Libero SoC v2021.2

RTG4™ High Speed Serial Interface Configuration User Guide

Introduction

This user guide covers the configurators for the following RTG4 high-speed serial interface cores:

- RTG4 High Speed Serial Interface EPCS and XAUI core
- RTG4 High Speed Serial Interface EPCS and XAUI core with initialization
- RTG4 High Speed Serial Interface PCIe, EPCS, and XAUI core
- RTG4 High Speed Serial Interface PCIe, EPCS, and XAUI core with initialization

Table of Contents

1. RTG4 High Speed Serial Interface Core Families

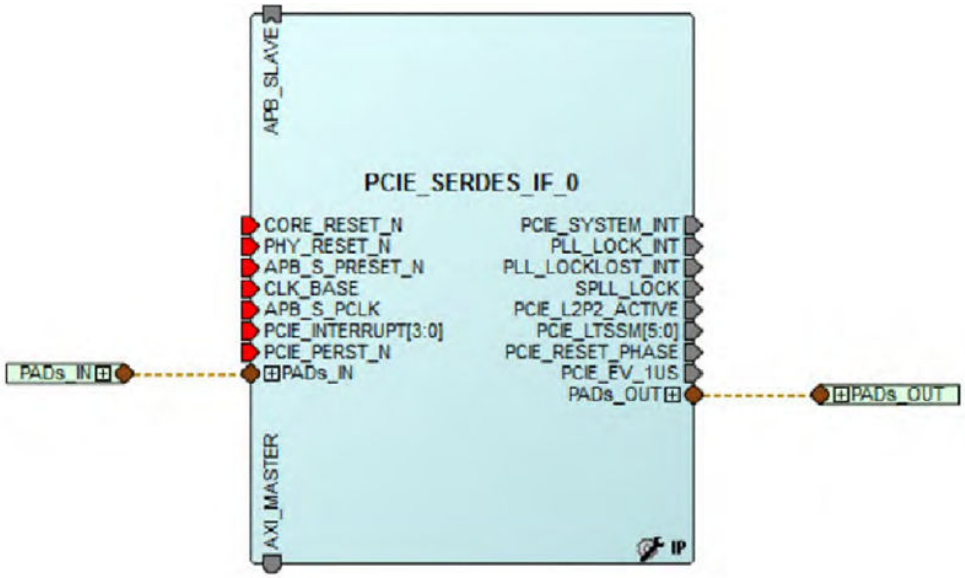
The following sections describe the RTG4 high-speed serial interface cores.

1.1 RTG4 High Speed Serial Interface PCIe, EPCS, and XAUI Core

The RTG4 High Speed Serial Interface PCIe, EPCS, and XAUI core (PCIE_SERDES_IF) provides multiple high-speed serial protocols, such as PCIe end-point, EPCS, and XAUI.

The device may contain one or more RTG4 high-speed serial interface blocks, depending on its size (see the RTG4 data sheet and product brief). For more information about the high-speed serial interface, see the RTG4 High Speed Serial Interfaces User's Guide.

Figure 1-1. PCIE_SERDES_IF Block Instantiation on the SmartDesign Canvas



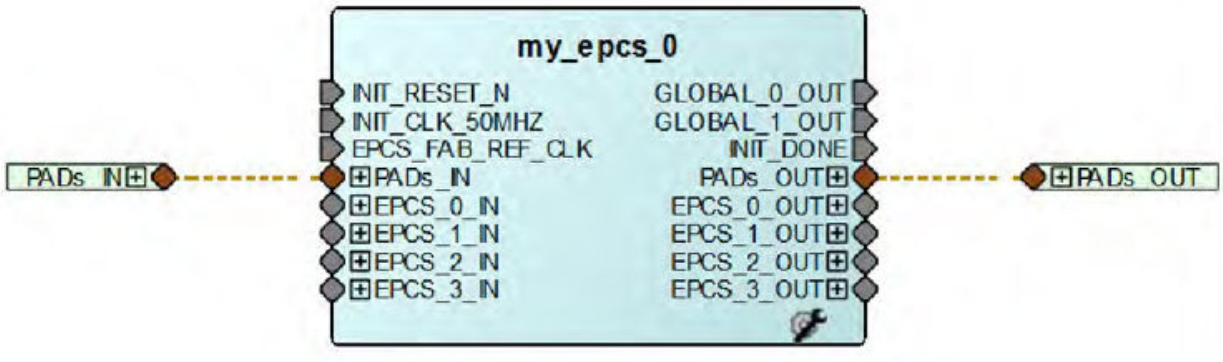
1.2 RTG4 High Speed Serial Interface EPCS and XAUI Core with Initialization

The RTG4 high-speed serial interface EPCS and XAUI core with initialization (my_epcs) supports the EPCS and XAUI protocols and includes built-in initialization circuitry. The NPSS prefix denotes a non-PCIe high-speed serial interface.

Note: This core does not support the PCIe protocol. For the PCIe protocol, use either the RTG4 high-speed serial interface PCIe, EPCS, and XAUI core or the RTG4 high-speed serial interface PCIe, EPCS, and XAUI with initialization core.

The device may contain one or more RTG4 high-speed serial interface blocks, depending on its size (see the RTG4 data sheet and product brief). For more information about the high-speed serial interface, see the RTG4 High Speed Serial Interfaces User's Guide.

Figure 1-2. my_epcs Block Instantiation on the SmartDesign Canvas



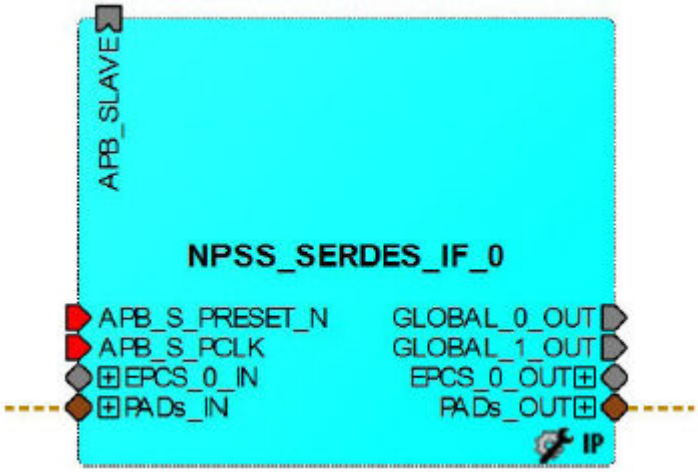
1.3 RTG4 High Speed Serial Interface EPCS and XAUI Core

The RTG4 High Speed Serial Interface EPCS and XAUI core (NPSS_SERDES_IF) supports the EPCS and XAUI protocols. The NPSS prefix denotes a non-PCIe High Speed Serial Interface.

Note: This core does not support the PCIe protocol. For the PCIe protocol, use either the RTG4 High Speed Serial Interface PCIe, EPCS, and XAUI core or the RTG4 High Speed Serial Interface PCIe, EPCS, and XAUI with initialization core.

The device may contain one or more RTG4 high-speed serial interface blocks, depending on its size (see the RTG4 data sheet and product brief). For more information about the high-speed serial interface, see the RTG4 High Speed Serial Interfaces User's Guide.

Figure 1-3. NPSS_SERDES_IF Block Instantiated on the SmartDesign Canvas

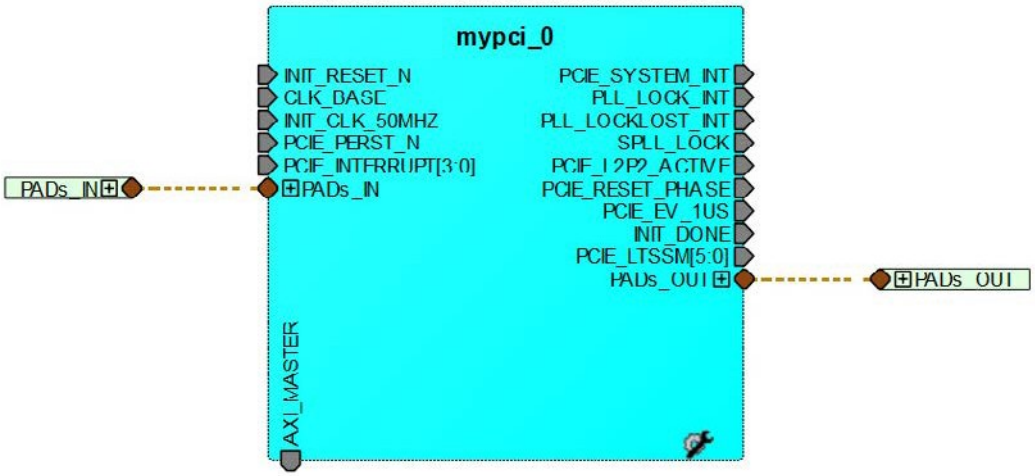


1.4 RTG4 High Speed Serial Interface PCIe, EPCS, and XAUI Core with Initialization

The RTG4 High Speed Serial Interface PCIe, EPCS, and XAUI core with initialization (mypci) provides multiple high-speed serial protocols, such as PCIe end-point, EPCS, and XAUI and includes built-in initialization circuitry.

The device may contain one or more RTG4 high-speed serial interface blocks, depending on its size (see the RTG4 data sheet and product brief). For more information about the high-speed serial interface, see the RTG4 High Speed Serial Interfaces User's Guide.

Figure 1-4. mypci Block Instantiation on the SmartDesign Canvas

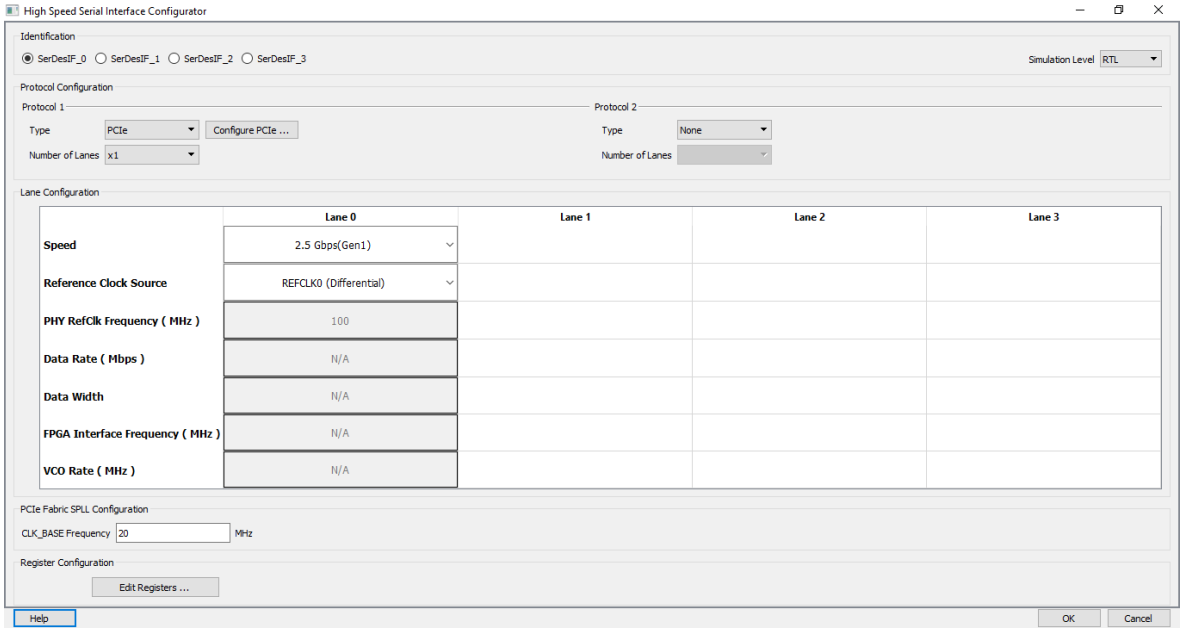


2. Accessing the Core Configurator

To access the Core Configurator:

1. Drag and drop the core from the Catalog into the SmartDesign Canvas.
2. When prompted, enter a component name.
3. Configure the component in the Configurator (see the following figure). As you make selections, the Configurator narrows down the choices and defaults, so that only the relevant ports appear in the generated macro.

