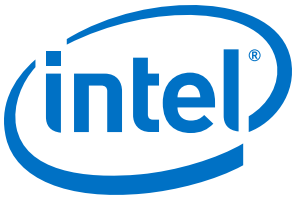


AN 756: Altera GPIO to Altera PHYLite Design Implementation Guidelines

AN-756
2017.05.08


Subscribe


Send Feedback

Contents

1 AN 756: Altera GPIO to Altera PHYLite Design Implementation Guidelines

Intel offers the Altera PHYLite for Parallel Interfaces IP core as an alternative solution for the Altera GPIO IP core for high speed applications that fail to close timing.

The Altera PHYLite for Parallel Interfaces IP core (or Altera PHYLite IP core) enables your design to achieve better timing performance. Intel recommends that you use the Altera PHYLite IP core for general purpose I/O (GPIO) data rates higher than 200 Mbps to achieve optimal out signal.

Note: Switching an Altera GPIO design to the Altera PHYLite IP core is very resource intensive.

- The Altera PHYLite IP core uses up the IOPLL, the `phy_clk` network, in the particular I/O bank.
- You may not be able to use the external memory interfaces and the Altera PHYLite IP core as normal memory protocols in this I/O bank unless they operate at the frequency multiplication of the Altera GPIO interface being switched.
- You also cannot use the Altera LVDS SERDES IP core in the same I/O bank with Altera PHYLite IP core.

Related Links

- Altera PHYLite for Parallel Interfaces IP Core User Guide
  Provides more information about the PHYLite IP Core.
- Altera GPIO IP Core User Guide
  Provides more information about the GPIO IP Core.

1.1 Implementing the Altera PHYLite Design

You can generate the Altera PHYLite design using the Intel® Quartus® Prime software, and customize the settings.

To implement your Altera GPIO IP core design to the Altera PHYLite IP core, follow these steps:

1. Generate a PHYLite design in the Quartus Prime software. Customize the design based on the Parameter Settings on page 4.
2. Connect the modules and the input and output ports, as shown in the following figures.

Figure 1. Input Interface

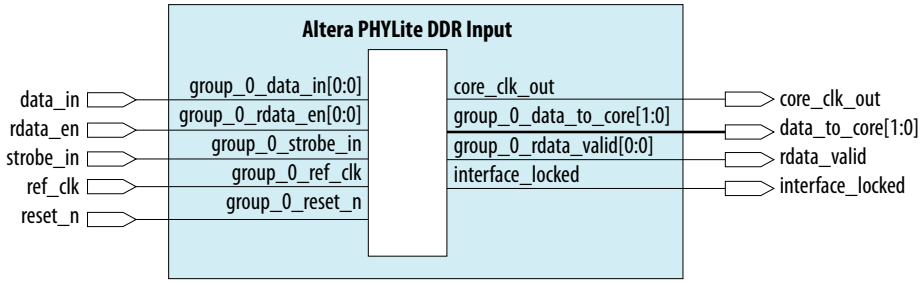
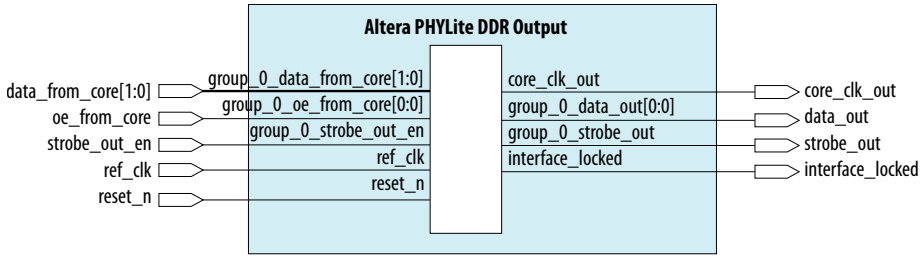


Figure 2. Output Interface



1.1.1 Parameter Settings

Follow the setting guidelines to instantiate a 1-bit single data rate (SDR) or double data rate (DDR) design.

Table 1. Parameter Settings Guidelines

Parameter	Setting	Notes
Number of groups	1	–
General Tab		
Interface clock frequency	Configure the frequency based on the required data rate.	The interface clock frequency is equal to data rate speed for SDR and half of the data rate speed for DDR.
Use core PLL reference clock connection	Turn on if your reference clock source is from the output of PLL or clock source from the core.	–
Use recommended PLL reference clock frequency	Turn on if the default frequency matches your reference clock frequency. Else, turn off the option and choose the desired reference clock frequency from the drop down list.	–
Clock rate of user logic	Specify the clock frequency to full, half, or quarter, based on the core logic.	For example, if the data rate speed sent from the FPGA to the external device is toggling at 200 Mbps in DDR mode, a half rate interface means that the user logic in the FPGA runs at 100 MHz.
I/O standard	Specify the desired I/O standard.	–
Group 0 Tab		

continued...

