External PHY Support in PCI Express MegaCore Functions

May 2007, ver. 1.0 Application Note 443

Introduction

The PCI Express Compiler generates customized PCI Express MegaCore® functions that you can use to design PCI Express endpoints. The PCI Express MegaCore functions are compliant with *PCI Express Base Specification Revision 1.1* or *PCI Express Base Specification Revision 1.0a*. The functions implement all required and most optional features of the specification for the transaction, data link, and physical layers.

If you target the MegaCore function for Stratix® GX or Stratix II GX devices, the function includes a complete PHY layer, including the MAC, PCS, and PMA layers. If you target other Altera® device architectures, the PCI Express Compiler generates the MegaCore function with Intel's PIPE interface. This interface makes the MegaCore function usable with other PIPE-compliant external PHY devices, allowing for a lower-cost solution.

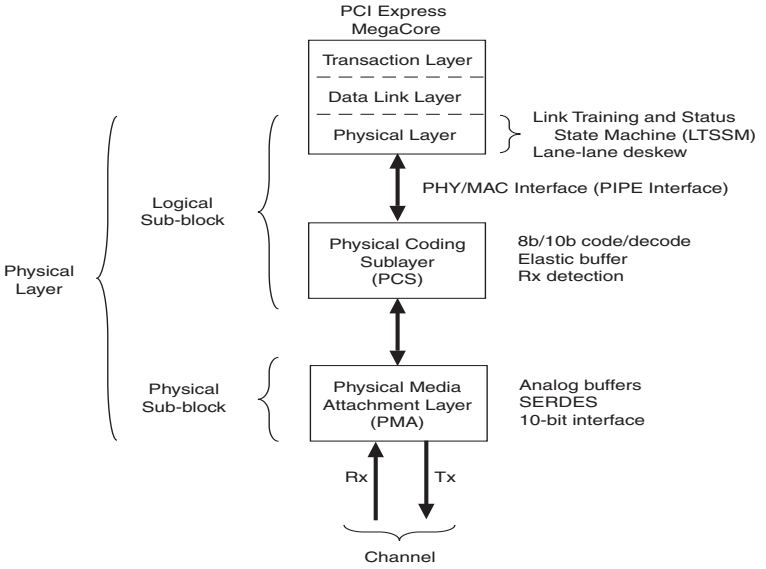
Some of the external PHYs, like the TI XIO1100 PHY and the Philips PX1011A PHY, have an interface that is derived from the Intel PIPE interface, so additional custom logic is required to interface directly with these PHYs. The PCI Express Compiler generates this custom logic, enabling you to connect the PCI Express MegaCore functions directly to the external PHY. In addition to the TI XIO1100 and the Philips PX1011A external PHYs, the PCI Express Compiler supports a wide range of custom PHYs that use 8-bit or 16-bit SDR and 8-bit DDR with or without Source Synchronous Transmit clock modes. Currently the support for external PHYs is limited to x1 and x4 implementations, though the PCI Express Compiler can also support x8 implementations.

This application note provides an introduction to the PIPE interface and focuses on external PHY support in the PCI Express MegaCore function. It covers design details such as clocking, reset, and board-level guidelines. This application note also provides guidelines for debugging the PIPE interface between the FPGA and the external PHY. It provides detailed information on how to interface the TI XIO1100 and Philips PX1011A external PHYs with the Altera PCI Express MegaCore function.

The Intel PIPE Interface

PIPE stands for PHY Interface for the PCI Express Architecture. The PIPE specification was developed by Intel. Although it is not a standard like PCI Express, the PIPE interface is the most widely adopted interface between the PHY and the MAC layers (Figure 1).

