AN 044: Aligning LVDS Clock and Data

AN044-v1.4
March 2024
www.efinixinc.com

Contents

Introduction

Titanium and Trion FPGAs support LVDS TX and RX 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.

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

Table 1: Titanium and Trion LVDS Feature Summary

Half-rate sends data out on both edges of the `fastclk`, whereas *full-rate* sends data out on the rising edge of the `fastclk`.

Feature	Titanium	Trion
Serialization width	Up to 10:1	Up to 8:1
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	1,500 Mbps (Half-rate) 1,000 Mbps (Full-rate)	800 Mbps

Learn more: Refer to the device data sheet for detailed data rates specifications.

Hardware Design

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

Use the LVDS blocks from the same side of the FPGA to minimize skew between the data lanes, and also between clock and data lanes in an LVDS interface.

Learn more: Refer to the following 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
- 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.

Edge-Alignment

LVDS TX Edge-Aligned

The following clocks are provided to the LVDS interface for signal generation:

- `tx_slowclk`—Parallel clock transmitted together with LVDS data. It is in the same clock domain as input data to the LVDS interface.
- `tx_fastclk`—Serial clock for LVDS serialization.

The `tx_fastclk` and `tx_slowclk` relationship are given by the following equations:

- Half-rate:

  tx_fastclk = tx_slowclk * serialization rate / 2
- Full-rate (Titanium only):

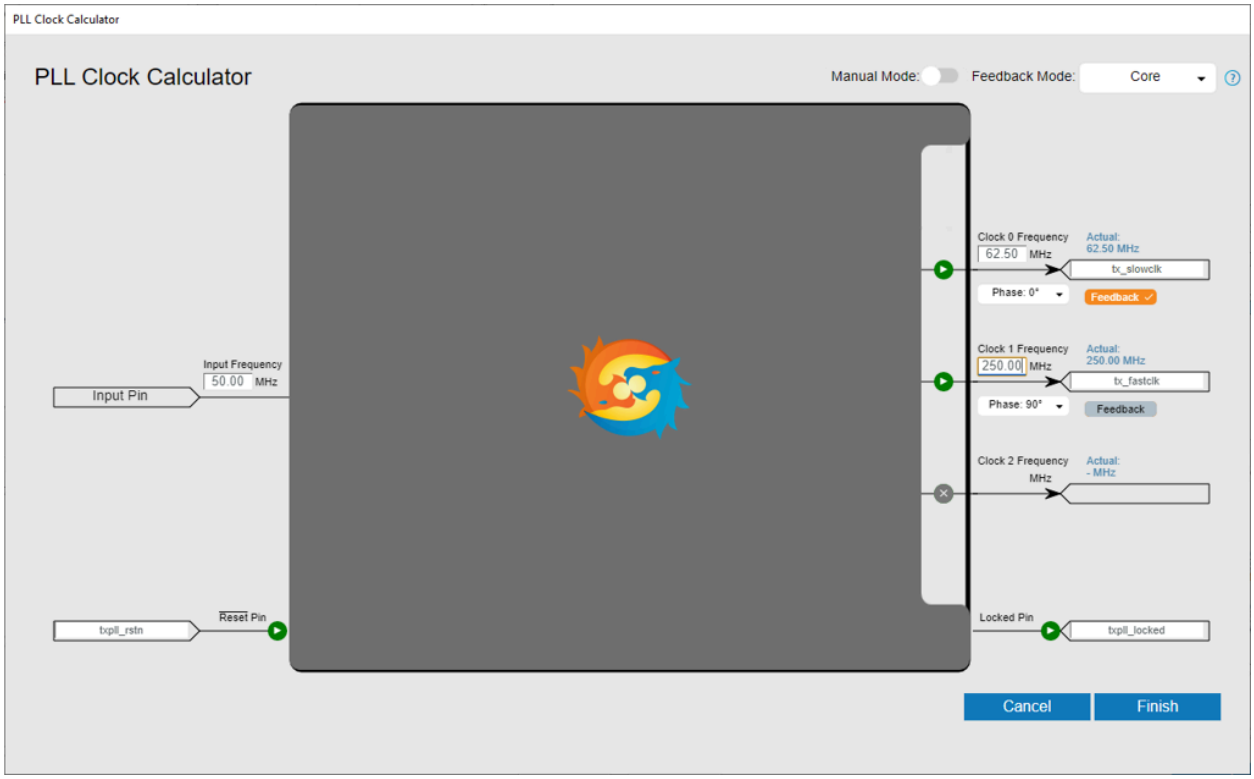
  tx_fastclk = tx_slowclk * serialization rate

The `tx_fastclk` clock edge is aligned to the edge of `dataout` bits. The `tx_slowclk` and `tx_fastclk` **must** be synchronous, and generated from the same PLL. The `tx_fastclk` has a 90° phase shift with respect to `tx_slowclk`. `tx_slowclk` is the feedback clock to the PLL.

The `tx_slowclk` is the feedback clock to PLL. It also functions as the parallel clock for the output clock and data lanes. The `tx_fastclk` has a 90° phase shift. It is the serial clock for the output clock and data lanes.