Parameter	Setting	Notes
Note: In the Altera PHYLite IP core, the `strobe_in` and `strobe_out` ports are equivalent to the source synchronous clock interface to the input and output data.		
Pin type	Specify based on the direction of the data pins.	–
Pin width	Specify the desired data width.	–
DDR/SDR	Specify the desired data rate mode.	–
Read Latency (When you configure the Altera PHYLite IP core as an input interface.)	Specify the expected read latency.	For example, a design with an external clock frequency of 100 MHz in full rate has a valid read latency of 3–63 external interface clock cycles.
Capture strobe phase shifty	Configure based on the desired phase shift of the input strobe relative to the input data.	For example in DDR mode, configure the phase shift to 90° would shift the edge-aligned input data/strobe to center-alignment at the read FIFO.
Write Latency (When you configure the Altera PHYLite IP core as an output interface.)	Specify a value within 0–3	Indicates the number of external interface clock cycles to delay the output data.
Use output strobe	Turn on if you want to enable the output strobe pin.	–
Output strobe phase	Specify based on the desired phase relationship between data and strobe being output from the IP core.	For example in DDR mode, configuring the phase shift to 90° would shift the strobe to center align with the output data
Data configuration	Specify the pin configuration of the data to be used as single ended or differential signaling.	–
Strobe configuration	Specify the type of strobe pin configuration to be used as single ended or differential signaling.	–
Use Default OCT Values	Turned on by default.	Turn off if you want to configure your desired input or output OCT values or you do not want any termination.
Generate Input/Output Delay Constraints for this group	Turned on by default.	Specifies the input/output delay setup or hold constraint against the input/output strobe of the group.

Related Links

- Altera GPIO Parameter Settings
- Altera PHYLite for Parallel Interfaces Parameter Settings

1.2 Mapping the Interface Signals

There are several differences in the interface signals between Altera GPIO and Altera PHYLite IP cores.

Table 2. Mapping the Altera GPIO Interface Signals to Altera PHYLite IP Core

Altera GPIO IP Core	Direction	Altera PHYLite IP Core
ck	Input/Output	ref_clk
aclr	Input/Output	reset
datain	Input	data_in
dataout_[h/l]	Input	data_to_core[1:0]
datain_[h/l]	Output	data_from_core[1:0]
oe	Output	oe_data
dataout	Output	data_out

Table 3. Additional Altera PHYLite Input and Output Signals

PHYLite	Direction	Description
interface_locked	Input/Output	Similar to the lock signal from the I/O phase locked loop (IOPLL). Typically the Altera GPIO IP core uses the IOPLL to generate the clock signal.
core_clk_out	Input/Output	Similar to the clock signal used for periphery-to-core or core-to-periphery transfer in the Altera GPIO IP core.
rdata_en	Input	Drive this signal high to ensure the input data/strobe port is always ready to receive incoming data.
strobe_in	Input	Sampling clock for the Altera PHYLite IP core that captures the input data.
strobe_out	Output	A generated clock/strobe signal from the PHYLite IP core. Similar to the synchronous clock out signal when using the GPIO IP core.
strobe_out_en	Output	Drive this signal high to ensure the Altera PHYLite IP core is always generating the source synchronous clock/strobe through the strobe_out port.
strobe_out_in[1:0]	Output	Tie strobe_out_in[0] to high and strobe_out_in[1] to low to generate a clock signal. This interface signal is generated in a similar manner you use the Altera GPIO IP core to generate the clock signal.
rdata_valid	Input	An output signal from the Altera PHYLite IP core that indicates the IP core has transmitted the data and the data is ready to be captured by the user logic.

Related Links

- Altera PHYLite for Parallel Interfaces IP Core Placement Restrictions
  Provides more information about the pin placement restrictions.
- Altera GPIO Interface Signals

- Altera PHYLite for Parallel Interfaces Signals

1.3 Performance Comparison

The performance comparison between the Altera GPIO and Altera PHYLite for Parallel Interfaces IP cores measures the setup and hold timing slack window at 200 Mbps.

The comparison between Altera GPIO and Altera PHYLite for Parallel Interfaces IP cores are based on the preliminary timing model in the Quartus Prime software version 15.1, using Arria 10 device (10AX115S2F45I2SGE2). The analysis is done using a simplified design with no user logic in the IP cores.

Note: The timing slack in the following tables do not account for any setup or hold requirements at the receiver or any channel skew between clock and data.