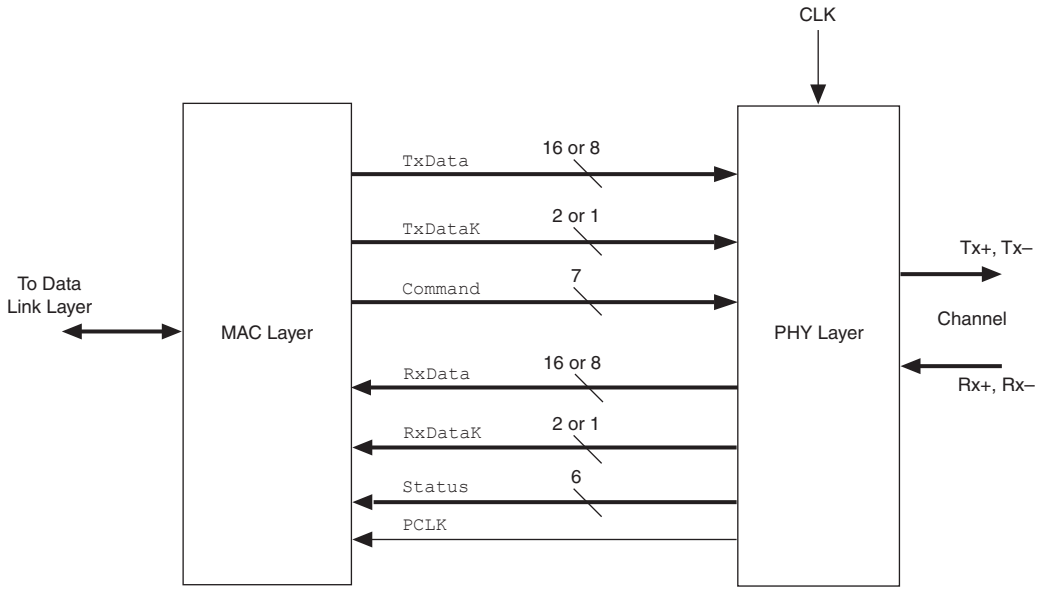
Figure 1. PIPE Interface



The PCI Express PHY layer handles the low-level PCI Express protocol and signaling, including 8b/10b encoding, data serialization and deserialization, analog buffers, and receiver detection. The MAC layer handles state machines for Link Training and Status and lane-lane deskew in addition to interfacing with the upper layers.

Figure 2 shows an overview of the data, control, and status signals between the PHY and MAC layers.

Figure 2. PIPE Signals



The PIPE specification allows for two different data path widths: 16-bits operating at 125 MHz and 8-bits operating at 250 MHz. Table 1 describes the PIPE interface signals for an x1 implementation.

Table 1. PHY Interface Signals (Part 1 of 2)

Signal Name	Direction	Description	Availability
phystatus_ext	I	PIPE Interface phystatus signal. PHY is signaling completion of the requested operation.	Always
powerdown_ext[1:0]	O	PIPE Interface powerdown signal, requesting the PHY to enter the specified power state (P0, P0s, P1, or P2).	Always
refclk	I	Input clock connected to the PIPE Interface pclk signal from the PHY. 125 MHz clock used to clock all of the status and data signals.	Always
pipe_rstn	O	Asynchronous reset to external PHY. The reset is driven by the FPGA and expects a pull-down resistor on the board. During FPGA configuration, the pull-down resistor resets the PHY. The FPGA then drives the PHY out of reset. This signal is only in MegaCore functions that are configured for the external PHY.	Always

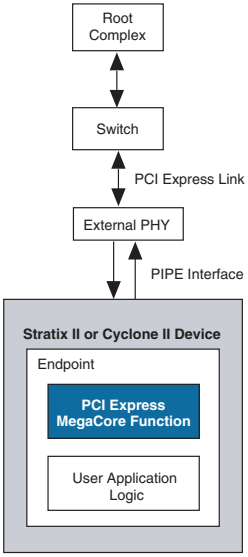
Table 1. PHY Interface Signals (Part 2 of 2)

Signal Name	Direction	Description	Availability
pipe_txclk	O	Source Synchronous Transmit Clock signal for clocking Tx Data and Control signals going to the PHY.	Only in modes that have the TxClk
rxdata_ext[15:0]	I	PIPE Interface Lane 0 Rx Data signals, carries the parallel received data for lane 0 of the PCI Express bus. For the 8-bit PIPE mode, only rxdata_ext[7:0] is available.	Always
rxdatak_ext[1:0]	I	PIPE Interface Lane 0 Rx Data K-character flags. Used to separate control and data symbols. For the 8-bit PIPE mode, only rxdatak_ext is available.	Always
rxelecidle_ext	I	PIPE Interface Lane 0 Rx Electrical Idle Indication.	Always
rxpolarity_ext	O	PIPE Interface Lane 0 Rx Polarity Inversion Control. This signal instructs the PHY layer to do a polarity inversion on the 8b/10b receiver decoding block.	Always
rxstatus_ext[1:0]	I	PIPE Interface Lane 0 Rx Status flags. Used to encode receive status and error codes for the receive data stream and receiver detection.	Always
rxvalid_ext	I	PIPE Interface Lane 0 Rx symbol lock and valid indication.	Always
txcompl_ext	O	PIPE Interface Lane 0 Tx Compliance control. Used to force the running disparity to negative in compliance mode.	Always
txdata_ext[15:0]	O	PIPE Interface Lane 0 Tx Data signals, carries the parallel transmit data for lane 0 of the PCI Express bus. For the 8-bit PIPE mode, only txdata_ext[7:0] is available.	Always
txdatak_ext[1:0]	O	PIPE Interface Lane 0 Tx Data K-character flags. Used to separate control and data symbols. For the 8-bit PIPE mode, only txdatak_ext is available.	Always
txelecidle_ext	O	PIPE Interface Lane 0 Tx Electrical Idle Control.	Always
txdetectrx_ext	O	PIPE Interface Lane 0 transmit detect receive signal. Used to tell the PHY layer to start a receive detection operation or to begin a loopback.	Always

For more information about the PIPE interface, refer to the PIPE specification from Intel, *PHY Interface for the PCI Express™ Architecture*.