Figure 2-1. RTG4 High Speed Serial Interface Configurator



3. Common Configuration Settings

The following topics describe the configuration settings applicable to all RTG4 High Speed Serial Interface cores covered in this guide.

3.1 Identification

RTG4 devices contain multiple high-speed serial interface blocks. Block names depend on the number of blocks in the device. For a list of resources available on a device, see the RTG4 device data sheet.

The first row of check boxes allows you to select which high-speed serial interface block you want to configure. The following table shows the blocks that can be selected for each core.
Note: The SERDES block names depend on the number of blocks present in the device. For more information about the number of blocks present in the device, see the device datasheet.

Table 3-1. High-Speed Serial Interface Blocks For Each Core

Core	Blocks That Can be Selected
EPCS and XAUI core	SERDES_1, SERDES_2, SERDES_3, and SERDES_4
EPCS and XAUI core with initialization	SERDES_1,SERDES_2, SERDES_3, and SERDES_4
PCIe, EPCS, and XAUI core	SERDES_PCIE_0 and SERDES_PCIE_5
PCIe, EPCS, and XAUI core with initialization	SERDES_PCIE_0 and SERDES_PCIE_5

3.2 Protocol Configuration

The RTG4 high-speed serial interface blocks support two protocols:

- Protocol 1
- Protocol 2

Note: Configure Protocol 1 before configuring Protocol 2. For each Protocol, you must configure the Type and Number of Lanes.

3.2.1 Protocol 1

Select your protocol type from the drop-down menu. The available choices depend on the core.

Table 3-2. Matching Cores with Protocol 1 Selections

Core	Available Protocol 1 Selections
EPCS and XAUI	XAUI and EPCS
EPCS and XAUI with initialization	XAUI and EPCS
PCIe, EPCS, and XAUI	PCIe, PCIe (Reverse), XAUI, and EPCS
PCIe, EPCS, and XAUI with initialization	PCIe, PCIe (Reverse), XAUI, and EPCS

3.2.2 Protocol 2

Select the protocol type from the drop-down menu. The selections for Protocol 2 are context sensitive and depend on the option you selected for Protocol 1.

Note: Protocol 2 type is disabled, if you have selected XAUI for Protocol 1.

3.2.3 Supported Protocol 1 and 2 Combinations

The following tables list Protocol 1 and Protocol 2 combinations supported by the single high-speed serial interface blocks described in this guide.

3.2.3.1 Supported Protocol Combinations for EPCS and XAUI

The following table lists Protocol 1 and Protocol 2 combinations supported by a single EPCS and XAUI block that supports or does not support initialization.

Table 3-3. Supported Protocol 1 and 2 Combinations for EPCS and XAUI

Protocol Type	Protocol #	Lane Width	Lane Assignment	Description	Speed Choices
XAUI	Protocol 1	x4	Lane 0, Lane 1, Lane 2, Lane 3		3.125 Gpbs
EPCS	Protocol 1	x1	Lane 0, Lane 1, Lane 2, Lane 3	Lane 2 or 3 can be selected when Protocol 2 is not used.	Custom Speed
		X2	Lane 0, Lane 1		
		x4	Lane 0, Lane 1, Lane 2, Lane 3		
	Protocol 2	x1	Lane 2, Lane 3	Not available when Protocol 1 is XAUI.	
		x2	Lane 2, Lane 3		

3.2.3.2 Supported Protocol Combinations for PCIe, EPCS, and XAUI

The following table lists the Protocol 1 and Protocol 2 combinations supported by a single PCIe, EPCS, and XAUI block that supports or does not support initialization.

Note: If you select the PCIe or PCIe Reverse protocol, use the **Configure PCIe** button to configure PCIe-Specific Configuration Settings. If you select the XAUI protocol, all four lanes are selected by default.

Table 3-4. Supported Protocol 1 and 2 Combinations for PCIe, EPCS, and XAUI

Protocol Type	Protocol #	Lane Width	Lane Assignment	Description	Speed Choices
PCIe	Protocol 1	x1	Lane 0	—	Gen1 (2.5 Gbps), Gen2 (5.0 Gbps)
		x2	Lane 0, Lane 1		
		x4	Lane 0, Lane 1, Lane 2, Lane 3		
PCIe Reverse	Protocol 1	x1	Lane 3	—	Gen1 (2.5 Gbps), Gen2 (2.5 Gbps)
		x2	Lane 2, Lane 3, or Lane 0, Lane 1 if Protocol 2 is used		
		x4	Lane 0, Lane 1, Lane 2, Lane 3		
XAUI	Protocol 1	x4	Lane 0, Lane 1, Lane 2, Lane 3	—	3.125 Gpbs

.....continued					
Protocol Type	**Protocol #**	**Lane Width**	**Lane Assignment**	**Description**	**Speed Choices**
EPCS	Protocol 1	x1	Lane 0, Lane 1, Lane 2, Lane 3	Select Lane 0, 1, 2, or 3, or Select Lane 2 or 3 when Protocol 2 is not used.	Custom Speed
		x2	Lane 0, Lane 1		
		x4	Lane 0, Lane 1, Lane 2, Lane 3		
	Protocol 2	x1	Lane 2, Lane 3	Available only when Protocol 1 is EPCS.	
		x2	Lane 2, Lane 3		

3.2.4 Lane Configuration

Use Lane Configuration to configure up to four lanes for your block. PCIe blocks can be configured to run in dual-protocol mode.

Figure 3-1. High Speed Serial Interface Configurator for Non-PCIe Blocks

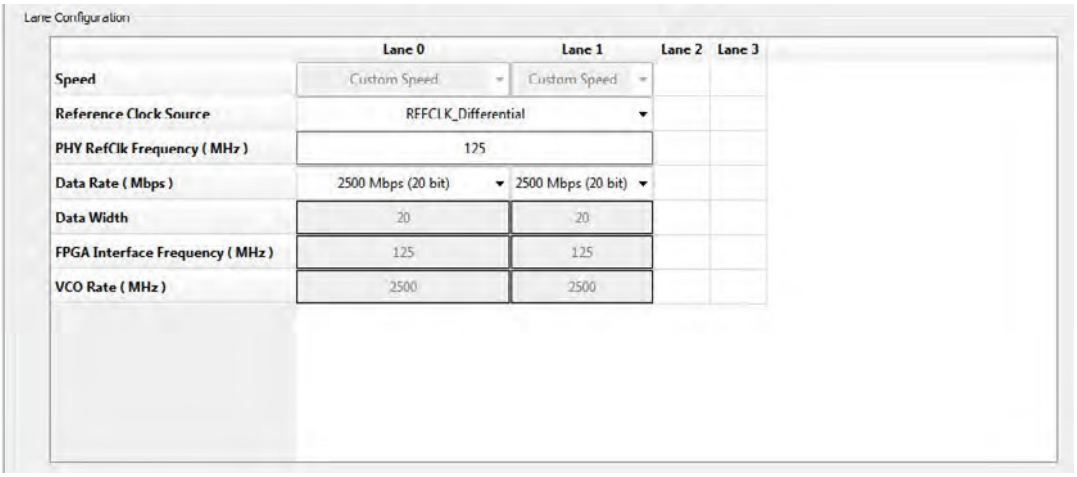


Figure 3-2. High Speed Serial Interface Configurator for PCIe Blocks

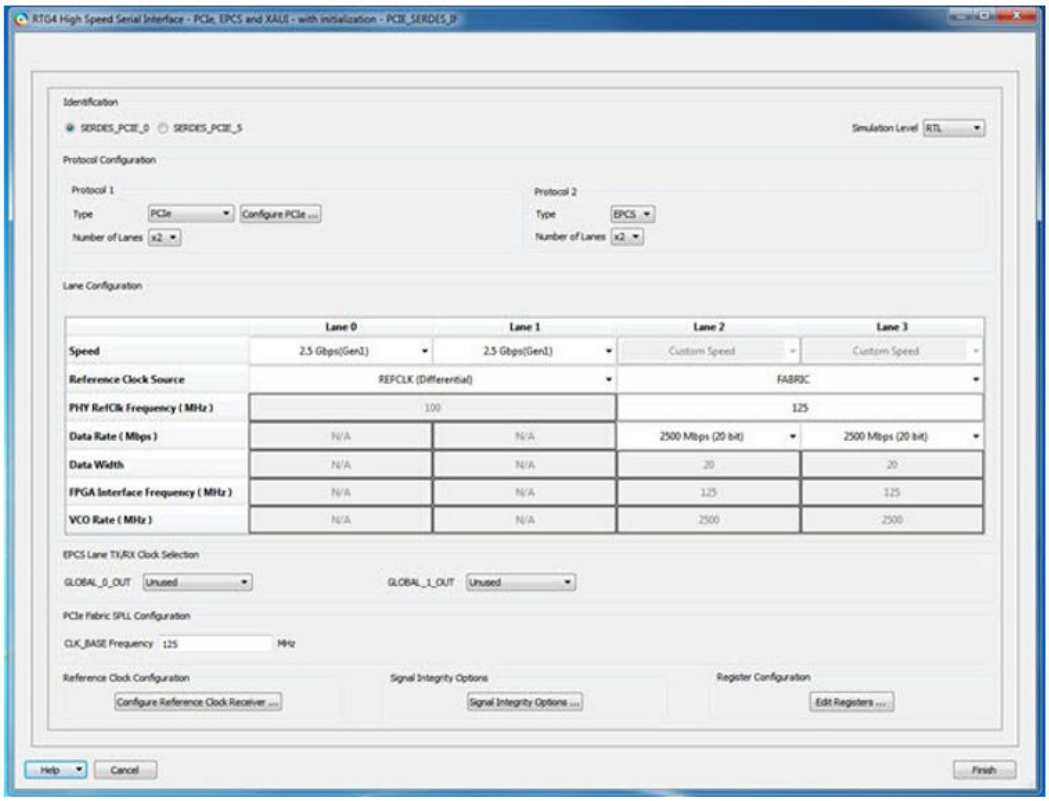


Table 3-5. Lane Configuration Settings

Parameter	Description
Speed	Available selections depend on the selected protocol. See 3.2.3.1. Supported Protocol Combinations for EPCS and XAUI and 3.2.3.2. Supported Protocol Combinations for PCIe, EPCS, and XAUI.
Reference Clock Source	Clock sources can be differential or single-ended. For PCIe blocks, two clock sources are available: REFCLK0 and REFCLK1. Select one of the following options for Protocol 1 and Protocol 2: • REFCLK (Differential) • REFCLK0 (Voltage_Referenced) • REFCLK1 (Voltage_Referenced) • REFCLK0 (Single_Ended) • REFCLK1 (Single_Ended) • Fabric (EPCS protocol only) **Note:** Lane 0 and Lane 1 share the same Reference Clock, and Lane 2 and Lane 3 share the same Reference Clock. The selected Reference Clock is always available as REFCLK0_OUT or REFCLK1_OUT, and can be used as clock source for logic inside Fabric.

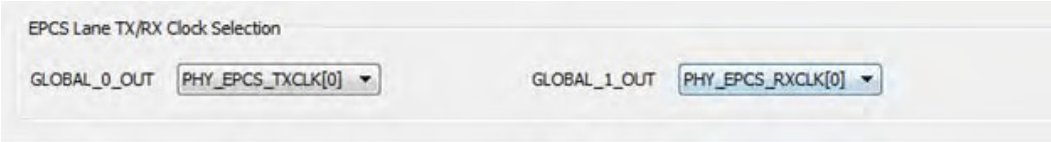
.....continued	
Parameter	**Description**
PHYRefClk Frequency (MHz)	Read-only fixed value for all protocols, except: • EPCS Custom Speed: enter values between 100 and 160 MHz. • For XAUI, the frequency is fixed at 156.25 MHz.
Data Rate (Mbps)	Read-only fixed value for all protocols, except: • EPCS Custom Speed: select a data rate from the drop-down menu. Data rates are computed based on the PHY RefClk frequency.
Data Width	Read-only fixed value for all protocols, except EPCS Custom Speed. For EPCS, the data width varies with Data Rate (Mbps) and PHY RefClk Frequency (MHz) as follows: • 20 bit (for 3125 Mbps and 156.25 MHz, or 2500 Mbps and 125 MHz, or 2000 Mbps and 100 MHz) • 16 bit (for 2500 Mbps and 156.25 MHz, or 2000 Mbps and 125 MHz, or 1600 Mbps and 100 MHz) • 10 bit (2500 Mbps and 125 MHz, 1250 Mbps and 125 MHz, or 2000 Mbps and 100 MHz, or 1000 Mbps and 100 MHz) • 8 bit (for 2000 Mbps and 125 MHz, or 1600 Mbps and 100 MHz, or 1000 Mbps and 125 MHz) • 5 bit (for 1250 Mbps and 125 MHz, or 1000 Mbps and 100 MHz) • 4 bit (for 1000 Mbps and 125 MHz) The displayed value is computed and updated based on the selected PHY RefClk frequency and data rate.
FPGA Interface Frequency (MHz)	Read-only fixed value for all protocols, except EPCS Custom Speed. The displayed value is computed and updated based on the selected PHY RefClk frequency and data rate.
VCO Rate (MHz)	Read-only fixed value for all protocols, except EPCS Custom Speed. The displayed value is computed and updated based on the selected PHY RefClk frequency and data rate.