Table 4. Timing Analysis for Source Synchronous I/O Using GPIO Path

Input/ Output	SDR/DDR	Architecture	Clock Network	Worst Setup Slack	Worst Hold Slack	Slack Window (Setup + Hold)[1]
GPIO Input	DDR	• Data on 0° • Clock on 90° (center-aligned)	Global Clock	1.093	1.177	2.270
			Regional Clock	1.149	1.235	2.384
			Periphery Clock	1.360	1.415	2.775
	SDR	• Data on 0° • Clock on 180° (center-aligned)	Global Clock	1.188	1.227	2.415
			Regional Clock	1.214	1.223	2.452
			Periphery Clock	1.409	1.404	2.462
GPIO Output	DDR	• Data on 0° • Clock on 90° (center-aligned)	Global Clock	0.023	1.831	1.854
			Regional Clock	0.129	1.844	1.973
			Periphery Clock	0.193	1.968	2.161
	SDR	• Data on 0° • Clock on 180° (center-aligned)	Global Clock	0.835	1.541	2.376
			Regional Clock	0.408	1.738	2.146
			Periphery Clock	0.668	1.842	2.186

1 Based from multi corner timing analysis.

Table 5. Timing Analysis for Source Synchronous I/O Using Altera PHYLite IP Core

Input/ Output	SDR/DDR	Architecture	Clock Network	Worst Setup Slack	Worst Hold Slack	Slack Window (Setup + Hold)
Input (1-bit Input)	DDR	• Data on 0° • Strobe on 90° (center-aligned)	PHY clock	2.054	1.995	4.049
	SDR	• Data on 0° • Strobe on 180° (center-aligned)		2.064	2.020	4.084
Output (1-bit Input)	DDR	• Data on 0° • Strobe on 90° (center-aligned)		2.197	2.216	4.413
	SDR	• Data on 0° • Strobe on 180° (center-aligned)		2.239	2.209	4.448

In summary, the timing analysis on the source synchronous I/O implementation on the PHYLite IP core provides much larger timing slack window compared to the GPIO IP core. The PHYLite IP core uses the PHY clock network instead of the global clock networks in the core. The change in the clock network enables the PHYLite IP core to achieve better timing performance and avoid core noise effect.

1.4 Use Case Examples

You use the use case examples as guidelines to migrate the Altera GPIO IP core application to the Altera PHYLite IP core.

1.4.1 Use Case 1: Source Synchronous I/O Interface

Example 1 shows how you can migrate the source synchronous I/O interface implemented using GPIO to the Altera PHYLite IP core.

Figure 3. Migration of Source Synchronous Output Interface Using GPIO (Output Path) to Altera PHYLite

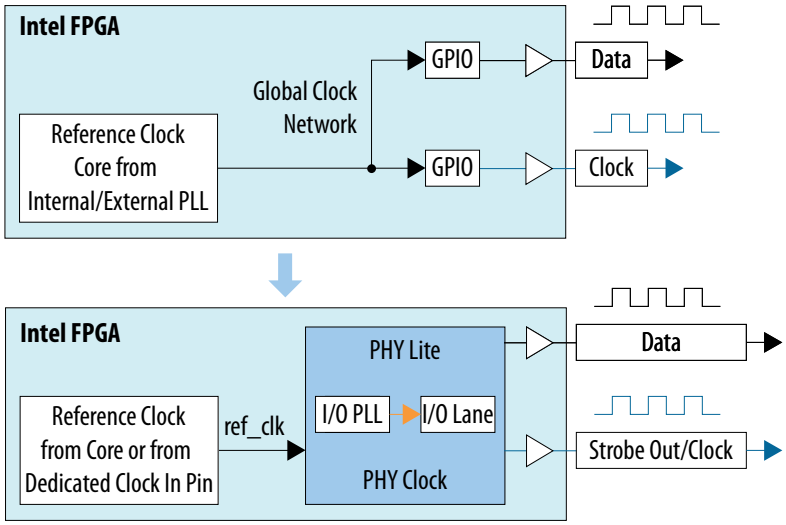
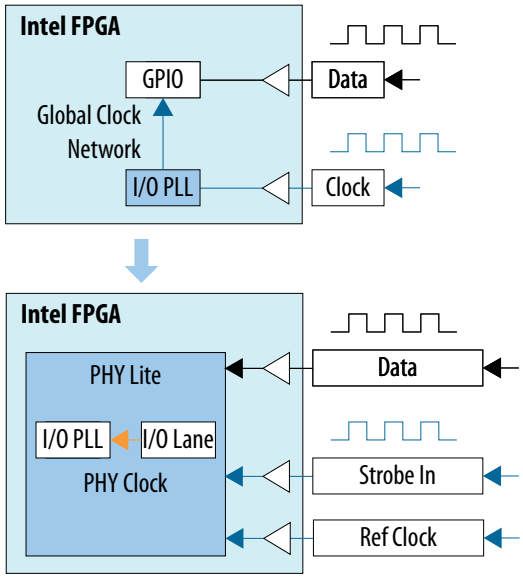


Figure 4. Migration of Source Synchronous Input Interface Using GPIO (Input Path) to Altera PHYLite



1.4.2 Use Case 2: Driving Data from FPGA through GPIO to External Device

Example 2 describes the driving of data from FPGA through GPIO to an external device with a shared oscillator clock. This case uses the PLL output clock as a sampling clock to the user logic.

