



# AN 040: Clocking Source-Synchronous Designs

---

**AN040-v1.0**  
**March 2022**  
**[www.efinixinc.com](http://www.efinixinc.com)**

# Contents

<b>Introduction.....</b>	<b>3</b>
<b>Center vs. Edge Aligned.....</b>	<b>3</b>
<b>Understanding Delays.....</b>	<b>4</b>
Clock Insertion Delay.....	5
PLL Compensation Delay.....	5
Core Feedback Mode.....	6
<b>Implementing a Source-Synchronous Receiver.....</b>	<b>7</b>
<b>Implementing a Source-Synchronous Transmitter.....</b>	<b>7</b>
<b>Conclusion.....</b>	<b>8</b>
<b>Revision History.....</b>	<b>8</b>

# Introduction

In a source-synchronous design, the transmitting device sends the clock signal along with the data signal. Usually, source-synchronous designs transmit data on both clock edges, which is called double-data-rate I/O (DDIO). Single-data-rate I/O (SDIO) only transmits data on one clock edge. Systems that incorporate memory devices such as DDR DRAM and HyperRAM typically use source-synchronous clocking for the high-speed interfaces. The receiving device wants to capture data when it is stable, which means you do not want the clock and data signals to change at the same time. Otherwise, you risk reading the data too soon or too late. This application note describes best practices when implementing source synchronous designs in an FPGA.



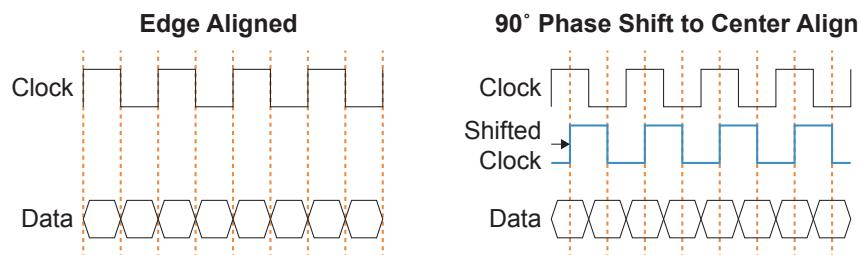
**Note:** This document talks generally about the global clock network. The same concepts apply to the Titanium regional clock network and GPIO configured as RCLK instead of GCLK.

## Center vs. Edge Aligned

In a source-synchronous design, the clock and data are sent together to the receiving device with the clock signal aligned to either the edge or center of the data.

- If you center align the clock, the clock changes when the data is at its most stable point.
- If you edge align the clock, the clock changes at the same time as the data. Therefore, to ensure you are reading stable data, you need to shift your clock signal to the center of the data window. With DDIO designs, the shift value is 90°; with SDIO data the shift value is 180°.

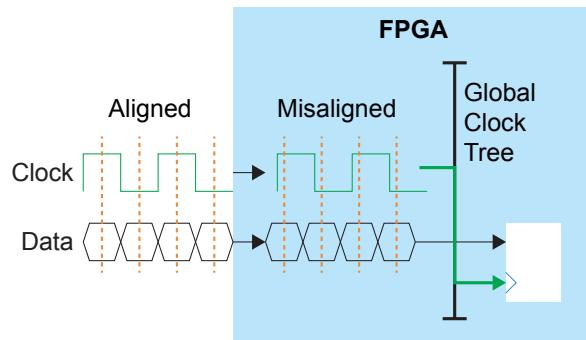
*Figure 1: DDIO Center vs. Edge Aligned Clock and Data*



# Understanding Delays

For source-synchronous design, you are often building a receiver that receives high-speed DDIO data. For the receiver, the clock and data arrive at the FPGA with a known relationship (edge or center aligned). But, routing those data and clock signals to registers in the FPGA disrupts this alignment. Therefore, you need to build your design to account for the delays.

*Figure 2: Clock and Data Signals become Misaligned*



Trion and Titanium I/O registers are designed to have a known, small, consistent delay on the data capture path. You should use them to synchronize the data signals.