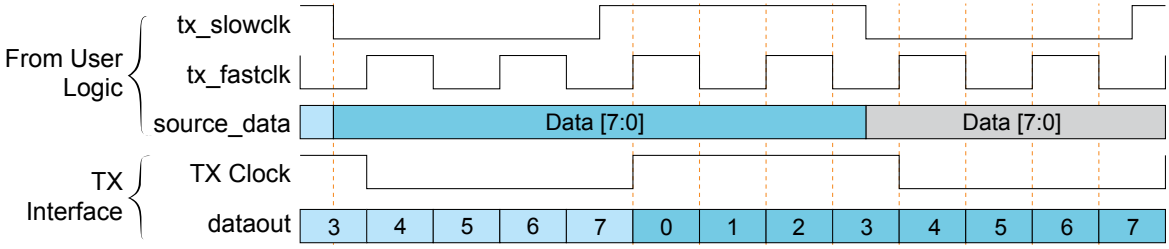
Figure 1: TX Edge-Aligned PLL Settings Example

Settings: Half-rate, 8-to-1 serialization



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 2: TX Edge-Aligned Waveform



LVDS 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 input clock is connected to an LVDS RX block with a `pll_clkin` connection type and generates the following clocks:

- `rx_fastclk`—For LVDS data sampling
- `rx_slowclk`—For synchronous clock on captured data (`data_latch`).

The `rx_fastclk` and `rx_slowclk` relationship are given by the following equations:

- Half-rate:

  rx_fastclk = rx_slowclk * serialization rate / 2
- Full-rate (Titanium only):

  rx_fastclk = rx_slowclk * serialization rate

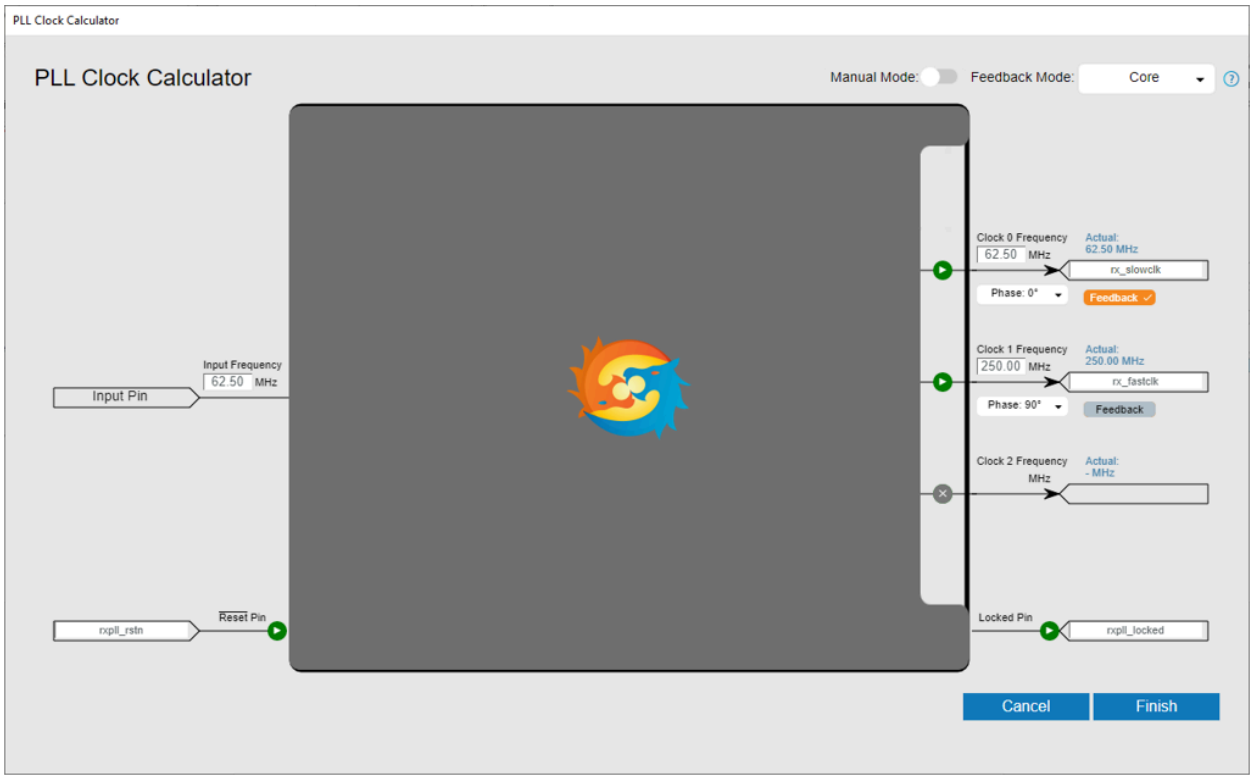
The input data is aligned with the edge of the input LVDS clock (`slowclkin`). The `rx_slowclk` is the feedback clock to PLL. The `rx_fastclk` has a 90° phase shift with respect to `rx_slowclk` to allow sufficient sampling time.

The `data_latch` is aligned with the next rising edge of `rx_slowclk`.

The `rx_slowclk` is the feedback clock to PLL. It also functions as the parallel clock for the input clock and data lanes. The `rx_fastclk` has a 90° phase shift. It is the serial clock for the input clock and data lanes.