The oscillator clock goes into the FPGA GCLK network to drive the output data through GPIO to Off-chip. Off-chip clock uses the shared oscillator. The clock source of the other user logic comes from the PLL clock out.

You can use one of the two following methods to migrate the GPIO to Altera PHYLite IP core using the output clock from the IOPLL supplied to the user logic:

1. Generate the output clock from the Altera PHYLite IP core's output clock.
   - The PHYLite IP core can export 4 additional IOPLL output clocks based on the specified configuration.
   - You can calculate the actual support clock frequency based on the formula:

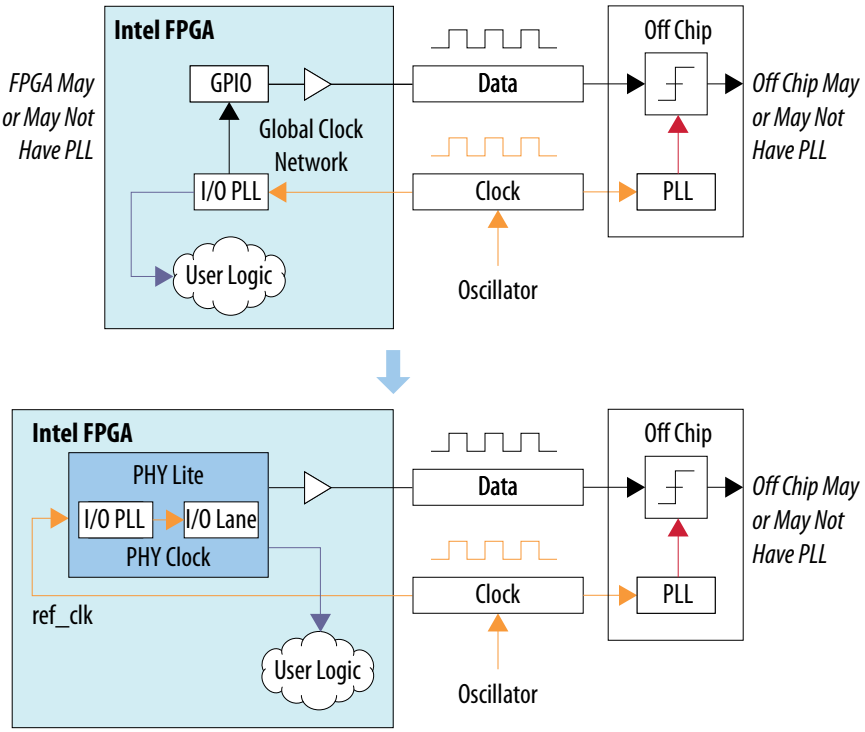
```
Output Clock Frequency = VCO Frequency/ c Counter
```

     — c Counter = Integers within 1–511

     — Voltage-controlled oscillator (VCO) frequency = (Memory/Interface clock frequency) × (VCO frequency multiplication factor)

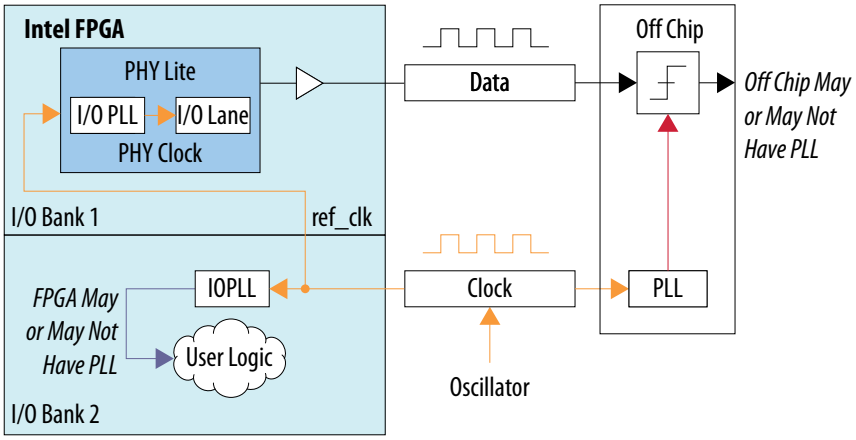
     Note: Refer to the Altera PHYLite for Parallel Interfaces IP core parameter editor for more information.