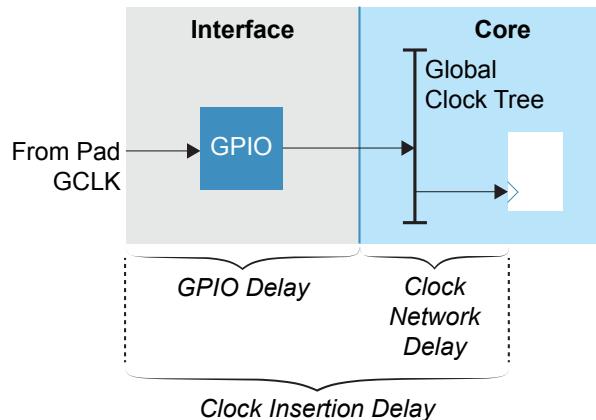


**Learn more:** Refer to the [Trion Interfaces User Guide](#) or [Titanium Interfaces User Guide](#) for information on how to use the GPIO registers.

## Clock Insertion Delay

The clock insertion delay is the time it takes for a clock signal to travel from the pad of a GPIO configured as a global clock (GCLK) through the global clock network to the destination register. This document calls it the "base" clock insertion delay. The [Efinity® Timing Closure User Guide](#) discusses how to handle these delays to close timing.

*Figure 3: GBUF to Register Delay*



Notice that the previous figure does not include a PLL. As soon as you use a PLL to generate clock signals, you need to account for it from a delay perspective. The delay effect caused by the PLL is called the PLL compensation delay.

## PLL Compensation Delay

The PLL compensation delay is the delay that is *removed* from the base clock insertion delay to create the actual insertion delay. It is always an amount you subtract from the clock insertion delay. Some PLL modes have a PLL compensation delay and some do not.

## Core Feedback Mode

Trion and Titanium PLLs support a core feedback mode. In this mode, the only delay is from the GBUF. The clock insertion delay is the GPIO buffer delay from the PLL reference pin to the PLL, which matches the GPIO buffer delay feeding the GPIO input registers. Therefore, the clock and data are aligned at the input register. There is no delay for the global clock network. So overall you have less insertion delay than the local feedback mode.

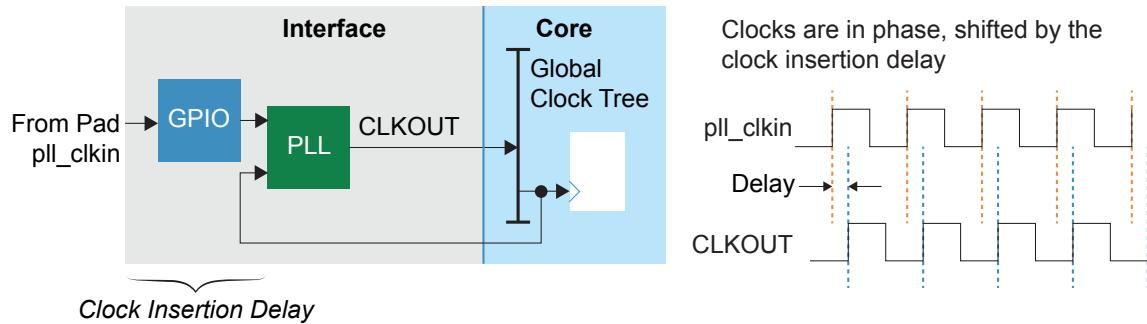
The clock at the global clock network is in phase with the reference clock, shifted by the insertion delay. The PLL compensation delay is equal to the core clock network delay. So the PLL compensation delay and the global clock network delay cancel out. You subtract the core clock network delay from the base clock insertion delay to get the actual delay.



**Important:** If you are creating a source-synchronous design, you should use this mode.

Remember, if your design is edge aligned, you need to use a phase shift on the PLL's CLKOUT to move the signal into the middle of the sampling window (90° for DDIO and 180° for SDIO).