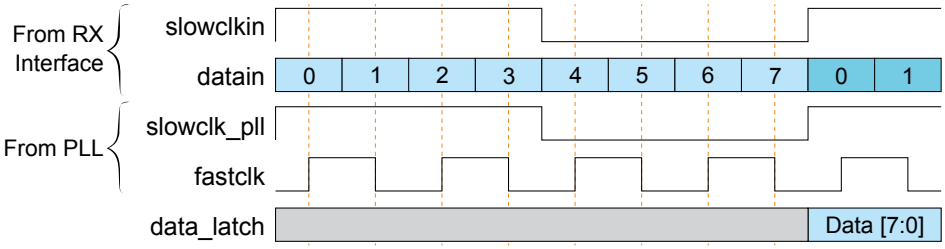
Figure 3: RX Edge-Aligned PLL Settings Example

Settings: Half-rate, 8-to-1 serialization



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 4: RX Edge-Aligned Waveform



Center-Alignment

LVDS TX Center-Aligned

Important: The center-aligned method for LVDS TX is supported in Efinity® software version 2021.2 with 2021.2.323.5.10 patch or later.

The following clocks are provided to the LVDS interface for signal generation:

- `tx_slowclk`—Parallel clock transmitted together with LVDS data. It is in the same clock domain as input data to the LVDS interface.
- `tx_fastclk`—Serial clock for LVDS serialization.

The `tx_fastclk` and `tx_slowclk` relationship are given by the following equations:

- Half-rate:

  tx_fastclk = tx_slowclk * serialization rate / 2
- Full-rate (Titanium only):

  tx_fastclk = tx_slowclk * serialization rate

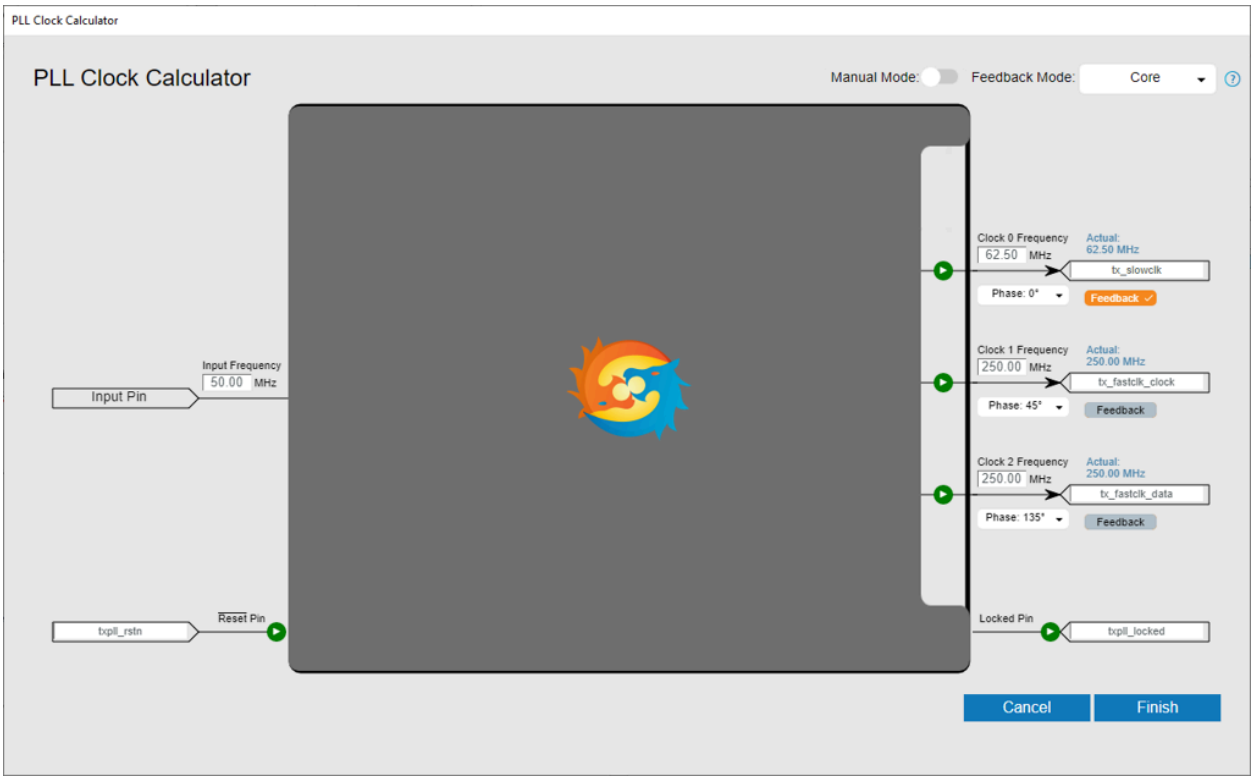
The `tx_fastclk_clock` edge is aligned to the center of `dataout` bits. The `tx_slowclk` and `tx_fastclk` must be synchronous and generated from the same PLL. The clock lane `tx_fastclk_clock` has a 45° phase shift and the data lane `tx_fastclk_data` has a 135° phase shift with respect to the `tx_slowclk`.

You need to generate three clocks in the PLL, one `tx_slowclk` and two `tx_fastclk`s. The `tx_slowclk` is the feedback clock to the PLL It also functions as the parallel clock for the output clock and data lanes. One of the `tx_fastclk` has a 45° phase shift and is the serial clock for the output clock lane, `tx_fastclk_clock`. The other `tx_fastclk` has a 135° phase shift and is the serial clock for the output data lane, `tx_fastclk_data`.