3.3 Clock Configuration

3.3.1 EPCS Lane TX/RX Clock Selection

The EPCS Lane TX/RX Clock Selection option can be configured when the selected protocol is EPCS. This option allows you to select the clocks to use for the TX/RX fabric interface and the logic in Fabric.

Figure 3-3. EPCS Lane TX/RX Clock Selection



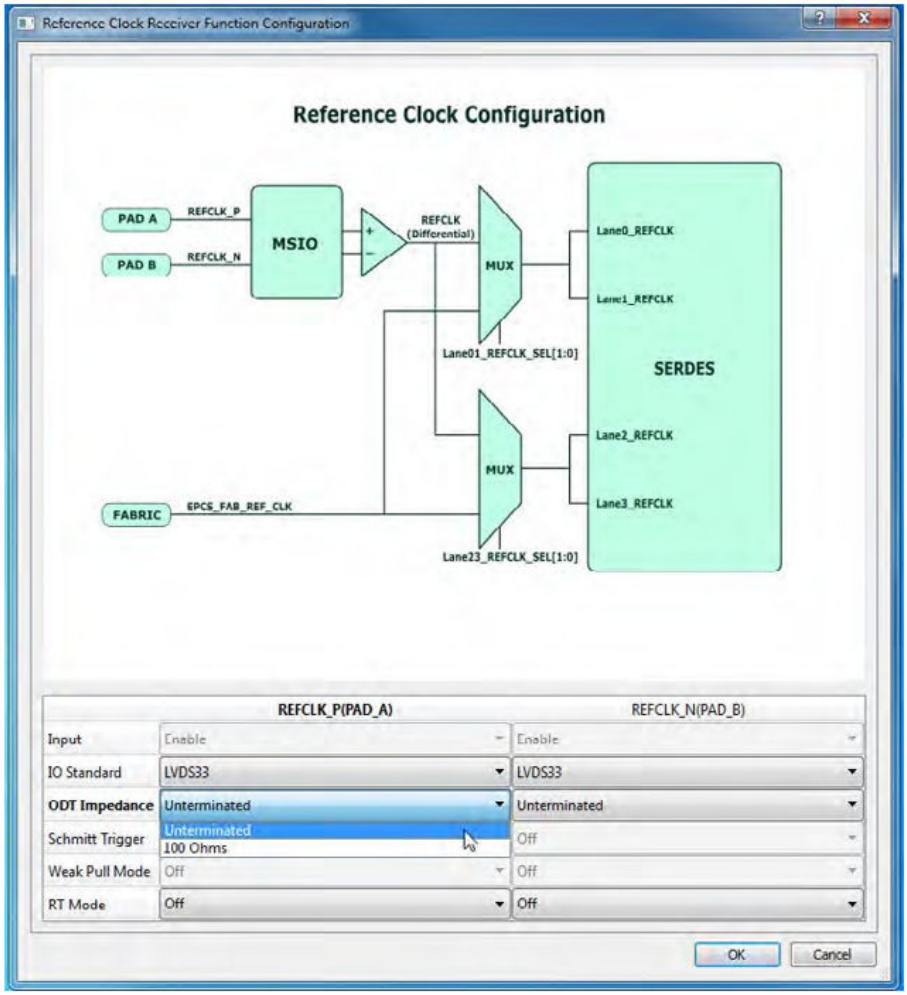
3.3.2 Reference Clock Configuration

The Reference Clock Configuration option allows you to configure the Multi-Standard User I/O (MSIO) Reference Clock. To access this option, click the **Configure Reference Clock Receiver** button to open the Configuration dialog box.

Based on the selected PHY Reference Clock source, supported I/O standards, Impedance, Receiver, Weak Pull-Up/Pull-Down, and other settings may be selected. For example, if you select REFCLK_Differential as the PHY Reference Clock source selection, only the supported differential I/O standards are displayed.

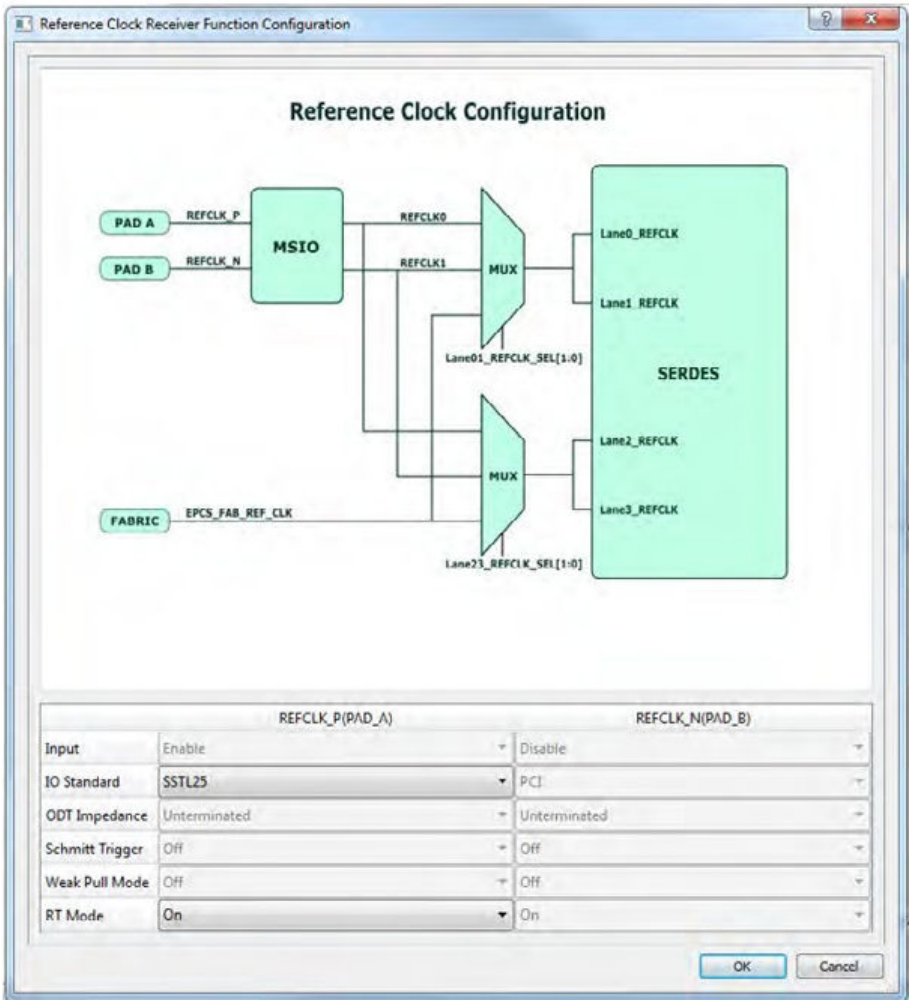
The following figure shows how the REFCLK selection is used for each channel when REFCLK Clock Source is REFCLK_Differential.

Figure 3-4. REFCLK Selection for Each Channel when REFCLK Clock Source is REFCLK_Differential



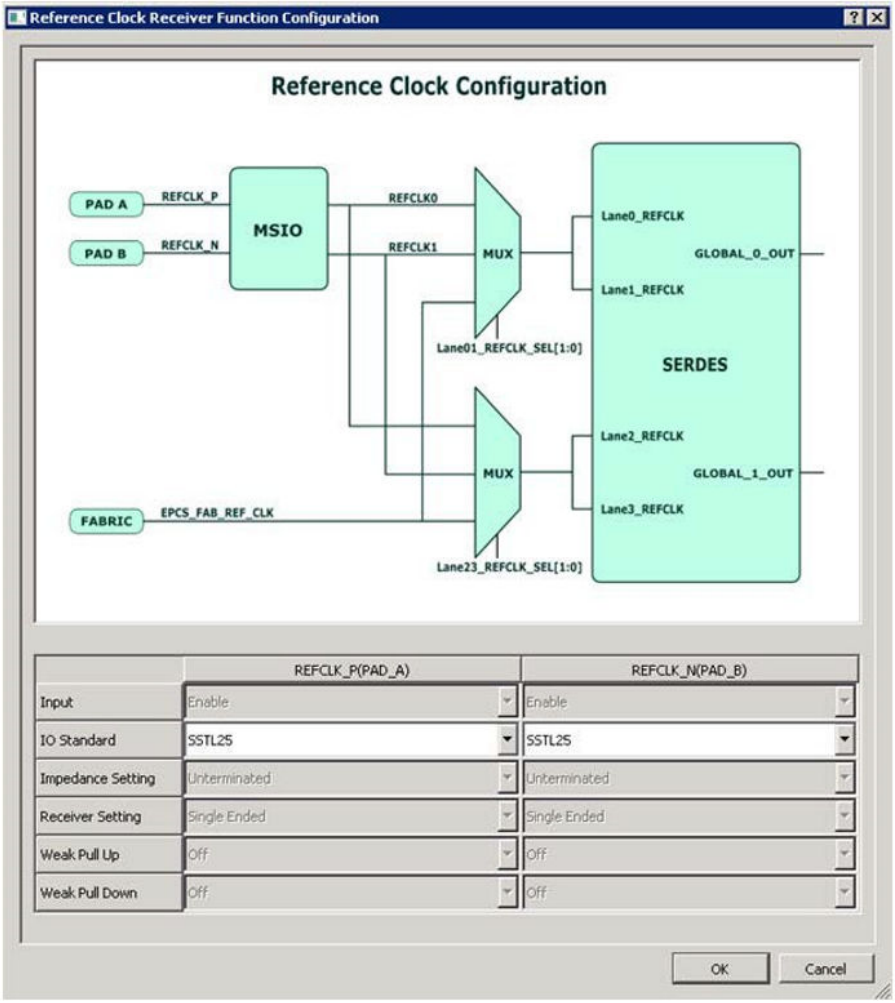
The following figure shows how REFCLK selection is used for each channel when REFCLK Clock Source is REFCLK0_Voltage or REFCLK1_Voltage.

Figure 3-5. REFCLK Selection for Each Channel when REFCLK Clock Source is REFCLK0_Voltage_referenced or REFCLK1_Voltage_referenced



The following figure shows how the REFCLK selection is used for each channel when REFCLK Clock Source is REFCLK_P_Voltage or REFCLK_N_Voltage.

Figure 3-6. REFCLK Selection for Each Channel when REFCLK Clock Source is REFCLK_P_Voltage_referenced or REFCLK_N_Voltage_referenced



3.4 I/O Standards

The following table lists the supported I/O standards.

Table 3-6. Supported I/O Standards

Supported I/O Standards	Available Values
Differential I/O standards	• LVDS33 • LVDS25 • RSDS • MiniLVDS • LVPECL33
Voltage-referenced I/O standards	• SSTL25 • SSTL18 • HSTL18

.....continued	
Supported I/O Standards	**Available Values**
Single-ended I/O standards	• LVTTL33 • LVCMOS33 • LVCMOS25 • LVCMOS18

Observe the following guidelines:

- The voltage for all block I/Os must be the same.
- For cores that do not support initialization, the I/O Standard voltage must be the same for both REFCLK_P(PAD_A) and REFCLK_N(PAD_B). If you change an I/O Standard to REFCLK_P(PAD_A), the Configurator makes the same change to REFCLK_N(PAD_B), and vice versa.
- For cores that support initialization, HCSL inputs to the RTG4 REFCLK inputs are supported directly with the LVDS25 I/O STD selected in Libero®. There is no specific HCSL I/O STD available in Libero, and designs requiring HCSL reference clocks are supported by using the LVDS25 I/O standard. For more information, see the RTG4 FPGA Data Sheet and RTG4 FPGA High Speed Serial Interfaces User Guide.