Figure 3 shows an example of a PCI Express system where the Altera PCI Express MegaCore function interfaces with an external PHY.

Figure 3. Example PCI Express System



External PHY Interface Modes

The Altera PCI Express MegaCore function supports a wide variety of external PHYs. The MegaWizard® Plug-in Manager for PCI Express contains three different external PHY selections:

- TI XIO1100
- Philips PX1011A
- Custom

Table 2 describes the different modes supported by the PCI Express Compiler.

Table 2. External PHY Interface Modes (Part 1 of 2)

PHY Interface Mode	Clock Frequency	Notes
16-bit SDR	125 MHz	In this generic 16-bit PIPE interface, both the `Tx` and `Rx` data are clocked by the `pclk` from the PHY.
TI XIO1100 16-bit SDR Mode (with Source Synchronous Transmit Clock)	125 MHz	This enhancement to the generic 16-bit PIPE interface adds a `TxClk` to clock the `TxData` source synchronously to the external PHY.

Table 2. External PHY Interface Modes (Part 2 of 2)

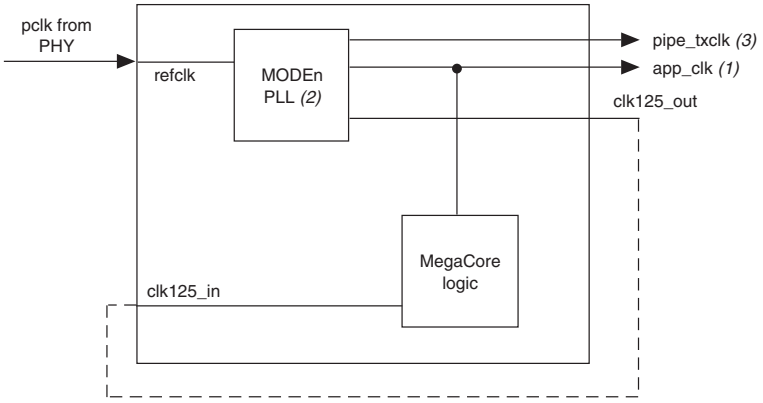
PHY Interface Mode	Clock Frequency	Notes
8-bit DDR	125 MHz	This double data rate version saves I/O pins without increasing the clock frequency. It uses a single `pclk` from the PHY for clocking data in both directions.
TI XIO1100 8-bit DDR Mode with SDR ctrl (with an 8-bit DDR Source Synchronous Transmit clock)	125 MHz	This double data rate version saves I/O pins without increasing the clock frequency. A `TxClk` clocks the data source synchronously in the transmit direction.
8-bit DDR Mode with `TxClk` and DDR ctrl	125 MHz	This double data rate version saves I/O pins without increasing the clock frequency. A `TxClk` clocks the data source synchronously in the transmit direction.
8-bit SDR	250 MHz	This is the generic 8-bit PIPE interface. Both the `Tx` and `Rx` data are clocked by the `pclk` from the PHY.
Philips 8-bit SDR Mode (with Source Synchronous Transmit Clock)	250 MHz	This enhancement to the generic 8-bit PIPE interface adds a `TxClk` to clock the `TxData` source synchronously to the external PHY.

The following sections cover design flow issues such as clocking, reset, and debug, when using an external PHY with the PCI Express MegaCore functions.

Clocking

Figure 4 shows the clocking for PCI Express MegaCore functions when they interface with an external PHY.

Figure 4. Clocking for PCI Express MegaCore Functions



Notes to Figure 4

(1) x1 implementations—internal 62.5 MHz clock
(2) 16-bit SDR Mode with a Source Synchronous TxClk does not include a PLL when the internal frequency is 125 MHz
(3) Source Synchronous transmit clock is available only in modes that require a source synchronous transmit clock

The `pclk` (parallel clock) output drives the `refclk` input of the MegaCore function. `refclk` drives the `inclock` input of the PLL. Based on the external PHY interface mode, the PLL settings are automatically generated. The PLL generates a 125 MHz clock (`clk125_out`). You must connect `clk125_out` to `clk125_in`. In addition, based on the core settings, the following clocks may be generated:

- `pipe_txclk`—Source synchronous transmit clock for modes that require a transmit clock
- `app_clk`—An optional 62.5 MHz application clock for x1 implementations

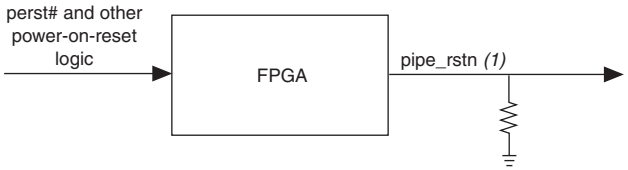
Based upon the external PHY mode, the PLL also generates other internal frequencies and phase offsets to capture the PIPE input signals or drive the PIPE output signals.