Note: In Trion FPGAs, the center-aligned method is not supported when the serialization rate is 3, and the maximum data rate is 600 Mbps.

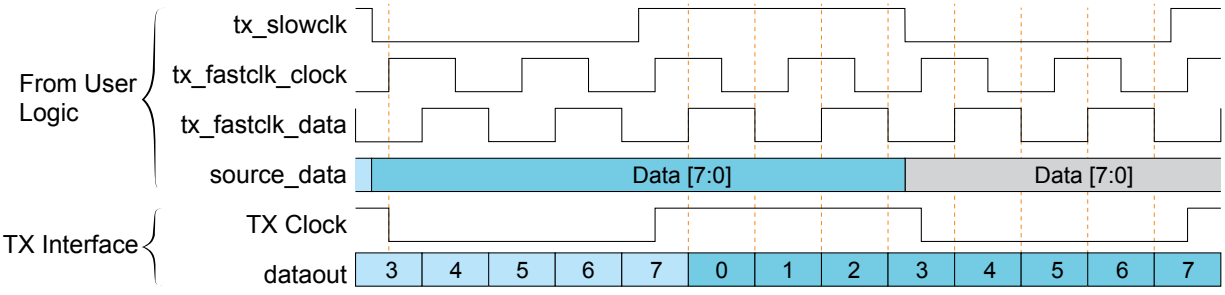
Figure 5: TX Center-Aligned PLL Settings Example

Settings: Half-rate, 8-to-1 serialization



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 6: TX Center-Aligned Waveform



LVDS RX Center-Aligned

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

The input clock is connected to an LVDS RX block with a `pll_clkin` connection type and generates the following clocks:

- `rx_fastclk`—For LVDS data sampling
- `rx_slowclk`—For synchronous clock on captured data (`data_latch`).

The `rx_fastclk` and `rx_slowclk` relationship are given by the following equations:

- Half-rate:
  rx_fastclk = rx_slowclk * serialization rate / 2
- Full-rate (Titanium only):
  rx_fastclk = rx_slowclk * serialization rate

The input data is aligned with the center of LVDS clock input (`slowclkin`). There are two ways to handle the phase relationship between `rx_fastclk` and `rx_slowclk`:

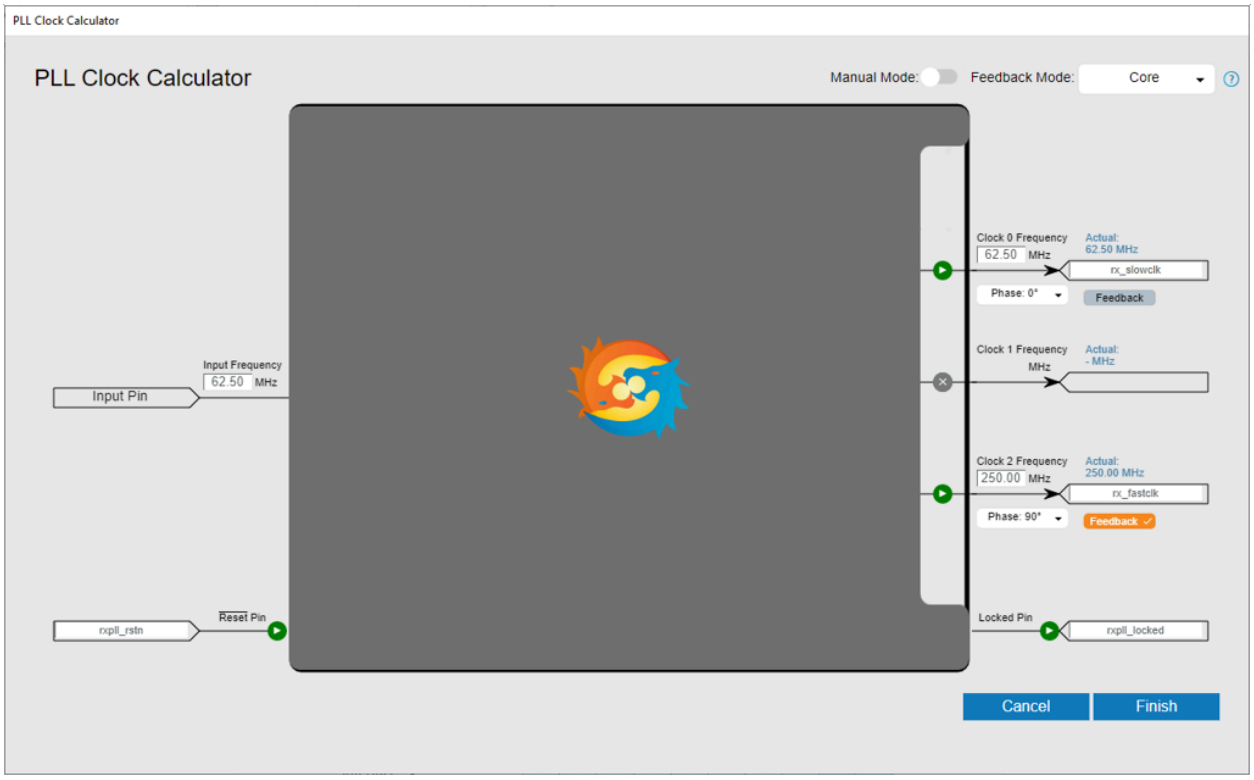
- `rx_fastclk` as PLL feedback
- Using Dummy `slowclk`

`rx_fastclk` as PLL Feedback (Titanium or Trion)

The `rx_fastclk` is the feedback clock to the PLL and has a 90° phase shift with respect to `rx_slowclk`. Byte alignment will be lost when `rx_fastclk` is used for feedback. User is required to realign the byte boundary with their own logic.