3.4.1 ODT Impedance Settings

The On Die Termination (ODT) impedance settings are:

- Unterminated
- 50 Ohms
- 100 Ohms

Note: For blocks that do not support initialization, the LVDS33 I/O standard does not support ODT.

3.4.2 Schmitt Trigger

The following Schmitt Trigger options are supported:

- On
- Off

3.4.3 Receiver Setting

The supported Receiver Setting options are:

- Single Ended
- Schmitt Trigger

3.4.4 Weak Pull-Up/Pull-Down

The supported Weak Pull-Up/Pull-Down options are:

- OFF
- ON

3.5 Signal Integrity Options

The Signal Integrity dialog box allows you to maintain signal integrity and mitigate signal integrity problems.

To open the dialog box, click **Signal Integrity Options**. The values you enter are used to set register values related to signal integrity. Lanes that are not used are grayed-out in the dialog box.

Figure 3-7. Signal Integrity Options Button

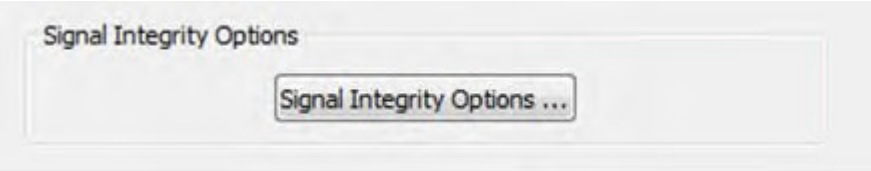
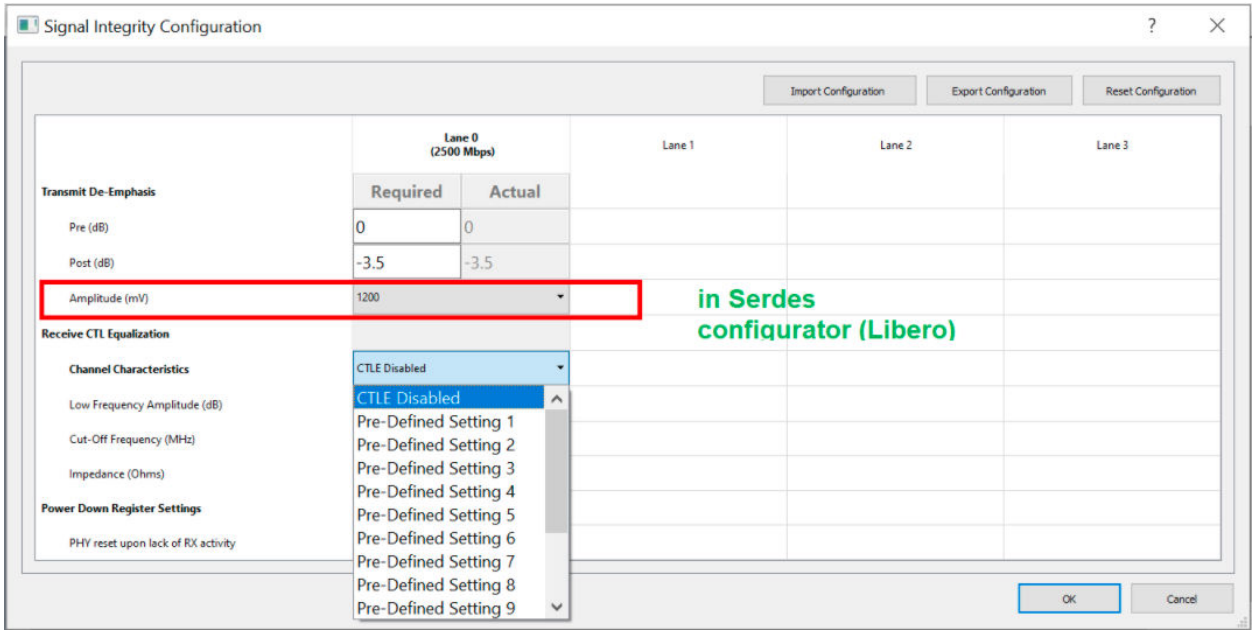


Figure 3-8. Signal Integrity Configuration Dialog Box



3.5.1 Transmit De-Emphasis

Enter any values between 0.1 and 36.1 (in dB) in the **Required** edit box for both Pre-Transmit and Post-Transmit stages. See the following table for Actual values supported.

The value you enter in the **Required** box is matched to the closest valid Actual value and displayed in the **Actual** box. The Configurator sets appropriate values for LANE<n>_TX_PRE_RATIO and LANE<n>_TX_PST_RATIO registers based on the Actual value.

The lane<n>_TX_AMP_RATIO lane register is always set to 0x80 for all lanes.

LANE<n>_TX_PRE_RATIO and LANE<n>_TX_PST_RATIO registers are set based on the Actual value, as shown in the following table. The default value for **Required** is 0 for **Pre-Transmit** and 3.5 for **Post-Transmit**.

In the following table, LANE<n> denotes the lane number, where <n> can be 0, 1, 2, or 3. For example, if you enter 2.4 dB in the **Required** box, the value **2.5 dB** (the closest match) appears in the **Actual** box and the registers are set as follows:

- LANE<n>_TX_PRE_RATIO registers are set to 0x10.
- LANE<n>_TX_PST_RATIO registers are set to 0x10.
- LANE<n>_TX_AMP_RATIO registers are set to 0x80 for all lanes.

Table 3-7. EPCS Configuration for Different Data Widths

Feature	Control Registers	Actual Value = Value Programmed in Register
De-Emphasis Pre	LANE<n>_TX_PRE_RATIO	0.1 dB = 0x1 0.3 dB = 0x2 0.4 dB = 0x3
De-Emphasis Post	LANE<n>_TX_PST_RATIO	0.5 dB = 0x4 0.7 dB = 0x5 0.9 dB = 0x6 1 dB = 0x7

.....continued		
Feature	**Control Registers**	**Actual Value = Value Programmed in Register**
		1.2 dB = 0x8
		1.3 dB = 0x9
		1.5 dB = 0xa
		1.6 dB = 0xb
		1.8 dB = 0xc
		2 dB = 0xd
		2.1 dB = 0xe
		2.3 dB = 0xf
		2.5 dB = 0x10
		2.7 dB = 0x11
		2.9 dB = 0x12
		3 dB = 0x13
		3.3 dB = 0x14
		3.5 dB = 0x15
		3.7 dB = 0x16
		3.9 dB = 0x17
		4 dB = 0x18
		4.3 dB = 0x19
		4.5 dB = 0x1a
		4.8 dB = 0x1b
		5 dB = 0x1c
		5.2 dB = 0x1d
		5.5 dB = 0x1e
		5.8 dB = 0x1f
		6 dB = 0x20
		6.3 dB = 0x21
		6.5 dB = 0x22
		7 dB = 0x23
		7.2 dB = 0x24
		7.5 dB = 0x25

.....continued		
Feature	**Control Registers**	**Actual Value = Value Programmed in Register**
		7.8 dB = 0x26
		8 dB = 0x27
		8.5 dB = 0x28
		9 dB = 0x29
		9.3 dB = 0x2a
		9.7 dB = 0x2b
		10.1 dB = 0x2c
		10.5 dB = 0x2d
		11 dB = 0x2e
		11.5 dB = 0x2f
		12 dB = 0x30
		12.6 dB = 0x31
		13.2 dB = 0x32
		13.8 dB = 0x33
		14.5 dB = 0x34
		15.2 dB = 0x35
		16.1 dB = 0x36
		17 dB = 0x37
		18 dB = 0x38
		19.2 dB = 0x39
		20.5 dB = 0x3a
		22.1 dB = 0x3b
		24 dB = 0x3c
		26.5 dB = 0x3d
		30.1 dB = 0x3e
		36.1 dB = 0x3f

3.5.2 Receive CTL Equalization

There are 14 predefined settings from which you can select in addition to the default setting of **CTLE Disabled**.

Each predefined setting has the corresponding Cut-Off Frequency and Low Frequency Amplitude values preset and displayed when the predefined setting is selected. The impedance value for all predefined settings is 100 Ohms. Depending on the predefined setting, lane registers LANE<n>_RE_AMP_RATIO and LANE<n>_RE_CUT_RATIO are set to the values listed in the following table.

Table 3-8. Predefined Receive CTL Equalization Settings

Predefined Setting #	Cut-Off Frequency (MHz)	Low Frequency Amplitude (dB)	LANE<n>_RE_AMP_ RATIO (HEX)	LANE<n>_RE_CUT_ RATIO (HEX)
CTLE Disabled	N/A	N/A	0x00	0x00
1	400	10.88	0x77	0x20
2	500	13.06	0x5B	0x2A
3	600	13.98	0x51	0x2F
4	700	12.04	0x4A	0x32
5	800	13.38	0x3E	0x3C
6	900	13.98	0x39	0x40
7	1000	12.57	0x70	0x4F
8	1100	11.6	0x7D	0x58
9	1200	10.88	0x37	0xFE
10	1300	9.95	0x22	0xFC
11	1400	8.52	0x52	0xFE
12	1500	7.47	0x5B	0xFE
13	1600	7.04	0x73	0xFE
14	1700	6.66	0x78	0xFE

Note: In the preceding table, LANE<n> denotes the lane number, where <n> can be 0, 1, 2 or 3.

3.5.3 Power Down Register Settings

Power Down Register settings control the Physical Reset behavior of the EPCS SERDES when there is a lack of RX activity. The selections are:

- Enabled — Physical Reset behavior is enabled.
- Disabled — Physical Reset behavior is disabled.

3.6 High-Speed Serial Interface Control Registers

The high-speed serial interface has a set of registers that can be configured at runtime. The configuration values for these registers represent different parameters. Most of these registers are populated automatically by the selections made in the configuration GUI. However, in some advanced configurations, you may have to modify these registers.

To enter the high-speed serial interface configuration values, specify the register values when you configure the high-speed serial interface. Click **Edit Registers** in the high-speed serial interface configurator to display the Registers Configuration dialog box. This dialog box allows you to enter high-speed serial interface register values using a graphical interface. At power up, the values you enter are written to the high-speed serial interface registers.

Alternatively, you can click **Import Configuration** and import a configuration text file to configure the registers.