For more information about the frequencies and phase offsets, refer to the *PCI Express Compiler User Guide*.

PIPE reset (pipe_rstn)

The external PHY requires an asynchronous active low reset from the FPGA. Altera recommends that you drive this signal using the `npor` signal and have a pull-down on the board. On power-up the PHY is reset while the FPGA is being configured. When the FPGA configuration is complete, the FPGA will drive this signal high (Figure 5).

Figure 5. PIPE Reset



Note to Figure 5
(1) `pipe_rstn` = `npor` in user logic

For a description of the `npor` signal, refer to the *PCI Express Compiler User Guide*.

The MegaCore functions have additional reset requirements. Refer to the *PCI Express Compiler User Guide* for more information.

Quartus II Timing and Logic Assignments

The PCI Express Compiler generates a sample Tcl file (*<variation>*_**example_top.tcl**) that can be sourced in the Quartus® II software to set up timing constraints and logic options for the PIPE signals. The Tcl file sets up the following constraints:

- `pclk` frequency constraint (125 MHz or 250 MHz)
- Setup and Hold constraints for the input signals
- Clock-to-out constraints for the output signals
- Fast I/O register option

The compiler also generates MegaCore pin attributes, including pin direction, location, I/O standard assignments, and drive strength in the Pin Planner support files (*<variation>*.**ppf** and *<variation>*.**ppx**). If you launch the MegaWizard Plug-In Manager outside the Pin Planner application, you must explicitly load this file to use Pin Planner.

Board-Level Issues

The PIPE interface signals operate at 125 MHz or 250 MHz, so high speed design techniques should be used for your board design. Altera recommends that you simulate your board design to verify any board-level issues.

The length of the transmit and receive trace data paths do not need to match each other. However, the trace data paths of the transmit signals `tx_data0_ext`, `txdatak0_ext`, `txcompl0_ext`, `txdetectrx_ext`, `txelecidle0_ext` and transmit clock `pipe_txclk` must have the same length. Similarly, the trace data paths of the receive signals `rxdata0_ext`, `rxdatak0_ext`, `rxelecidle0_ext`, `rxvalid0_ext`, `rxstatus0_ext`, and `clk125_in` must have the same length.

For modes that do not require a source synchronous transmit clock, ensure that the board flight delays for the `pclk` and transmit signals are as small as possible.

Meeting the external PHY setup and hold requirements for the transmit signal is very complicated. Several of the external PHYs require additional board-level assignments and considerations. The requirements for the TI XIO1100 and the Philips PX1011A PHYs are described in "Interfacing with the TI XIO1100 PHY" on page 10 and "Interfacing with the Philips PX1011A PHY" on page 16.

Debug

In addition to the MegaCore functions, the PCI Express Compiler generates an example design and a Quartus II project. This example design enables you to quickly debug board-level issues and verify that the PIPE interface works correctly.

For more information about the example design, refer to the *PCI Express Compiler User Guide*.

To verify the PIPE interface is using the example design, you need to modify the pin number assignments for the PIPE signals and the `pcie_rstn` signal in the Quartus II project and then recompile the project. The example design includes some additional inputs (`local_rstn_ext`, `usr_sw`) and debug LED outputs that can be optionally set to an inactive state. The default values for the input signals can be tied to benign values and these additional signals will be synthesized away.

If you need further debug information, you can use the SignalTap II Logic Analyzer to monitor the PIPE signals and the MegaCore `ltssm` signal (`test_out[324:320]`). Use the 125 MHz `clk125_in` signal to sample the SignalTap II data.

Modifying PLL Settings

Based on your board delays, you may have to modify the PLL settings inside the core so that you can meet the timing constraints. Perform the following steps to modify the PLL settings:

1. Identify the PLL to be modified by viewing the PCI Express <*variation*>**.v** or <*variation*>**.vhd** file in a text editor.
2. Copy the corresponding PLL from the <*install*>**/lib** directory to your project directory, where <*install*> is the PCI Express Compiler installation directory.
3. Using the MegaWizard Plug-in Manager, modify the PLL settings.
4. Recompile the Quartus II project and verify that timing is met.

Interfacing with the TI XIO1100 PHY

The TI XIO1100 PHY is a PCI Express PHY that is compliant with *PCI Express Base Specification Revision 1.1*. It interfaces the PCI Express Media Access Layer (MAC) with a PCI Express serial link by using a modified PIPE interface. This interface is referred to as a TI-PIPE interface.

The TI-PIPE interface is a pin-configurable interface that can be configured as either an 8-bit or a 16-bit interface. The Altera PCI Express MegaCore function can support both these modes of operation.

