + lower edge.

Figure 32: Static Delay Calibration - Passing Window 1

Clock Delay							0
Data Delay	0	1 3	23	4	56	78	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
Data Lane 0							
Data Lane 1							
Data Lane 2							
Data Lane 3							

 If valid windows are found at both ends, the passing window is assumed to be a loop. Select the middle value. For example, referring to the passing window following, there are passing ranges on both ends, lower edge = 13, upper edge = 50, select middle value = 0 for all data lanes.

```
Formula: middle value = lower edge - [(64 - upper edge) + lower edge] \div 2.
```

```
Figure 33: Static Delay Calibration - Passing Window 2
```

Clock Delay																			0															
Data Delay	0	1	23	4	56	78	9 10	11 12	2 13 14	4 15 16	17 18	19 20	21 2	2 23 2	4 25	26 27	28 29	9 30 3 [,]	1 32 3	33 34	35 36	37 38	39 40	41 4	2 43	44 45	46 47	48 49	9 50 51	1 52 53	54 55 56	57 58	59 60 61	1 62 63
Data Lane 0																																		
Data Lane 1																																		
Data Lane 2																																		
Data Lane 3																																		

 (\mathbf{i})

Note: Bit slip is a condition in which a bit falls into another word (either MSB slipped as LSB of the next word, or LSB slipped as MSB of the previous word, or multiple bits have slipped to another word). Bit slip happens when a clock-data delay mismatch occurs due to PCB traces, reflects as another passing window. in the figure above, TX sends a counter sequence, (8'h28, 8'h29, 8'h2A...), if the passing window on left (step 0 to 13) receives a counter sequence without any bit slip (8'h28, 8'h29, 8'h2A...); therefore, a passing window on the right (step 50 to 63) will receive a counter sequence with one bit slip (8'h50, 8'h52, 8'h56, ...) as a word:{d0[6:0], d1[7]}.

1. If the data rate is 700 Mbps or lower, change the static delay setting for the clock lane to 63 and the sweeping static delay setting from 0 to 63 again for the data lanes. Using the same method as described in step 6 or 7 (depending upon the situation), use the middle value of the passing window. For example in the case of the passing window shown below, select 50 for all data lanes, and 63 for the clock lane.

Figure 34: Static Delay Calibration - Passing Window 3

i



1. Repeat the above steps for each part until you have a static delay setting that adequately compensates for any manufacturing variation.

Note: The center point of the passing window is dedicated to the PLL and global clock buffer. If any changes to these settings changes the center point of the window, you might need to screen the passing window again to get a new center point.

Revision History

Table	9:	Revision	History	
-------	----	----------	---------	--

Date	Version	Description
June 2025	2.1	Updated Calibrating with Dynamic Delay (Titanium/Topaz only) on page 32. (DOC-2531)
		Fixed typo in Figure 13: Titanium/Topaz RX, Half-Rate, Edge-Aligned Waveform on page 16 and Figure 14: Titanium/Topaz RX, Full-Rate, Edge-Aligned Waveform on page 16. (DOC-2482)
		Fixed wrong image in Figure 8: Trion RX Center-Aligned PLL Settings Example on page 12. (DOC-2518)
March 2025	2.0	Major update of whole document. (DOC-2333)
August 2024	1.5	Corrected half-rate serialization for Titanium FPGAs. (DOC-1937)
March 2024	1.4	Corrected RX_DLY_RST in dynamic delay waveform (Figure 1). (DOC-1697)
February 2024	1.3	Added description about using LDS blocks from the same FPGA side to minimize skew. (DOC-1150) Updated dynamic delay waveform (Figure 1). (DOC-1697)
February 2023	1.2	Updated Feature Summary table. (DOC-1079)
		Updated PLL setting description to mention setting the PLL F_{VCO} above 1.6 GHz. (DOC-1026)
		Updated Calibrating Titanium FPGAs using DPA topic.
October 2022	1.1	Added DPA example waveform.
		Added TX center-aligned method information.
		Added Titanium dynamic and static delay options.
		Added PLL setup example settings.
May 2022	1.0	Initial release.