Figure 3-9. Edit Registers Button

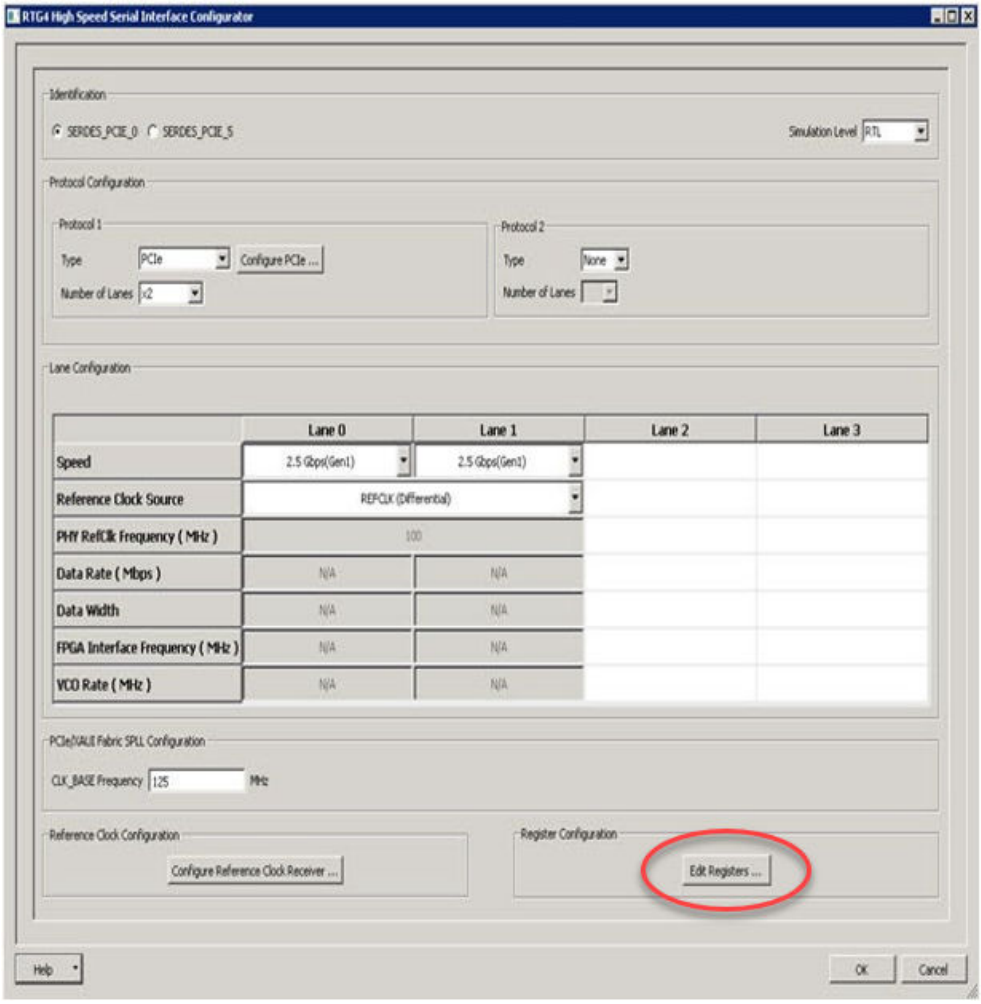
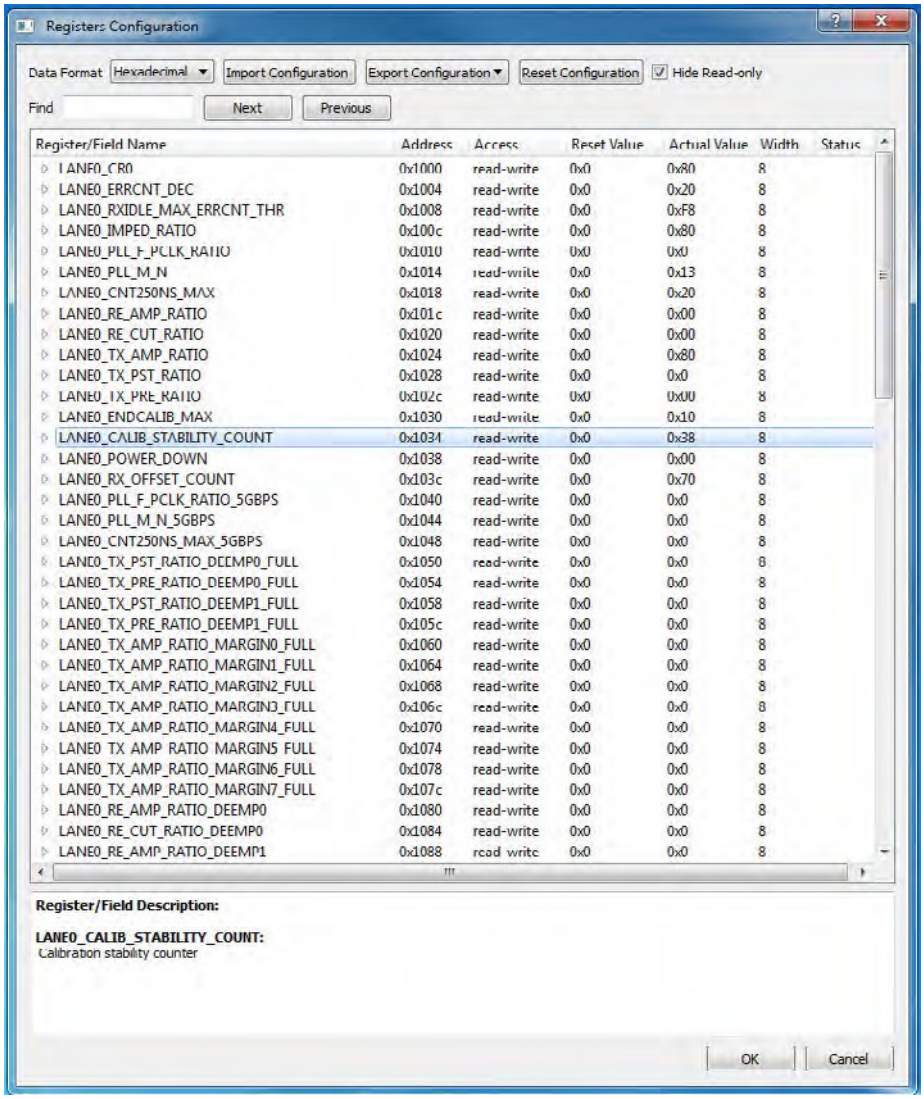


Figure 3-10. Registers Configuration Dialog Box



The Registers Configuration dialog box has the following features:

- Registers Table — allows you to enter individual register values in the Registers Table. To enter a register value, expand the register data tree using the arrow or + sign, and then click the **Actual Value** column to edit.
- Import Configuration — imports complete register configurations from text files. Register configuration syntax is shown in the figure above; Microchip recommends using this method.
- Export Configuration — exports the current register configuration data into a text file. The syntax of the exported file is the same as that of importable register configuration text files.

Table 3-9. Non-PCIe Example

Register/Field Name	Address
LANE0_CR0	0x80
LANE0_ERRCNT_DEC	0x20
LANE0_RXIDLE_MAX_ERRCNT_THR	0xF8
LANE0_IMPED_RATIO	0x80
LANE0_PLL_F_PCLK_RATIO	0x0

.....continued	
Register/Field Name	**Address**
LANE0_PLL_M_N	0x13
LANE0_CNT250NS_MAX	0x20
LANE0_RE_AMP_RATIO	0x00
LANE0_RE_CUT_RATIO	0x00
LANE0_TX_AMP_RATIO	0x80
LANE0_TX_PST_RATIO	0x0
LANE0_TX_PRE_RATIO	0x00
LANE0_ENDCALIB_MAX	0x10

Table 3-10. PCIe Example

Register/Field Name	Address
PCIE_AXI_MASTER_WINDOW0_0	0x00000000
PCIE_AXI_MASTER_WINDOW0_1	0xfffff001
PCIE_AXI_MASTER_WINDOW0_2	0x00000000
PCIE_AXI_MASTER_WINDOW0_3	0x00000000
PCIE_AXI_MASTER_WINDOW1_0	0x00001000
PCIE_AXI_MASTER_WINDOW1_1	0xfffff001
PCIE_AXI_MASTER_WINDOW1_2	0x4

- Reset Configuration — click to undo changes made to the register configuration. Undoing changes deletes all register configuration data and returns to the hardware reset values. You must then re-enter or re-import data.
- Hide Read-Only Registers — shows or hides the read-only registers in the Register Table. These registers are primarily status registers and do not contribute to the configuration.

When you generate your FPGA, the configuration register data entered in this configurator is used to initialize the high-speed serial interface simulation model when performing a BFM simulation.

4. PCIe-Specific Configuration Settings

The configuration settings described in this chapter apply to the following blocks:

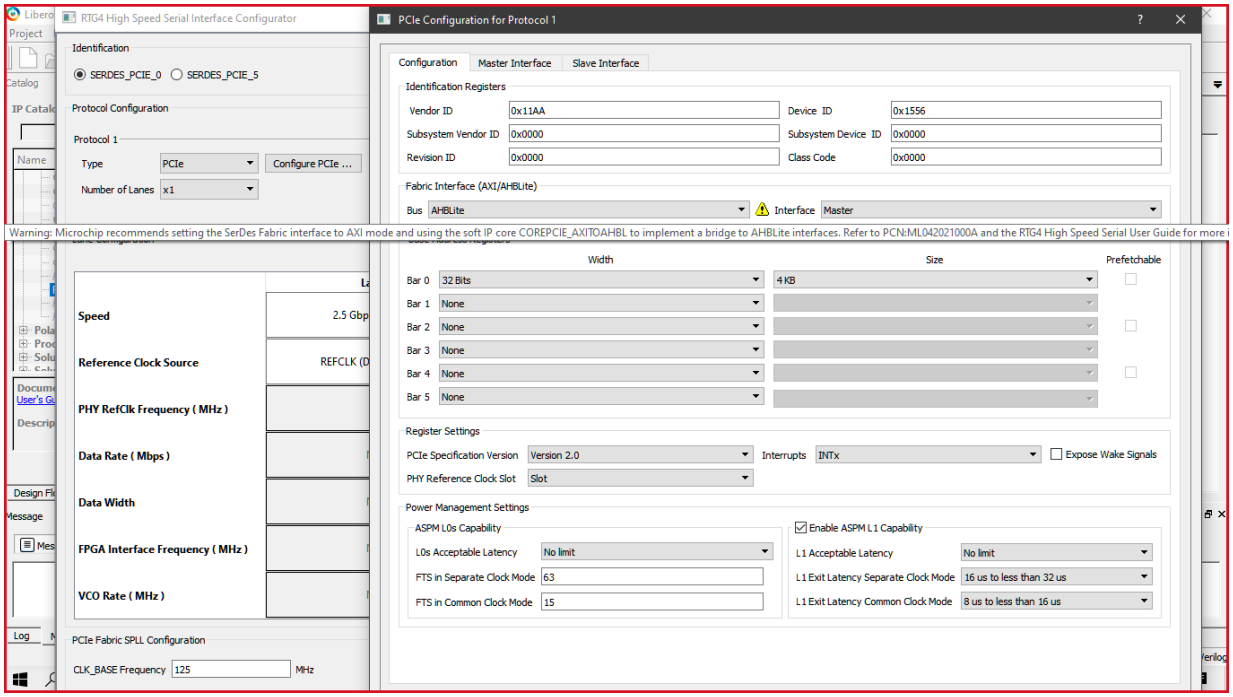
- RTG4 High Speed Serial Interface PCIe, EPCS, and XAUI core
- RTG4 High Speed Serial Interface PCIe, EPCS, and XAUI core with initialization

4.1 Configuration

The Configuration tab allows you to configure Identification Registers, Fabric Interface, Base Address Registers, and Options.

In the G4 /RTG4 PCIe SERDES Configurator, a warning icon is added and the following warning message is displayed as a Tooltip.

Figure 4-1. PCIe Configuration-Configuration Tab



The following ToolTip is displayed for RTG4:

Note: Microchip recommends setting the SERDES Fabric interface to AXI mode and using the soft IP core COREPCIE_AXITOAHBL to implement a bridge to AHBLite interfaces. For more information, see PCN_ML042021000A.

The following table lists the conditions when the warning is displayed.

Table 4-1. Show\Hide Warning Symbol

Bus	Interfce	Description
AXI bus	Master	No Warning
AXI bus	Slave	No Warning
AXI bus	Both	No Warning
AHBLite	Master	Warning
AHBLite	Slave	No Warning

.....continued		
Bus	**Interfce**	**Description**
AHBLite	Both	Warning

4.1.1 Identification Registers

For PCIe blocks, you can assign 16-bit hexadecimal signatures to the following six identification registers.

Table 4-2. Identification Registers Parameters

Parameter	Description
Vendor ID	Vendor ID that PCI-SIG assigns to Microchip is 0x11AA. To allocate subsystems under the Microchip vendor ID, contact Microchip.
Subsystem Vendor ID	Card manufacturer's ID.
Device ID	Manufacturer's assigned part number assigned by the vendor.
Revision ID	Revision number, if available.
Subsystem Device ID	Assigned by the subsystem vendor.
Class Code	PCIe device's generic function.

4.1.2 Fabric Interface

Use the Fabric Interface field to configure the Bus Standard (AXI or AHBLite) and Interface (Master Only, Slave Only, or both) for the PCIe protocol.

In PCIe mode, the SERDES block can act as an AXI or AHBLite Master.