For more information about interfacing with the TI XIO1100 PHY, refer to the *XIO1100 Data Manual* from Texas Instruments.

8-Bit DDR Mode Operation

The 8-bit TI-PIPE interface is a 250 MHz 8-bit parallel interface with the following buses:

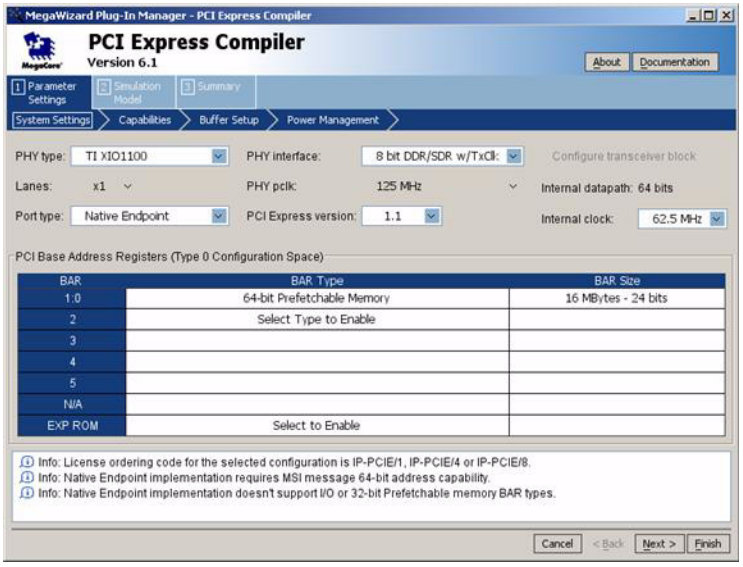
- An 8-bit output bus (RXDATA) that is clocked by the RXCLK output clock
- An 8-bit input bus (TXDATA) that is clocked by the TXCLK input clock

Both buses are clocked using Double Data Rate (DDR) clocking, in which the data transitions are on both the rising edge and the falling edge of the clock.

Core Generation

In the **System Settings** tab of the PCI Express MegaWizard Plug-In Manager, set PHY type to **TI XIO1100** and PHY interface to **8 bit DDR/SDR w/TxClk**, as shown in Figure 6.

Figure 6. TI 8-Bit DDR Mode Operation



In this PHY interface mode, the data signals operate a 250 MHz (clock frequency of 125 MHz in DDR Mode) and the control/status signals operate at 125 MHz. The MegaWizard Plug-In Manager automatically selects the correct setting for Power Management. The I/O standard for the TI-PIPE signals is 1.8 V LVTTL.

Additional Board Considerations

Figure 7 shows the mapping of the signals between the TI PHY and the Altera MegaCore function in the 8-bit DDR mode.