Figure 4: Core Feedback Mode Delay



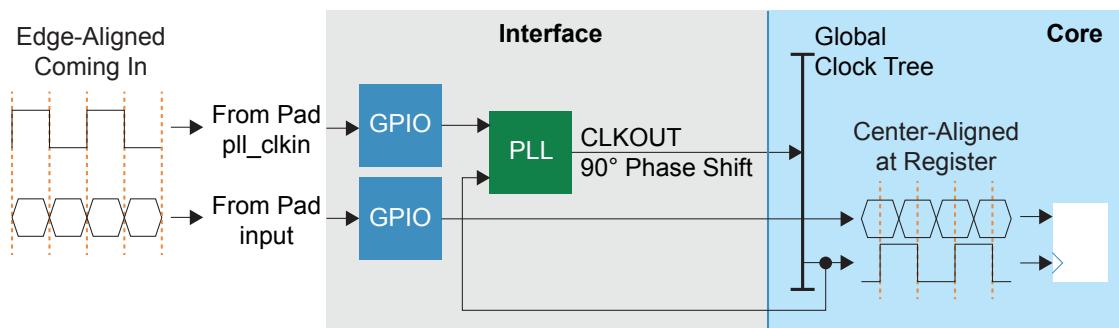
# Implementing a Source-Synchronous Receiver

After discussing the delays and PLL mode, it is time to understand the whole receiver setup. To align the clock signal, the incoming clock should feed a PLL and the PLL should be in core feedback mode.

If the data is center aligned with the clock when it enters the FPGA, no PLL phase shift is required. If it is edge aligned, use a 90° (DDIO) or 180° (SDIO) phase shift.

The data should be one or more GPIO in input mode. If you want to use DDIO, choose **Register Option > register** in the Interface Designer to enable the DDIO options. (Refer to the [Trion Interfaces User Guide](#) or [Titanium Interfaces User Guide](#) for more information on these options.)

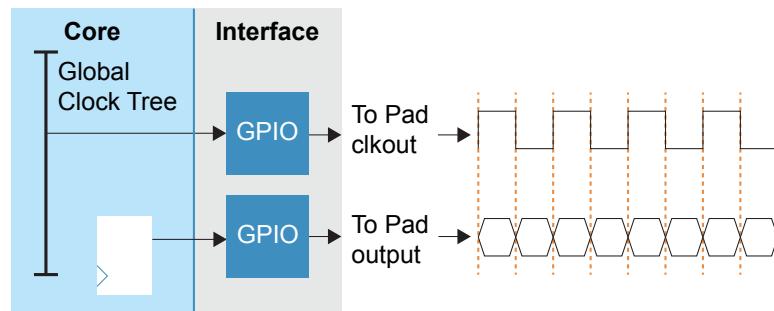
*Figure 5: Source-Synchronous DDIO Receiver*



# Implementing a Source-Synchronous Transmitter

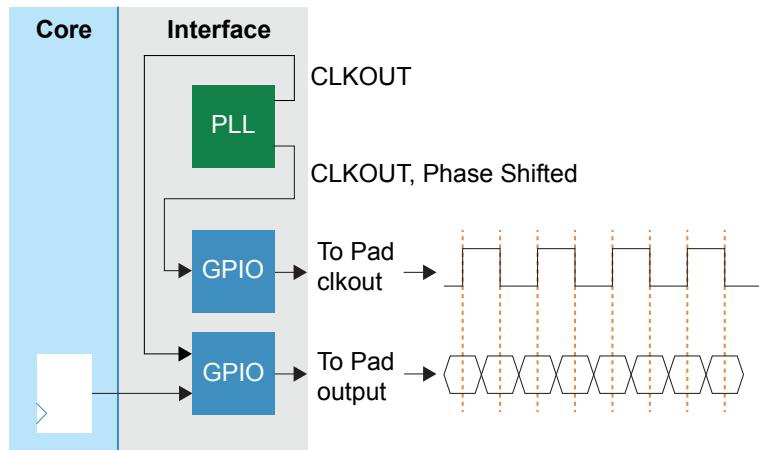
Setting up a source-synchronous transmitter is actually very simple. Use the I/O registers for the output data and use a GPIO in `clkout` mode to generate the transmitted clock signal. This configuration generates edge-aligned clock/data, which is the normal method used in FPGAs.

*Figure 6: Edge-Aligned Output*



If you need the transmitter to have a center-aligned clock, you need to use two outputs from the PLL with the normal clock driving the I/O registers and the phase-shifted clock driving the GPIO's clkout.

*Figure 7: Center-Aligned Output*



## Conclusion

Clocking source-synchronous designs is straightforward if you remember to follow the key guidelines described in this document. Use a PLL in core feedback mode and phase shift the clock to center align it with the data.

## Revision History

*Table 1: Revision History*

Date	Version	Description
March 2022	1.0	Initial release.