Microchip recommends you to instantiate a COREAXI or CoreAHBLite Bus into the SmartDesign Canvas, and then connect the SERDES Master and/or Slave Bus Interface (BIF) to the Master and/or Slave BIF of the COREAXI or COREAHBLite bus.

4.1.3 Base Address Registers

The following table describes the fields of the six base address registers (Bar 0 through Bar 5).

Table 4-3. Base Address Registers Parameters

Parameter	Description
Width	The width on even registers can be 32-bit or 64-bit. If you set an even register to 64 bits wide, then the subsequent (odd) register serves as the upper half of 64 bits. Otherwise, the width of odd registers is restricted to 32 bits.
Size	Ranges from 4 Kbytes to 2 GB. Some devices might go up to 1 GB. See the device data sheet (IGLOO2 FPGA and SmartFusion2 Soc FPGA Datasheet).
Prefetchable	Enabled only on even registers that have a 64-bit width.

4.1.4 Options

The following table describes the Options parameters.

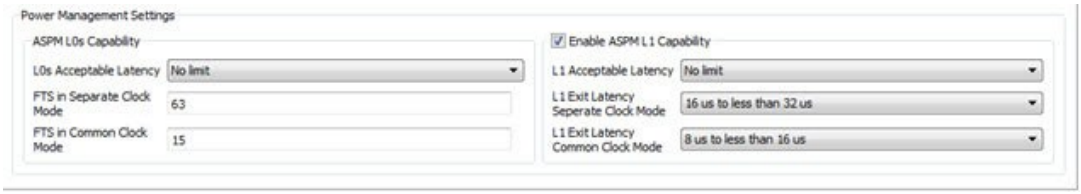
Table 4-4. Options Parameters

Parameter	Description
PHY Reference Clock Slot	Sets your reference clock to Slot or Independent.
L2/P2	Click the check boxes to add PCIE_WAKE_N, PCIE_WAKE_REQ, and PCIE_PERST_N ports.
PCIe Specification Version	Sets the Specification Version to 1.0, 1.1, or 2.0.
Interrupt	Sets the interrupt to one of the following values: • MSI 1 • MSI 2 • MSI 4 • MSI 8 • MSI 16 • MSI 32 • INTx Your Interrupt selection sets bit [16] of the PCIE_MSI_CTRL_STATUS register and bits [19:17] of the PCIE_MSI_CTRL_STATUS register.

4.2 PCIe Power Management Settings

Use the Power Management Settings to configure the Active State Power Management (ASPM) settings. The Configurator sets the appropriate register values for your SERDES block based on your selections.

Figure 4-2. Power Management Settings



4.2.1 ASPM L0s Capability

The following parameters are required.

Table 4-5. ASPM L0s Capability Parameters

Parameter	Description
L0s Acceptable Latency	Select a latency value: • Maximum of 64 ns • Maximum of 128 ns • Maximum of 256 ns • Maximum of 512 ns • Maximum of 1 µs • Maximum of 2 µs • Maximum of 4 µs • No Limit

.....continued	
Parameter	**Description**
FTS in Separate Clock Mode	Enter the number of Fast Training Sequences (FTS) required in separate clock mode. Range: 0 - 255.
FTS in Common Clock Mode	Enter the number of FTS required in common clock mode. Range: 0 - 255.

4.2.2 ASPM L1 Capability

By default, the ASPM L1 Capability is enabled. Configure the settings for ASPM L1 as follows.

Table 4-6. ASPM L1 Capability Parameters

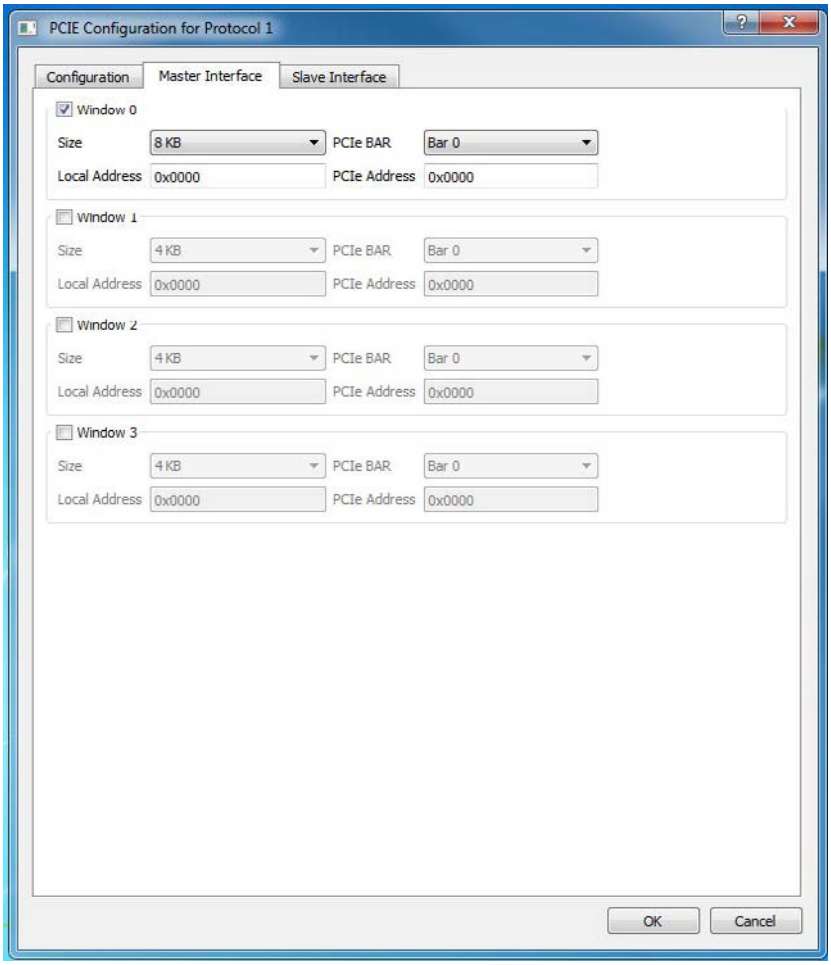
Parameter	Description
L1 Acceptable Latency	Select a latency value: • Maximum of 1 µs • Maximum of 2 µs • Maximum of 4 µs • Maximum of 8 µs • Maximum of 16 µs • Maximum of 32 µs • Maximum of 64 µs • No Limit
L1 Exit Latency Separate Clock/Common Clock Mode	Select the Exit Latency value of Separate/ Common Clock Mode. The Common Clock Mode value must be smaller than the Separate Clock Mode value. • Less than 1 µs • 1 µs to less than 2 µs • 2 µs to less than 4 µs • 4 µs to less than 8 µs • 8 µs to less than 16 µs • 16 µs to less than 32 µs • 32 µs to 64 µs • More than 64 µs

4.3 Master Interface

The Master Interface tab allows you to set up local AHB/AXI addresses on the master interface that will be mapped to or from a PCIe BAR and the address offset of the BAR. The PCIe or PCIe Reverse protocol allows you to use the Master Interface tab to configure up to four PCI windows (Window 0 through Window 3) with the following parameters.

- Size
- PCIe BAR (Base Address Register)
- Local Address
- PCIe Address

Figure 4-3. PCIe Configuration - Master Interface Tab



4.3.1 Size

For windows 0 through 3, select an available window size. The size you select is mapped to bits [31:12] of Windows*X*_1, where *X* can be 0, 1, 2, or 3.

- 4 Kbytes (*default*)
- 8 Kbytes
- 16 Kbytes
- 32 Kbytes
- 64 Kbytes
- 128 Kbytes
- 256 Kbytes
- 512 Kbytes
- 1 MB
- 2 MB
- 4 MB
- 8 MB
- 16 MB
- 32 MB
- 64 MB
- 128 MB
- 256 MB

- 512 MB
- 1 GB

4.3.2 PCIe BAR

Select one of the following Base Address Registers (BAR). The Bar Size you select is mapped to bits [5:0] of Windows*X*_2, where *X* can be 0, 1, 2, or 3.

- BAR0
- BAR1
- BAR2
- BAR3
- BAR4
- BAR5
- BAR0/1
- BAR2/3
- BAR4/5

4.3.3 Local Address

The address set in this field is the base address where transactions matching the window is mapped. The Local Address is 20 bits wide and covers the AXI/AHB address [31:12]. The lower order AXI/ AHB [11:0] are all zero in this control.

4.3.4 PCIe Address

The address set in this field is the offset address inside the selected BAR to which the transaction address is matched.

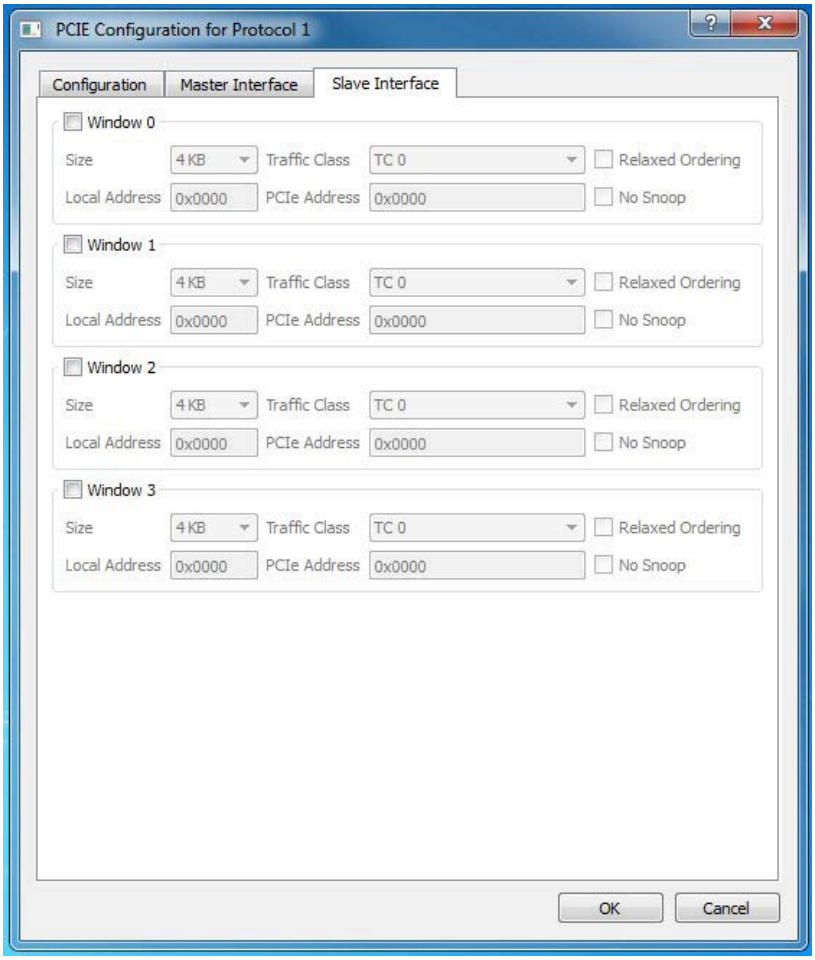
4.4 Slave Interface

The Slave Interface tab allows you to set a local AHB/AXI address on the slave interface that is mapped to a PCIe address.

- Size
- Local Address
- PCIe Address
- Traffic Class — selects the PCIe Traffic Class in the PCIe packet header.
- Relaxed Ordering — generates the PCIe TLP using a selectable relaxed ordering bit.
- No Snoop — generates the PCIe TLP using a selectable no snoop bit.

The Size, Local Address, and PCIe Address options are the same as those for the Master Interface. For more information, see 4.3. Master Interface.

Figure 4-4. PCIe Configuration - Slave Interface Tab



4.5 Traffic Class

Traffic Class allows you to set your Traffic Class and corresponding register bits. The traffic class is required by the PCI Express packet header. Its value determines the relative priority of a given transaction as it traverses the PCIe link. You can use the Traffic Class value to create a priority scheme for different packets.

- TC 0 (*default*)
- TC 1
- TC 2
- TC 3
- TC 4
- TC 5
- TC 6
- TC 7

5. PCIe- and XAUI-Specific Configuration Settings

5.1 PCIe/XAUI Fabric SPLL Configuration

The **PCIe Fabric SPLL Configuration** field applies to PCIe and XAUI protocols only.