Figure 7: RX Center-Aligned PLL Settings Example (PLL feedback)

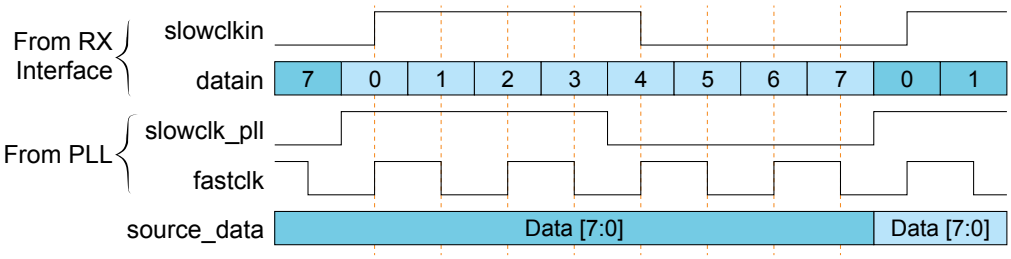
Settings: Half-rate, 8-to-1 serialization



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.

The `data_latch` is aligned with the next rising edge of `slowclk_pll`.

Figure 8: RX Center-Aligned Waveform



Using Dummy `slowclk` (Titanium only)

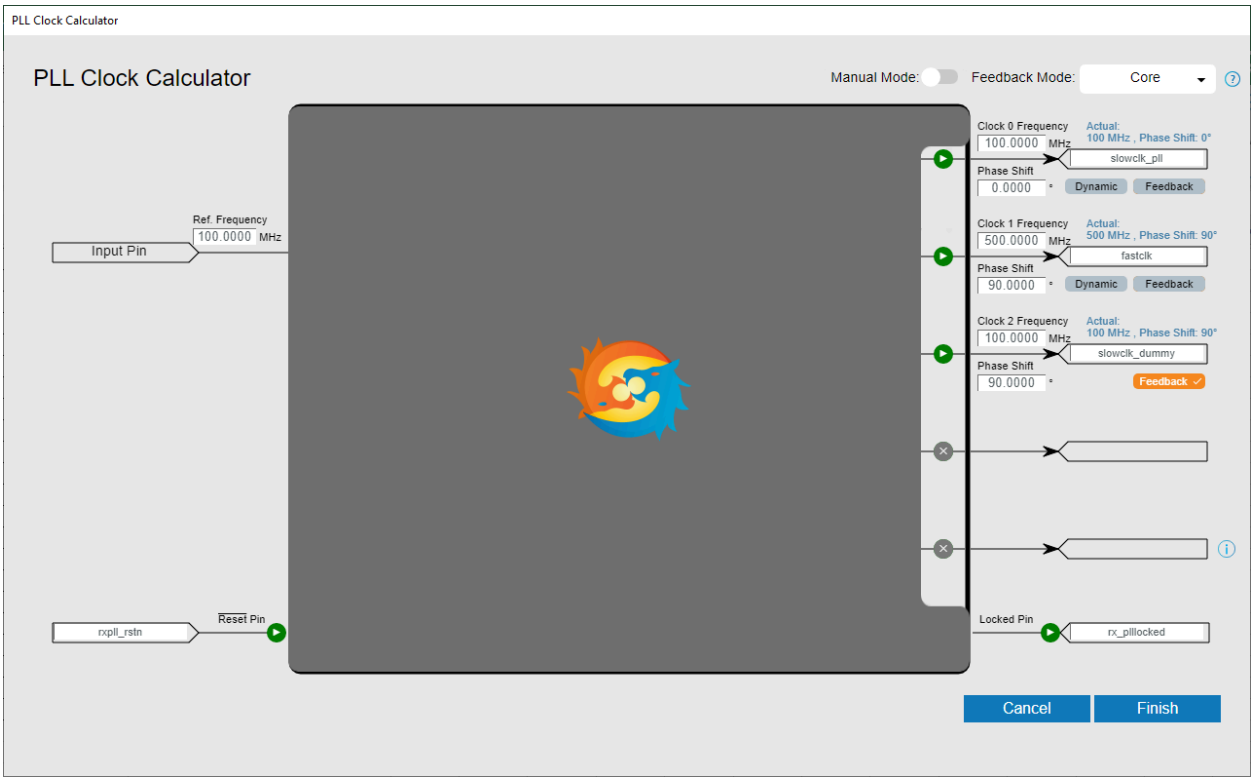
You need to generate the following three clocks in the PLL:

- `slowclk_pll`—Serial clock with 0° phase shift. The `slowclk_pll` is the clock domain for `data_latch` from the LVDS interface.
- `fastclk`—Parallel clock with 90° phase shift.
- `slowclk_dummy`—Serial clock with 90° phase shift and the feedback clock to the PLL.