Figure 7. Mapping Signals in TI 8-Bit DDR Mode

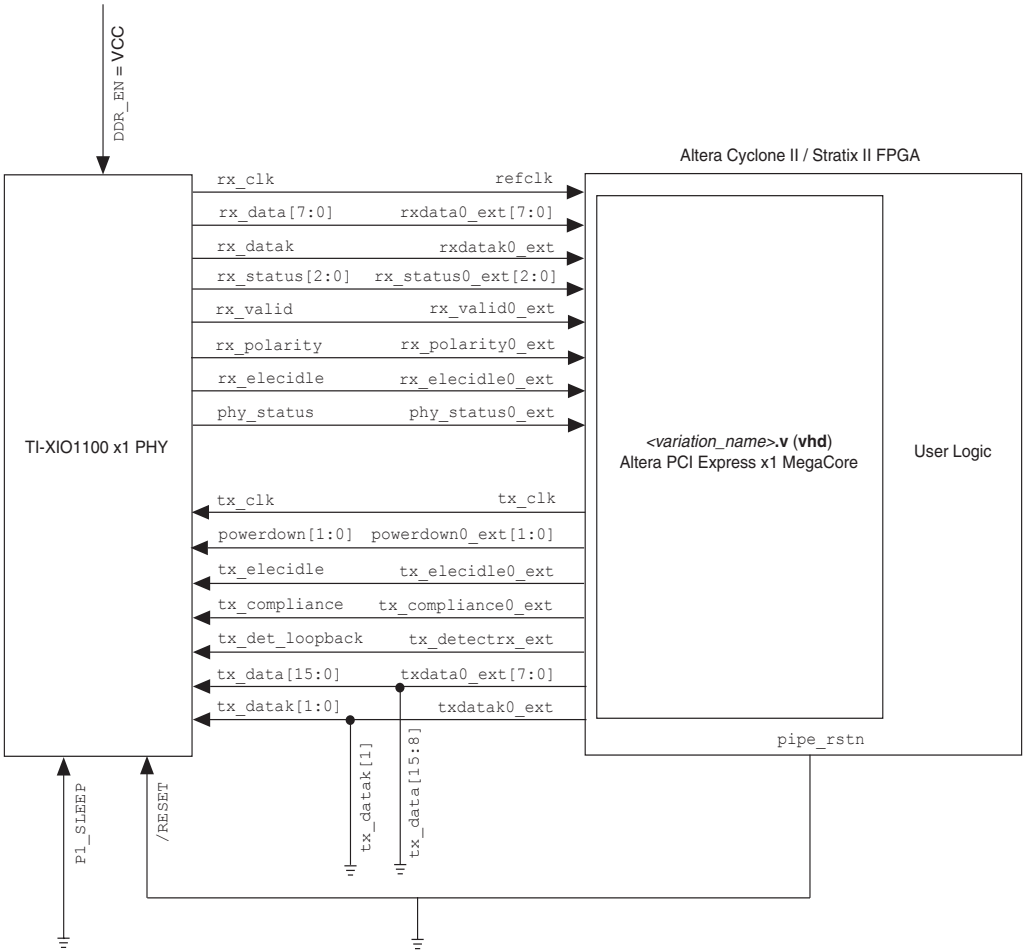


Table 3 shows the board-level assignments required to operate the PHY in this mode.

Table 3. Board-Level Assignments for TI 8-Bit Mode

TI Signal Name	Board-Level Assignment	Comments
DDR_EN	Pull up	Sets the TI PHY to operate in the DDR mode.
TX_DATA[15:8]	Pull down	These signals are not used in the DDR mode.
TX_DATAK[1]	Pull down	This signal is not used in the DDR mode.
/RESET (pipe_rstn)	Pull down	Asynchronous reset signal to the TI PHY. This signal must be driven by the FPGA in the user logic and have a pull-down on the board. On power-up the PHY is reset while the FPGA is being configured. When the FPGA configuration is complete, the FPGA drives this signal high.
CLK_SEL	Connected to a real pin/signal on the board	Selects the reference clock frequency. Refer to the TI data sheet for more information.
P1_SLEEP	Pull down	Refer to the TI data sheet for more information.

16-Bit SDR Operation

The 16-bit TI-PIPE interface is a 125 MHz 16-bit parallel interface with the following buses:

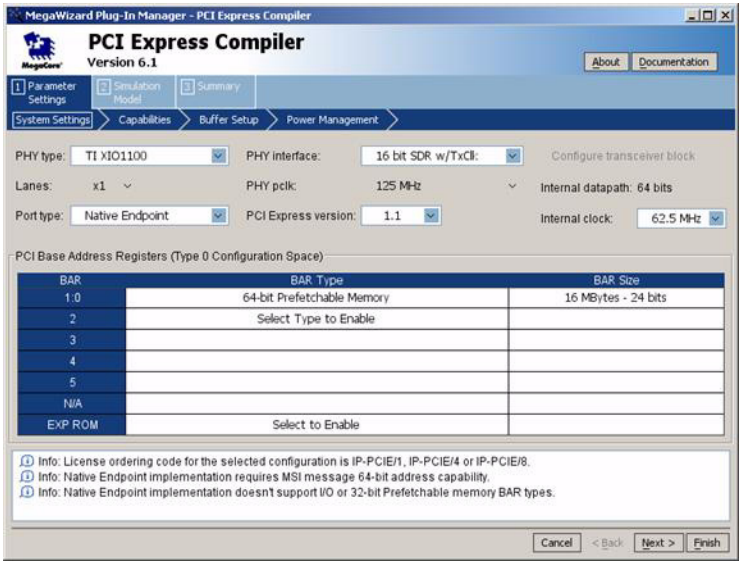
- A 16-bit output bus (RXDATA) that is clocked by the RXCLK output clock
- A 16-bit input bus (TXDATA) that is clocked by the TXCLK input clock

Both buses are clocked using Single Data Rate (SDR) clocking, in which the data transitions are on the rising edge of the associated clock.

Core Generation

In the **System Settings** tab of the PCI Express MegaWizard Plug-In Manager, set PHY type to **TI XIO1100** and PHY interface to **16 bit SDR w/TxClk**, as shown in Figure 8.

Figure 8. TI 16-Bit DDR Mode Operation



In this PHY interface mode, all data signals operate at 125 MHz. The MegaWizard Plug-In Manager automatically selects the correct setting for Power Management. The I/O standard for the TI-PIPE signals is 1.8 V LVTTL.

Additional Board Considerations

Figure 9 shows the mapping of the signals between the TI PHY and the Altera MegaCore function in the 16-bit SDR mode.