Figure 5. Generating Output Clock from PHYLite Output Clock



2. Instantiate a new IOPLL in the adjacent I/O bank (where the PHYLite block resides), if the Altera PHYLite IOPLL output clocks do not support the desired output clock frequency.

Figure 6. Instantiating a New I/O PLL

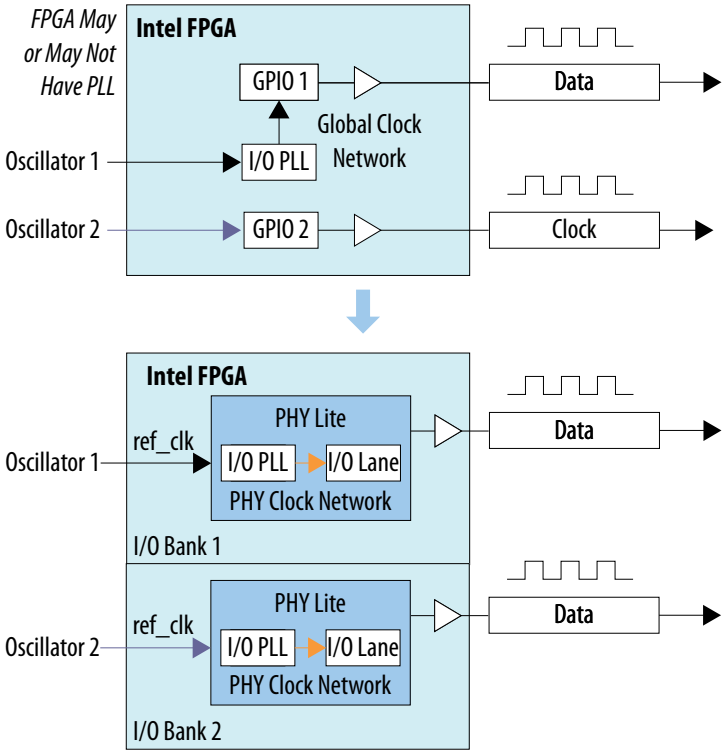


1.4.3 Use Case 3: Multiple-Speed Parallel Interfaces

Example 3 shows multiple-speed parallel interfaces with specific reference clock frequency.

To migrate the multi-speed parallel interface GPIO solution to Altera PHYLite, you must split the solution into two different I/O banks to run at different data rate.

Figure 7. Multiple-Speed Parallel Interfaces



1.4.4 Use Case 4: GPIO Interface Using 3.0V/2.5V I/O Standard

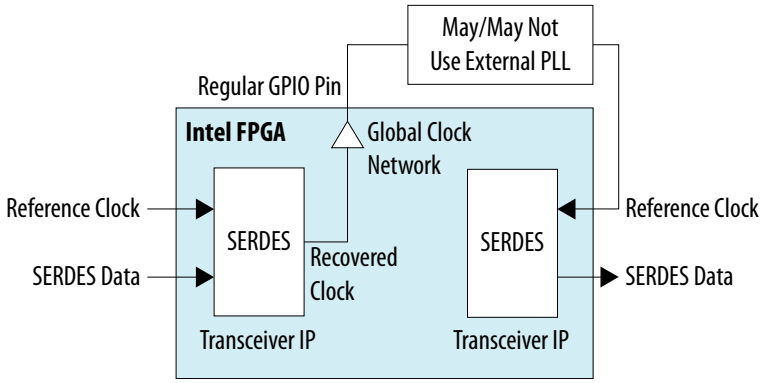
The current version of the Altera PHYLite IP core does not support 3.0V/2.5V I/O standard.

You can manually set the pin to use 3.0V/2.5V I/O standard through the Quartus Prime assignment editor.

1.4.5 Use Case 5: Generating Output Clock through GPIO Output Pin

Example 5 describes the generation of an output clock through a GPIO output pin. The source of the output clock is from the transceiver SERDES recovered clock.

Figure 8. Recovered Clock from Transceiver Produced Through GPIO Interface



Generate the output clock through the PHYLite IP core if you use a regular I/O.