This method provides more clock frequency output options.

Figure 9: RX Center-Aligned PLL Settings Example (Dummy `slowclk`)

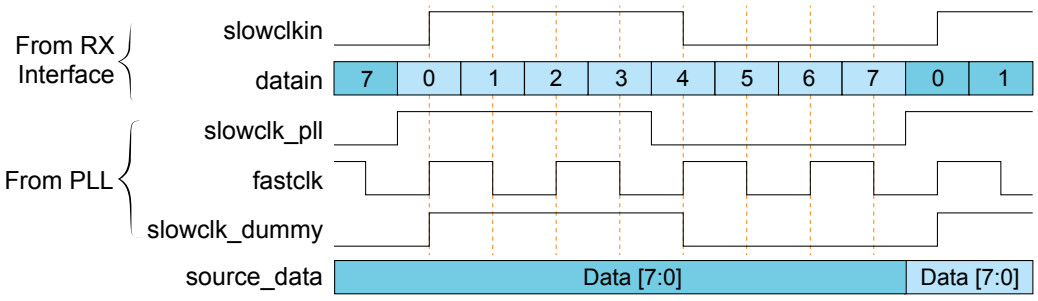
Settings: Half-rate, 8-to-1 serialization



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.

The `data_latch` is aligned with the next rising edge of `slowclk_pll`.

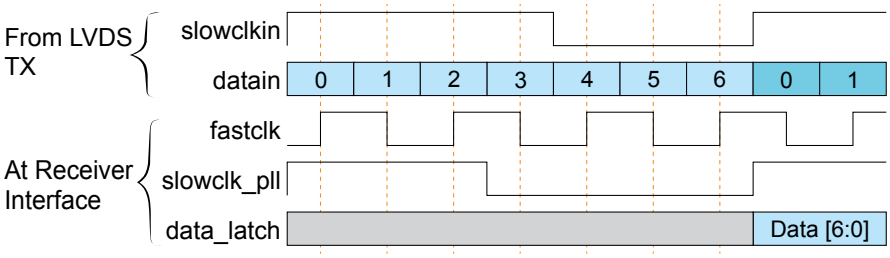
Figure 10: RX Center-Aligned Waveform (Dummy `slowclk`)



7:1 Serialization Examples

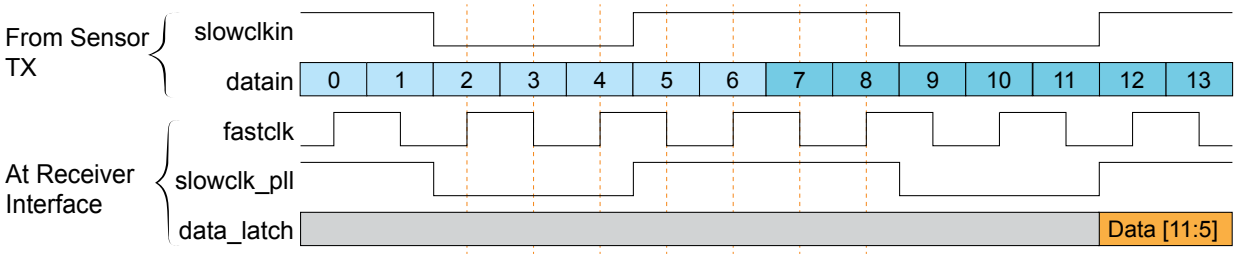
When receiving from Trion and Titanium LVDS TX, the input clock signal is `1110000` for 1 data set. You have to set up the clock and PLL depending on the data alignment as described previously.

Figure 11: Data Input from Trion and Titanium LVDS TX Example Waveform



Many sensors that are available in the market send out LVDS TX data with clock signal `1100011`, which is different from Trion and Titanium LVDS RX lane's clock alignment. Therefore, each 7-bit word received on the LVDS RX interface is in `{<current_data>[4:0], <previous_data>[6:5]}` format. Additionally, you need an extra stage to align the bits in each word to data [6:0].

Figure 12: Data Input from Common Sensors Example Waveform



Titanium LVDS Delay Modes

The LVDS TX and RX in Titanium FPGAs support several delay modes that you can use for LVDS clock and data alignment. The following table shows the supported delay modes in Titanium FPGAs.

Table 2: Delay Modes for LVDS TX and RX in Titanium FPGAs

Mode	LVDS TX Clock and Data	LVDS RX Clock and Data
Static delay	Yes	Yes
Dynamic delay	No	Yes
DPA	No	Yes

Static Delay

Both LVDS TX and LVDS RX support static delay mode. You can apply static delays to the clock and data lanes. Each lane offers 64 taps, where each tap adds approximately 25 ps of delay. You have to enable the **Static Mode Delay Setting** in Efinity Interface Designer and enter the delay value manually.

Refer to 'Using the GPIO Block' in Titanium Interfaces User Guide for the detailed settings.