Figure 9. Mapping Signals in TI 16-Bit SDR Mode

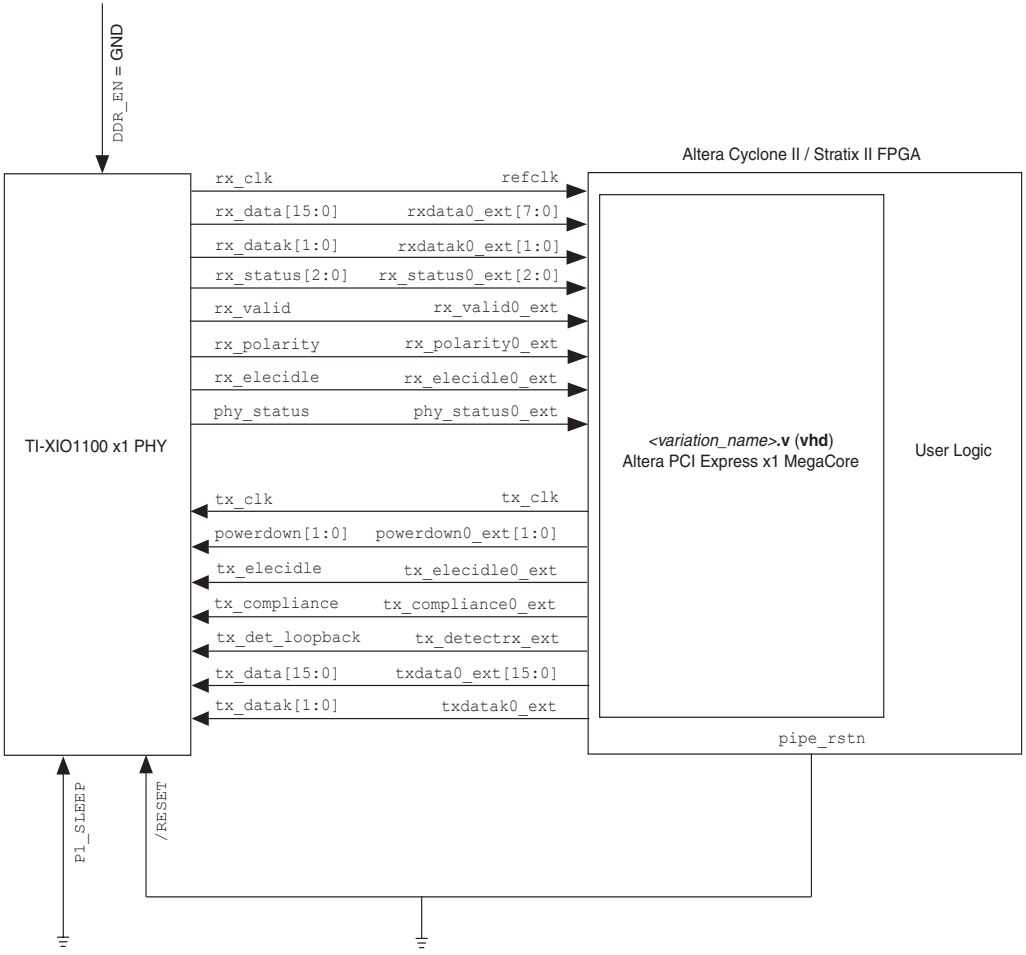


Table 4 shows the board-level assignments required to operate the PHY in this mode.

Table 4. Board-Level Assignments for TI 16-Bit Mode

TI Signal Name	Board-Level Assignment	Comments
DDR_EN	Pull down	Sets the TI PHY to operate in the SDR mode.
/RESET (pipe_rstn)	Pull down	Asynchronous reset signal to the TI PHY. This signal must be driven by the FPGA in the user logic and have a pull-down on the board. On power-up the PHY is reset while the FPGA is being configured. When the FPGA configuration is complete, the FPGA drives this signal high.
CLK_SEL	Connected to a real pin/signal on the board	Selects the reference clock frequency. Refer to the TI data sheet for more information.
P1_SLEEP	Pull down	Refer to the TI data sheet for more information.

Interfacing with the Philips PX1011A PHY

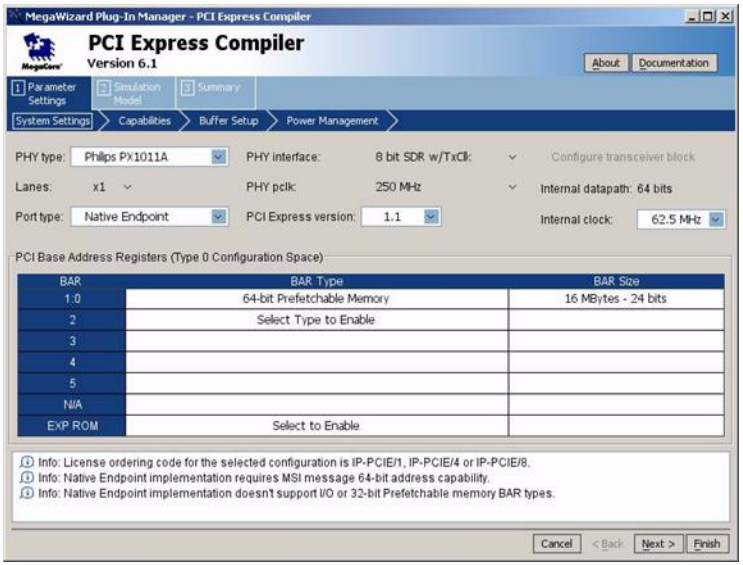
The PX1011A is a 2.5 Gbit/s PCI Express PHY with an 8-bit data PXPIPE interface. The PXPIPE interface is a superset of the PIPE interface, enhanced and adapted for off-chip applications with the introduction of a source synchronous clock for transmit and receive data. The 8-bit data interface operates at 250 MHz with SSTL_2 signaling.