Figure 9. Recovered Clock Feeding Through I/O PLL to Generate Output Clock Using Dedicated Clock Out Pin

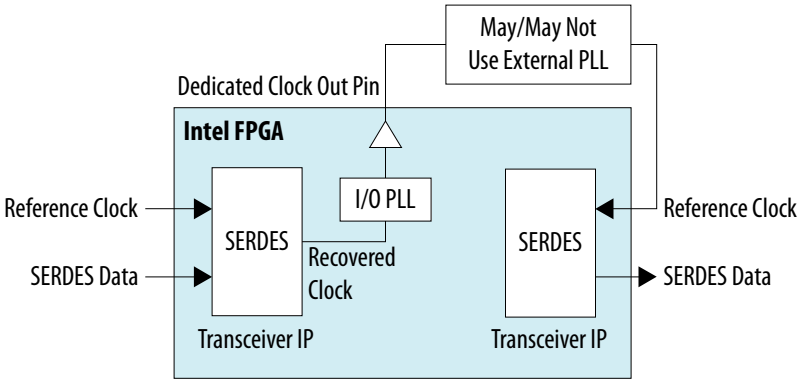
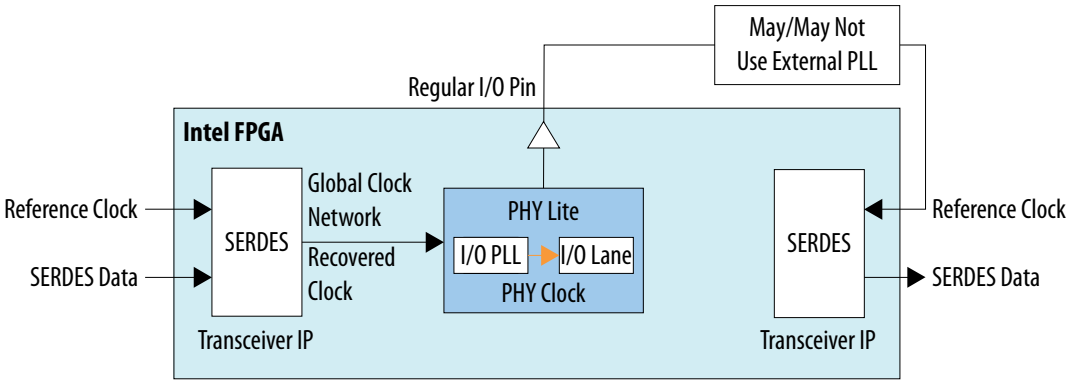


Figure 10. Recovered Clock Feeding Through Altera PHYLite IP Core to Generate Output Clock Using Regular GPIO Pin



Generating the recovered clock through a dedicated clock out pin from the IOPLL or through the Altera PHYLite IP core results in equivalent jitter performance. The preferred solution is to use the IOPLL, because using the Altera PHYLite IP core is resource intensive.

1.5 Altera GPIO to Altera PHYLite Design Example

You can use the provided design example as a reference to instantiate the Altera PHYLite IP core (input and output in double-data rate mode).

The Altera GPIO to Altera PHYLite design example provides the Altera PHYLite configuration to mimic the GPIO input and output path usage and the expected behavior in simulation and hardware.

The design example consists the following components:

- Pattern generator (`tx_data_output`) to generate fixed serial 1100 pattern transmitting through the Altera PHYLite output (`ddr_out`)
- Two Altera PHYLite instances configured to mimic the GPIO input and output path usage to transmit and receive source synchronous data respectively.

The `reset_n` signal connects to the push button switch to control the reset port for the pattern generator and Altera PHYLite input and output instances.

Related Links

Altera GPIO to Altera PHYLite Design
Provides the design files for the Altera GPIO to Altera PHYLite design example.

1.5.1 Simulation Diagrams

The simulation diagrams show the behavior of the signals in the design example.

Figure 11. Pattern Generator Starts Generating Fixed 1100 Pattern

The pattern generator begins generating the fixed pattern to the Altera PHYLite output when the Altera PHYLite output generates the clock signal to the core through the `core_clk_out` port.

Figure 12. Interface Locked Signals from Altera PHYLite Instances

The `rdata_en`, `oe`, and `strobe_out_en` signals of the Altera PHYLite input (`ddr_in`) and output (`ddr_out`) are always asserted after the `interface_locked` signal of the Altera PHYLite instances transitions from low to high. The `interface_locked` signal has to be high to indicate that all the necessary clocks in the Altera PHYLite instances function correctly.