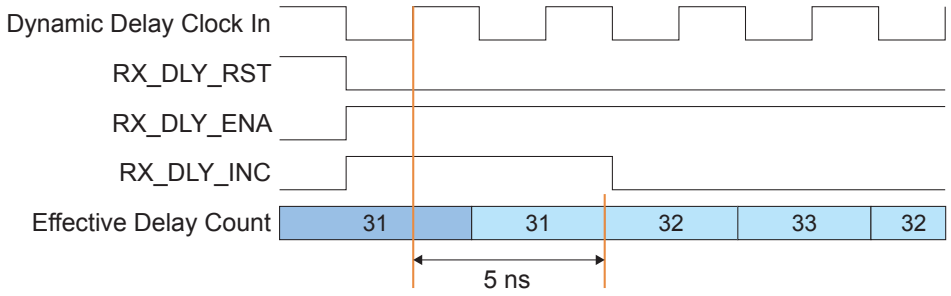
Note: The 25 ps delay per tap is an approximation. You need to measure the delay increment on your board for the accurate delay.

Dynamic Delay

Only the LVDS RX supports the dynamic delay mode. You can apply dynamic delays to clock and data lanes by enabling the **Dynamic Mode Delay Setting** in Efinity Interface Designer. Additional control signals are made available for you to change the delay value.

When you enable the delay control on the device (`RX_DLY_ENA` = 1 and `RX_DLY_RST` = 0), you can increase or decrease the delay count by 1 clock per cycle through the increment port, `RX_DLY_INC`, (1 to increase, 0 to decrease). Each lane offers 64 taps (value 0 to 63), where each tap adds approximately 20 ps of delay. The reset signal (`RX_DLY_RST`) resets the delay count to half of the maximum count value.

Figure 13: Dynamic Delay Example Waveform



DPA

The Dynamic Phase Alignment (DPA) automatically eliminates skew for clock-to-data channels and data-to-data channels by adjusting a delay chain setting so that the data is sampled at the center of the bit period.

To enable the DPA function on LVDS, you need to switch the **Delay Mode** setting to **dynamic** in LVDS Block Editor under Interface Designer. You can control the DPA ports from RTL as timing diagram in next session.

Figure 14: DPA Signal Block Diagram

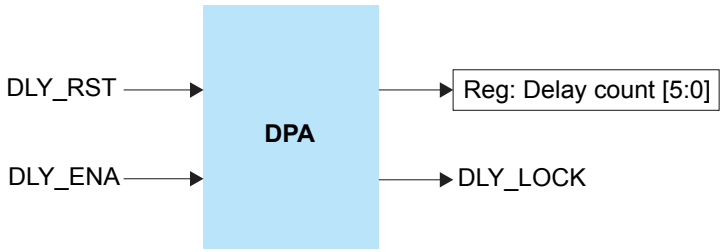


Table 3: DPA Signal Description

Signal	Direction (from RTL)	Description
DLY_RST	Output	Reset DPA circuit
DLY_ENA	Output	Enable DPA circuit
DLY_LOCK	Input	Value '1' indicates that the DPA process has achieved training lock and data can be passed.

Note: Refer to Calibrating Titanium FPGAs using DPA on page 13 for information about using DPA to calibrate LVDS clock and data alignment.

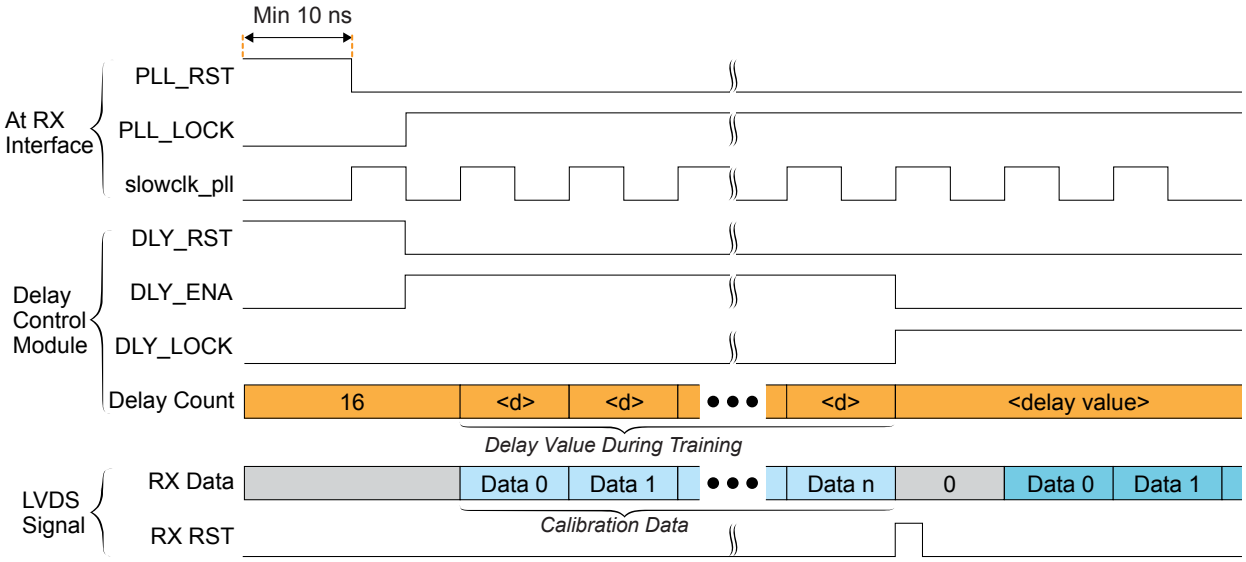
Calibrating LVDS Clock and Data Alignment

Titanium and Trion LVDS RX uses the optimal window which is the center point of a valid data window for capturing input data based on the source synchronous clock and data arriving at the inputs at the same time. However, this optimal window varies with the LVDS lane position and the PCB design. Therefore, Efinix strongly recommends that you calibrate the LVDS clock and data alignment to ensure the best performance.