- For the PCIe protocol, enter a valid value between 20 and 200 MHz for the **CLK_BASE Frequency**.
- For the XAUI protocol, the **CLK_BASE Frequency** is read-only and fixed at 156.25 MHz.

6. High-Speed Serial Interface Initialization Procedures

6.1 Cores with Built-in Initialization

If your core has a built-in initialization state machine, the core block is initialized with the user configurations at assertion/de-assertion of the INIT_RESET_N (Active Low) signal. When the configuration phase completes, the INIT_DONE signal is asserted and the SERDES block is ready for normal operations.
Note: The clock used for initialization should be a 50 MHz clock and connected to the INIT_CLK_50MHz signal.

You can have the SERDES initialization start automatically at power-up by connecting INIT_RESET_N (Active Low) input of the SERDES block to the POWER_ON_RESET_N (Active Low) signal of the SYSRESET macro.

6.2 Cores without Built-in Initialization

If your core does not have a built-in initialization state machine, you must build the configuration and initialization circuitry in SmartDesign for your core in addition to specifying configuration register values.

The following section describes how to build the initialization circuitry in a SmartDesign component EPCS_INIT.

The EPCS_INIT component consists of:

- CoreABC soft IPcore
- CoreAPB3 bus soft IPcore

Figure 6-1. EPCS_INIT SmartDesign Component (Non-PCIe)

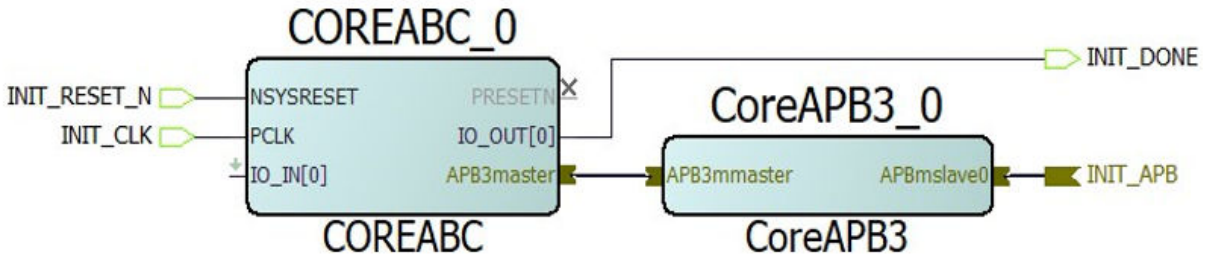
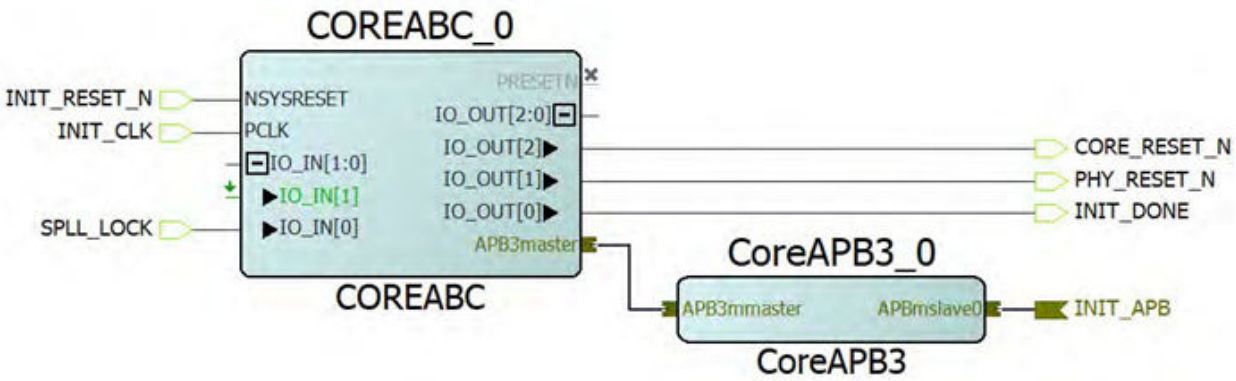


Figure 6-2. pcie_init Smartdesign Component



6.2.1 Configuring and Generating the SERDES Component

To configure and generate the SERDES component:

1. From the Catalog, right-click **RTG4 High Speed Serial Interface (PCIe, EPCS and XAUI)** and select **ConfigureCore**.

2. Click **OK** to exit the configurator. Your SERDES component is generated, and Libero generates the following CoreABC program: `<project_location/component/work/../<SERDES_component_name>/<SERDES_component_name>_0/<SERDES_location_name>_init_abc.txt`

   The CoreABC program performs the following activities:
   - Configures PMA, system registers
   - De-asserts PHY_RESET_N
   - Performs the SPLL calibration/workaround, which includes polling for SPLL_LOCK output from SERDES
   - De-asserts CORE_RESET_N
   - Configures PCIe registers (PCIe cores only)
   - Asserts INIT_DONE

After configuring and generating the SERDES component, build the initialization circuitry for the non-PCIe core or PCIe core.

6.2.2 Non-PCIe Cores

The following sections apply to non-PCIe cores.

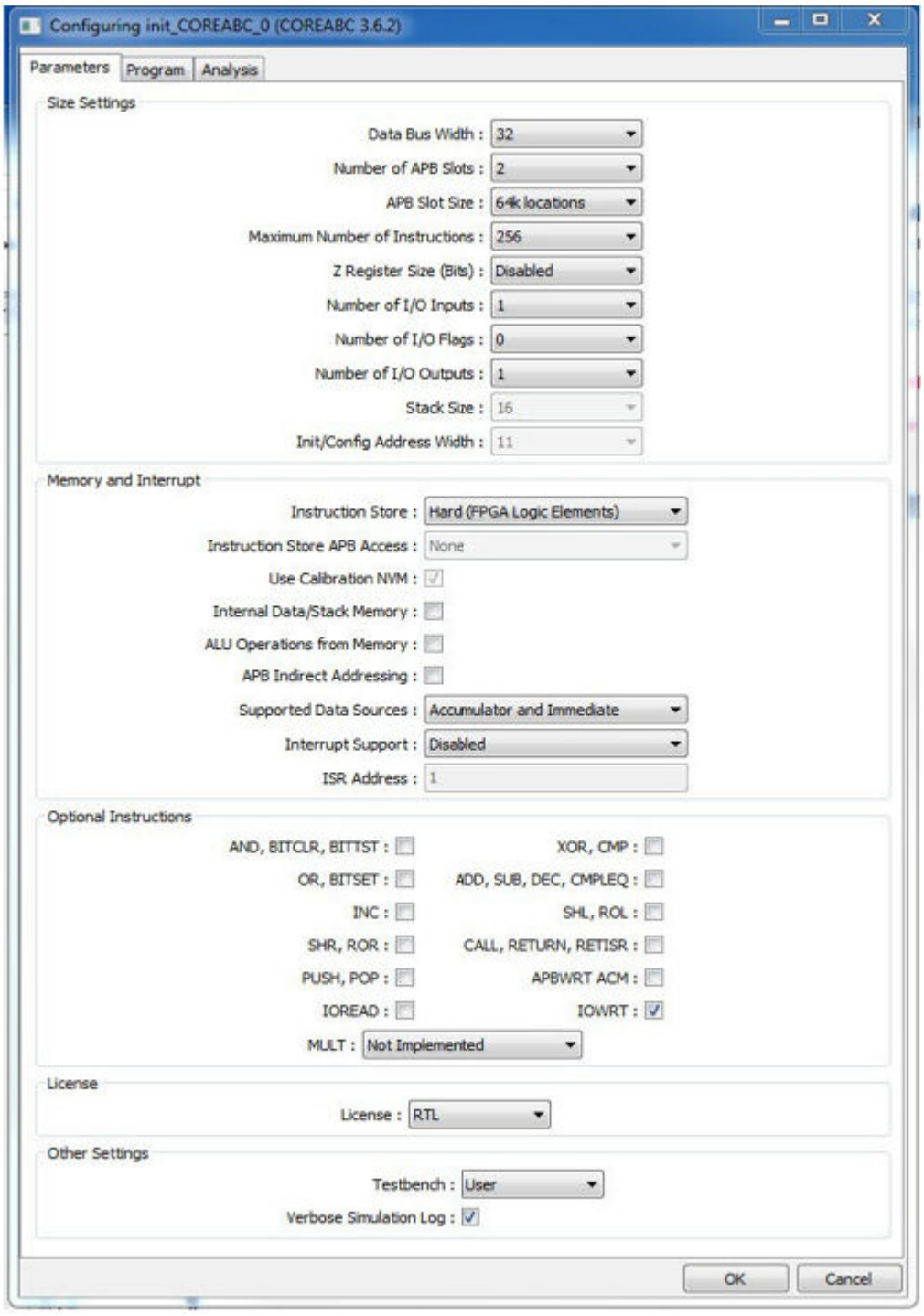
6.2.2.1 Building the Initialization Circuitry for Non-PCIe Cores

For non-PCIe cores, use the following procedure to build the initialization circuitry after configuring and generating the EPCS component.

To build the initialization circuitry for non-PCIe cores:

1. Configure and generate the SERDES component.
2. Create a SmartDesign component with the name EPCS_INITnamepcie_init.
3. From the Catalog, drag and drop CoreABC (v3.6.100 or later) to the SmartDesign component pcie_init Enter the tasks to be performed after finishing this task (optional).
4. Specify a name to the CoreABC component and click **OK** to open its configurator.
5. Configure the CoreABC component as shown in the following figure. Make sure the following parameters are selected in the Parameters tab:
   - Data Bus Width: 32
   - Maximum Number of Instructions: at least 256
   - Number of I/O Inputs: 2
   - Number of I/O Flags: 1
   - Number of I/O Outputs: 1
   - Instruction Store: Hard (FPGA Logic Elements)
   - Optional Instructions: IOWRT

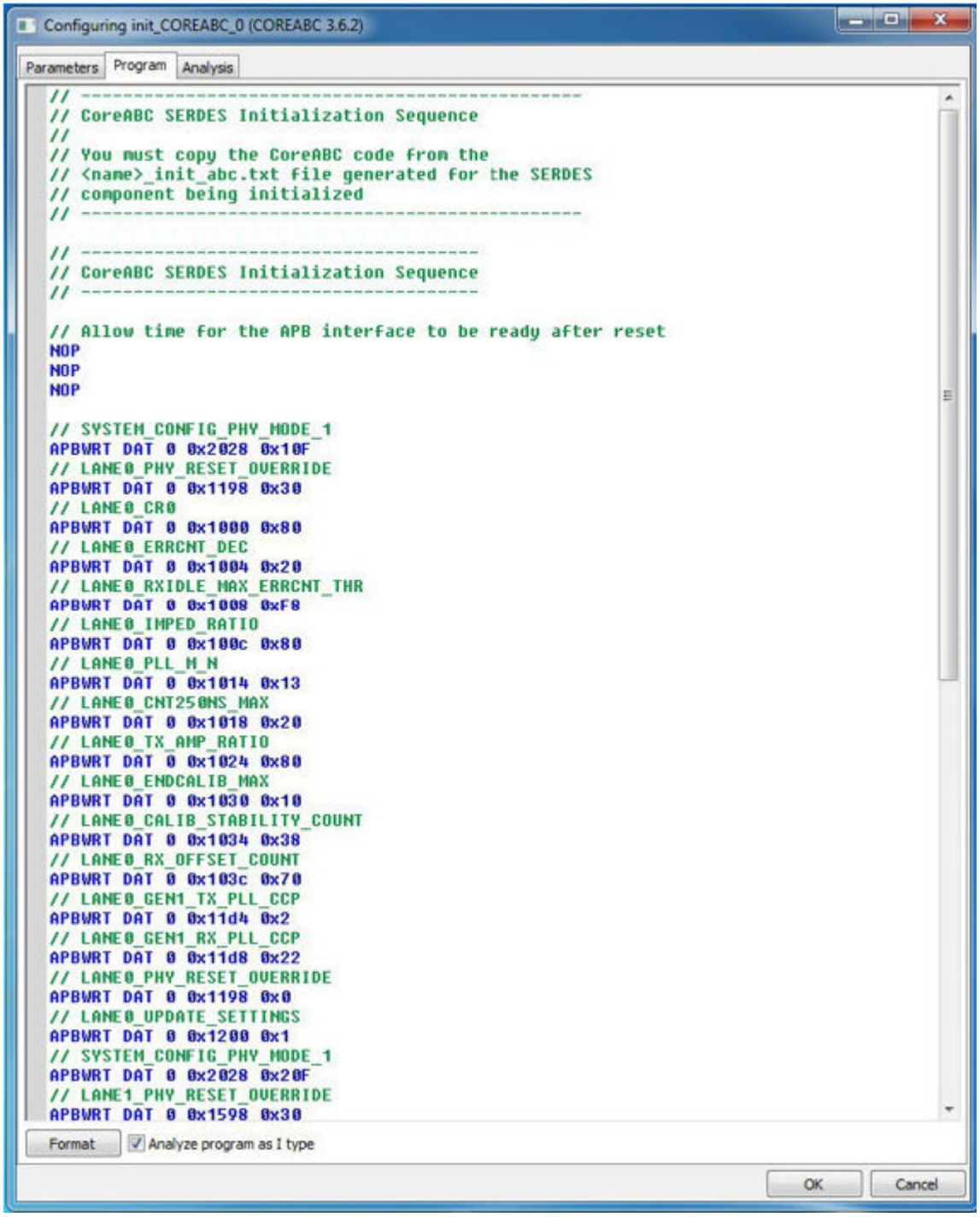
Figure 6-3. CoreABC Parameter Configuration