For more information about interfacing with the Philips PX1011A PHY, refer to the product data sheet *PX1011A/PX1012A PCI Express stand-alone X1 PHY* (Rev. 02 — 18 May 2006), published by Philips Semiconductors.

Core Generation

In the **System Settings** tab of the PCI Express MegaWizard Plug-In Manager, set PHY type to **Philips PX1011A**. The PHY interface is automatically set to **8 bit SDR w/TxClk** as shown in Figure 10.

Figure 10. Philips 8-Bit SDR Mode Operation

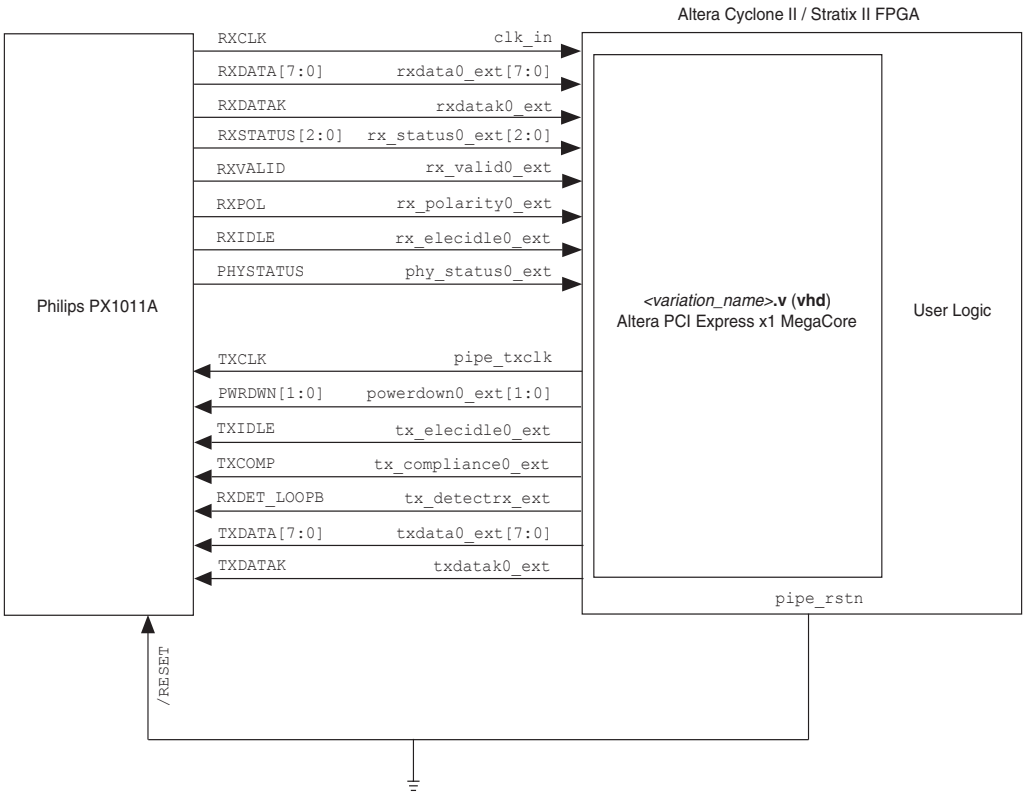


In this mode the PIPE interface operates at 250 MHz. Refer to the Philips PX1011A documentation to set the values for Power Management.

Additional Board Considerations

Figure 11 shows the mapping of the signals between the Philips PX1011A PHY and the Altera MegaCore function in the 8-bit SDR mode. This PHY does not require additional board-level assignments.

Figure 11. Mapping Signals in Philips 8-Bit SDR Mode



For more information about the layout guideline with the Philips PHY, refer to the application note *AN10373: PCI Express PHY PCB Layout Guideline*, published by Philips Semiconductors.

Document Revision History

Table 5 shows the revision history for this application note.

Table 5. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2007 v1.0	Initial release.	—

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support:
www.altera.com/support/
Literature Services:
literature@altera.com