Calibrating Titanium FPGAs using DPA

The serial clock from PLL (`slowclk_pll`) is the source clock for the DPA module. The DPA feature is only supported in full-rate serialization and the LVDS RX must have the same frequency stability, in parts per million (PPM), as the LVDS TX side.

Figure 15: Dynamic Phase Alignment Example Waveform



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

You need to create a training module for the DPA to get the best input delay setting on LVDS RX. The following steps describe the delay training:

1. Prepare toggling LVDS RX data on data lane for the delay training.
2. Before starting the LVDS RX delay training (before or during user mode), assert the 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. De-assert the `DLY_RST` and assert `DLY_ENA` after PLL locked signal is asserted and stable.
5. Apply the DPA training pattern to LVDS RX and allow the DPA circuit to lock. The DPA may take up to 3,000 `slowclk_pll` cycles to lock.
6. After the `DLY_LOCK` signal is asserted, de-assert the `DLY_ENA` signal and assert LVDS RX reset signal for at least one `fastclk` cycle.

Calibrating Trion FPGAs

Note: To use LVDS delay calibration in Trion FPGAs, you need the Efinity software version 2021.2.323 with patch 2021.2.323.3.14 or later.

You need a project for your board to validate the received data. The validation project is usually a loop-back design where the LVDS TX sends out a synchronous pattern and then verifies the LVDS RX data. The calibration is done for each LVDS RX data lane at a time.

Table 4: Trion Delay Calibration Methods

Calibration Method	Note
Single-pass calibration	Use for LVDS RX with 700 Mbps or higher data rate.
Two-pass calibration	Use for LVDS RX with data rate lower than 700 Mbps.

Single-Pass Calibration

To perform single-pass calibration:

1. Open your validation project in the Efinity software.
2. Open the Interface Designer.
3. Select the LVDS RX data resource in the **Design Explorer** window.
4. In the **Block Editor** window, turn on the **Enable Delay Setting**, and set the **Static Mode Delay Setting** to 0.
5. Save and re-compile the project.
6. Program your board with the validation project.
7. Send out the synchronous pattern TX data, check whether the synchronize pattern is valid, and note the **Static Mode Delay Setting** value.
8. Repeat steps 4 till 7 for **Static Mode Delay Setting** with values from 1 to 63.
9. Repeat the procedures for other LVDS RX data lanes if needed.

The two-pass calibration sweeps 64 possible **Static Mode Delay Setting** values. The **Static Mode Delay Setting** values with a valid synchronize pattern is the optimized receive window range for this lane. The best **Static Mode Delay Setting** value is the middle value of this range. For example, **Static Mode Delay Setting** values from 10 to 20 return a valid synchronize pattern, the best **Static Mode Delay Setting** for this LVDS RX data lane is 15.

Note: The best **Static Mode Delay Setting** value is always in the middle of the optimized window range. With this in mind, you can optimize the calibration effort by targeting certain **Static Mode Delay Setting** values to find the optimized range instead of trying each value one by one.

Two-Pass Calibration

To perform two-pass calibration:

1. Open your validation project in the Efinity, and open the Interface Designer.
2. Select the LVDS RX clock resource in the **Design Explorer** window.
3. In the **Block Editor** window, turn on the **Enable Delay Setting**, and set the **Static Mode Delay Setting** to 0.
4. Select the LVDS RX data resource in the **Design Explorer** window.
5. In the **Block Editor** window, turn on the **Enable Delay Setting**, and set the **Static Mode Delay Setting** to 0.
6. Save the Efinity Interface Designer settings and re-compile the project.
7. Program your board with the validation project.
8. Send out the synchronous pattern TX data, check whether the synchronize pattern is valid, and note the **Static Mode Delay Setting** value.
9. Repeat steps 5 till 8 for LVDS RX data using **Static Mode Delay Setting** with values from 1 to 63.
10. Select the LVDS RX clock resource in the **Design Explorer** window again, and now set the **Static Mode Delay Setting** to **63**.
11. Repeat steps 4 till 9.
12. Repeat the procedures for other LVDS RX data lanes if needed.

The two-pass calibration sweeps 127 possible **Static Mode Delay Setting** values. The **Static Mode Delay Setting** values with a valid sync pattern is the optimized receive window range for this lane. The best **Static Mode Delay Setting** value is the middle value of this range. For example, **Static Mode Delay Setting** values from 30 to 50 return a valid synchronize pattern, therefore the best **Static Mode Delay Setting** for this LVDS RX data lane is 40.

Note: The best **Static Mode Delay Setting** value is always in the middle of the optimized window range. With this in mind, you can optimize the calibration effort by targeting certain **Static Mode Delay Setting** values to find the optimized range instead of trying each value one by one.

Revision History

Table 5: Revision History

Date	Version	Description
March 2024	1.4	Corrected RX_DLY_RST in dynamic delay waveform (**Figure 13: Dynamic Delay Example Waveform** on page 12). (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 13: Dynamic Delay Example Waveform** on page 12). (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.