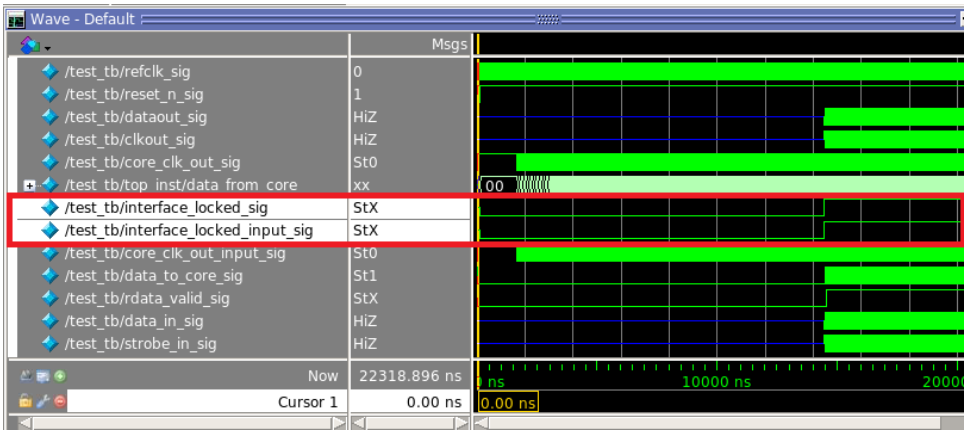


Figure 13. Generated Strobe Signal Edge-Aligned with Data Out Signal

To simulate the behavior of the Altera PHYLite output, the fixed serial 1100 pattern transmits through the Altera PHYLite output `dataout` pin after the `interface_locked` signal locks. The `strobe_out_en` signal remain asserted to generate the synchronous clock and strobe signals transmitted through the `clkout` pin with the Altera PHYLite output serial data.

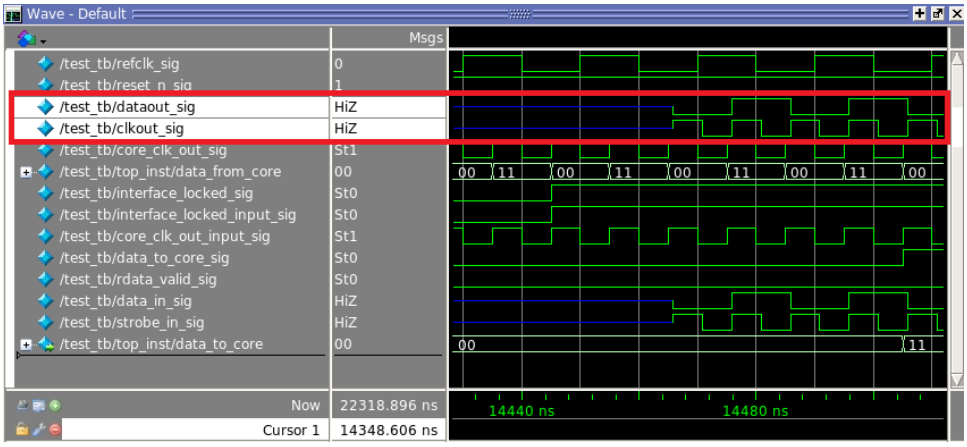


Figure 14. Data Out and Clock Out Signals Loopback

To simulate the behavior of Altera PHYLite input instance, the `dataout` and `clkout` signals loop back to the `data_in` and `strobe_in` of the Altera PHYLite input interface.

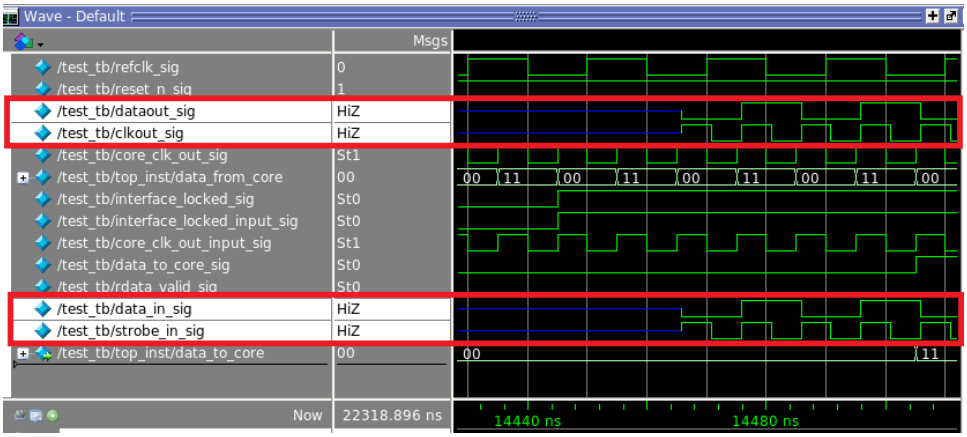


Figure 15. PHYLite Input Instance Transfers Valid Loopback Signals to the IP Core

The loopback signals received by the PHYLite input instance transfer to the IP core when the `rdata_valid` signal is high.

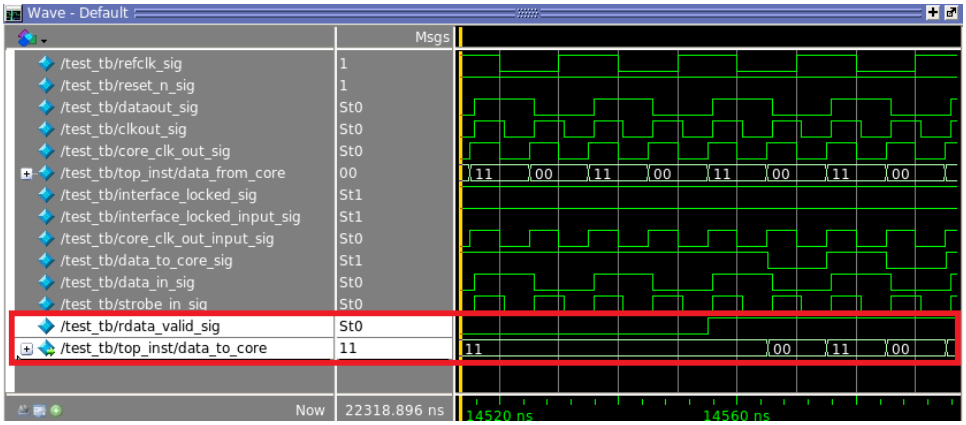
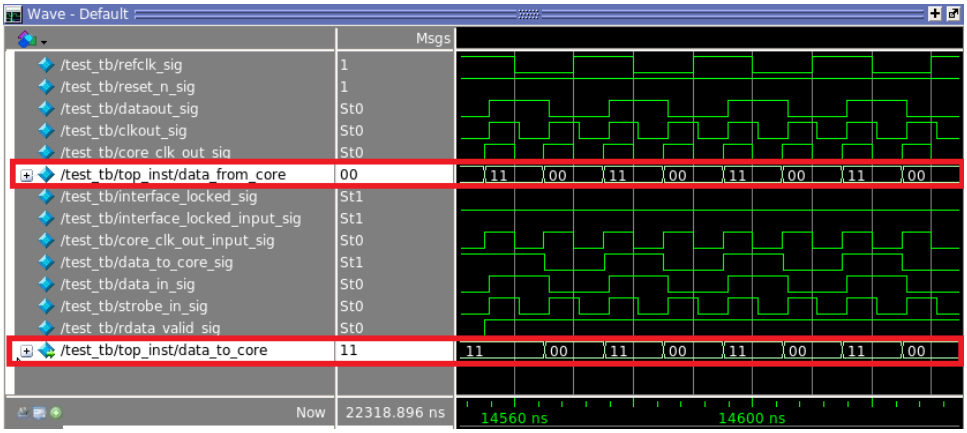


Figure 16. Identical Received and Generated Data

The data transfered to the IP core should look identical to the data generated by the pattern generator in the IP core.



1.5.2 Design Verification

You can verify the Altera GPIO to Altera PHYLite design example using the Intel Arria® 10 FPGA development kit.

To verify the design example, you require the following hardware:

- Arria 10 FPGA development kit
- Loopback FPGA Mezzanine Card (FMC) attached to the FMC port B (FMCB)
- Intel FPGA Download Cable II

The design example demonstrates a simple loopback that loops back the fixed serial 1100 pattern transmitted from the Altera PHYLite output (`ddr_out`) to the Altera PHYLite input (`ddr_in`) through the Loopback FMC daughter card.

The Altera PHYLite output signals from the `dataout` and `clkout` pins connect directly to the `data_in` and `strobe_in` pins that feed the Altera PHYLite input.

The data received from the Altera PHYLite input after you program the SOF in the Signal Tap Logic Analyzer should be the same as the data transmitted from the IP core through the Altera PHYLite output.

Figure 17. Signal Tap Results for Data from the IP Core

This figure shows the Signal Tap results captured for the data from the IP core, (using the `auto_signaltap_0` instance in the `.stp` file).

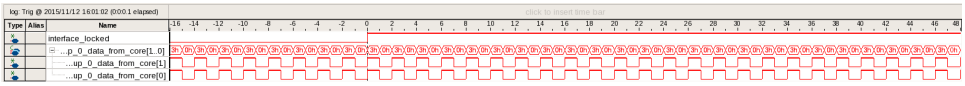
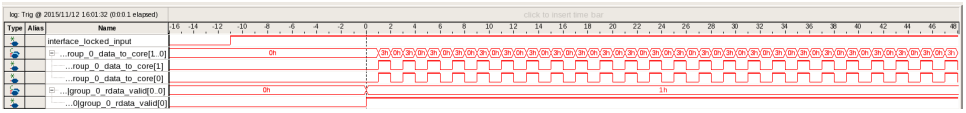


Figure 18. Signal Tap Results for Data to the IP Core

This figure shows the Signal Tap results captured for the data to the IP core after the `rdata_valid` signal goes high, (using the `auto_signaltap_1` instance in the `.stp` file).



1.6 Document Revision History

The following table lists the revision history for this document.

Table 6. Document Revision History

Date	Version	Changes
May 2017	2017.05.08	Rebranded as Intel.
December 2016	2016.12.30	Edited the link to the design files.
December 2015	2015.12.14	Initial release.