6. Copy the CoreABC program generated for your SERDES (EPCS) block from the `<SERDES_location_name>init_abc.txt` file created under the `<project_location>/component/work/../ component_name/component_name_<0>` folder and paste it to the CoreABC Program tab (see the following figure). The program code loads the SERDES configuration registers with the values you configured and starts the initialization sequence.

Note: Depending on the number of instructions generated in the `<SERDES_location_name>_init_abc.txt` file, you may have to increase the maximum number of instructions.

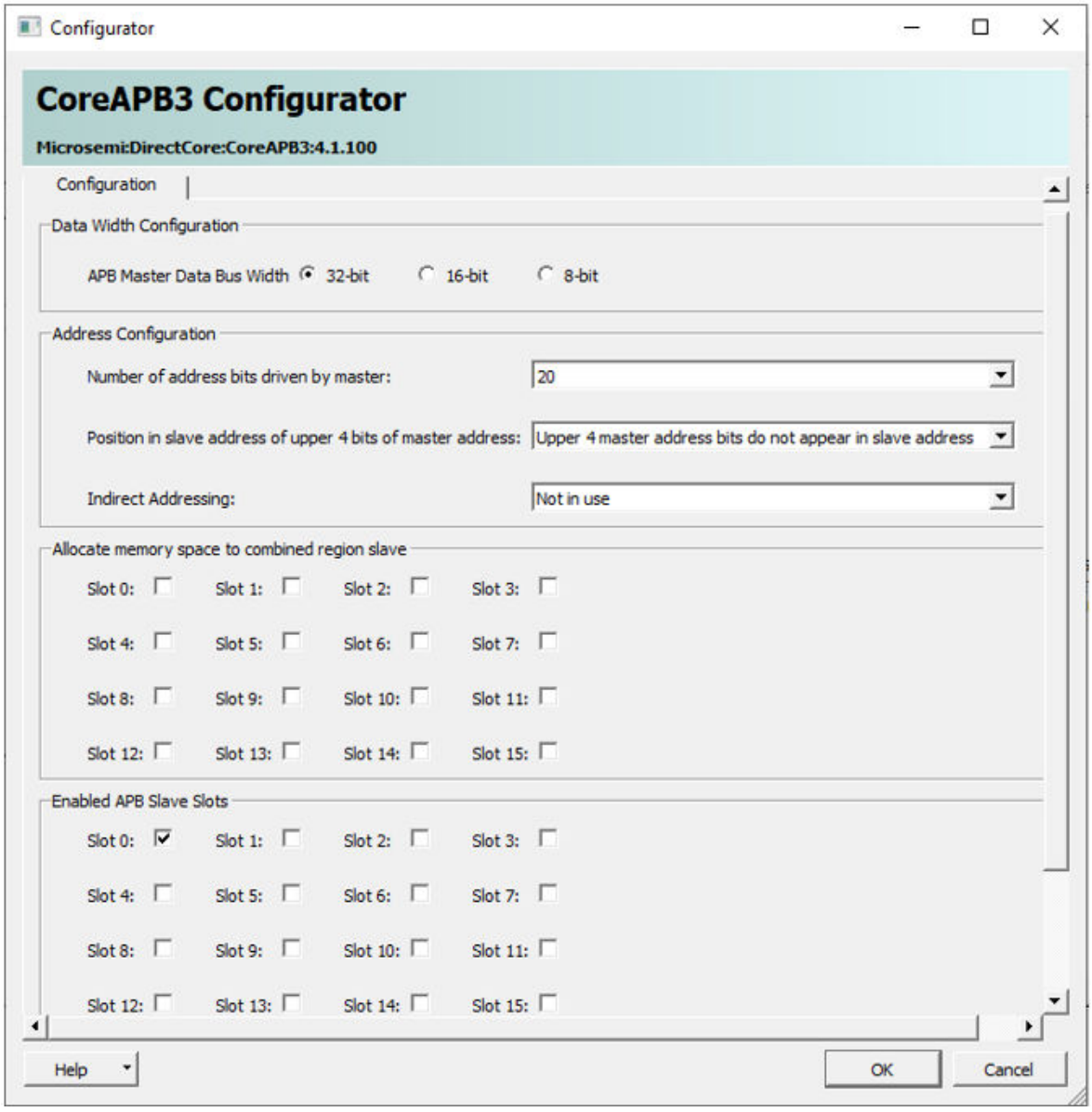
Figure 6-4. CoreABC Program Code for EPCS SERDES



7. Click **OK** to exit the configurator and generate the CoreABC component.
8. From the Catalog, drag and drop CoreAPB3 (v4.1.100 or later) to the SmartDesign component pcie_init.
9. Specify a name to the CoreAPB3 component and click **OK** to open its configurator.

10. Configure the CoreAPB3 component as shown in the following figure. Make sure it has the following selections:
    - APB Master Data Bus Width: 32
    - Number of address bits driven by master: 20
    - Enabled APB Slave Slots is Slot 0. All other Save Slots are disabled.

Figure 6-5. CoreAPB3 Configuration



11. Click **OK** to exit the configurator and generate the CoreAB3 component.
12. Make the necessary port tie-offs, promotions, and connections between the CoreABC and CoreAPB3 components in the pcie_init SmartDesign canvas (see the following table).

Table 6-1. Port Tie-offs, Promotions, and Connections in pcie_init

Port/Bus Interface (BIF) Name/Block	Action
NSYSRESET/CoreABC	Promote to top and rename to INIT_RESET_N.
PCLK/CoreABC	Promote to top and rename to INIT_CLK.

.....continued	
Port/Bus Interface (BIF) Name/Block	**Action**
IO_IN[0]/CoreABC	Tie-off to GND.
PRESETN/CoreABC	Mark as unused.
IO_OUT[0]/CoreABC	Promote to top and rename to INIT_DONE.
APB3master BIF/CoreABC	Connect to APB3mmaster BIF of CoreAPB3.
APBmslave0/CoreAPB3	Promote to top and rename to INIT_APB.

13. Click the **Generate** button in the SmartDesign canvas to generate the pcie_init component.

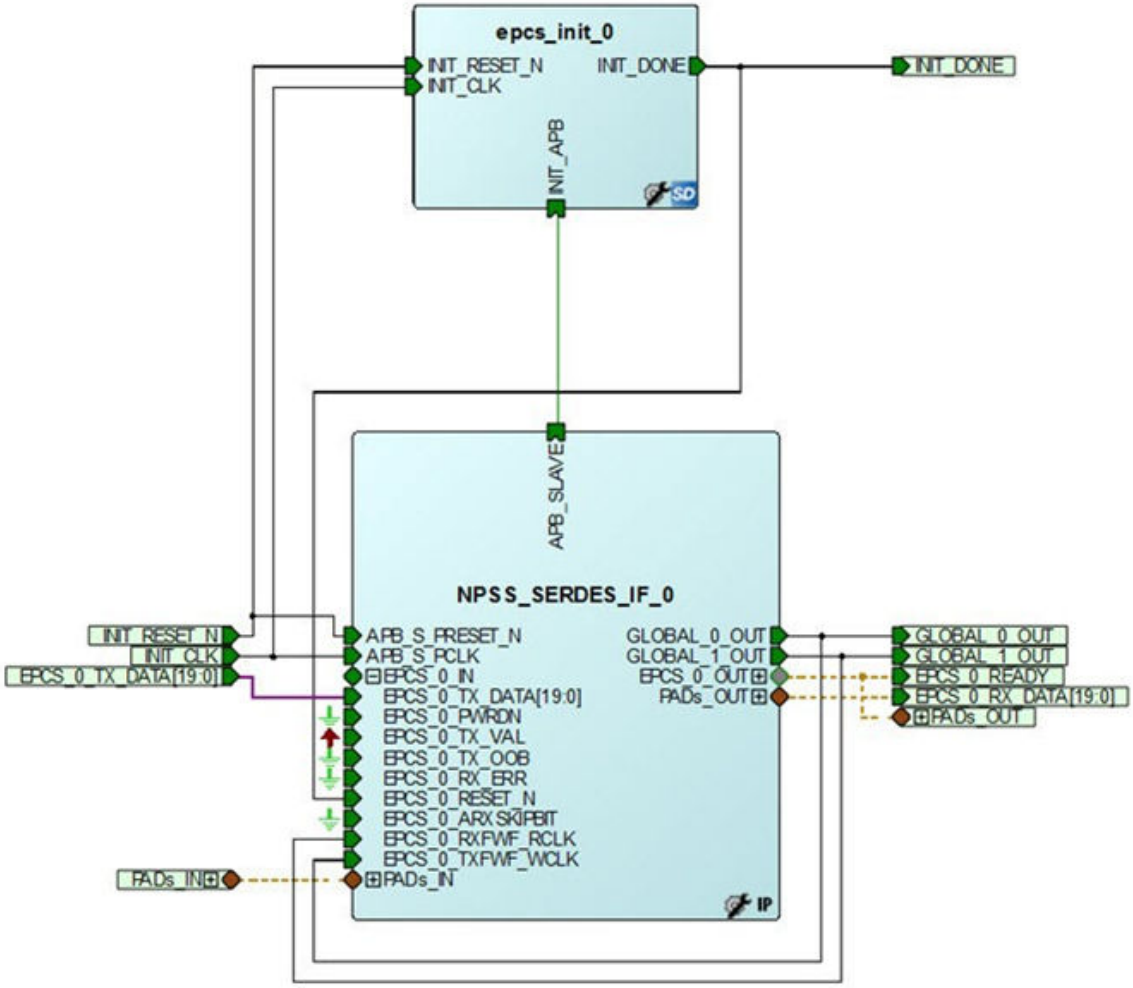
Proceed to the next section to interface SERDES (EPCS) core with the initialization logic.

6.2.2.2 Interfacing SERDES (EPCS) with the Initialization Logic

To interface SERDES with the initialization logic:

1. Build the initialization circuitry for SERDES (EPCS) core.
2. Drag and drop the generated SERDES (EPCS) component from the Design Hierarchy window into the top level SmartDesign canvas.
3. Drag and drop the SmartDesign component EPCS_INIT from the Design Hierarchy window into the same SmartDesign canvas where you instantiated the SERDES (EPCS) component.
4. To interface the NPSS_SERDES to the initialization logic block EPCS_INIT, make the following connections between the EPCS_INIT block and your NPSS_SERDES block in the SmartDesign.
5. When completed, click the **Generate** button in SmartDesign to generate the EPCS SERDES subsystem.

Figure 6-6. SERDES (EPCS) Subsystem Initialization Circuity



6.2.3 PCIe Cores

The following sections apply to PCIe cores.

6.2.3.1 Building the Initialization Circuitry for PCIe Cores

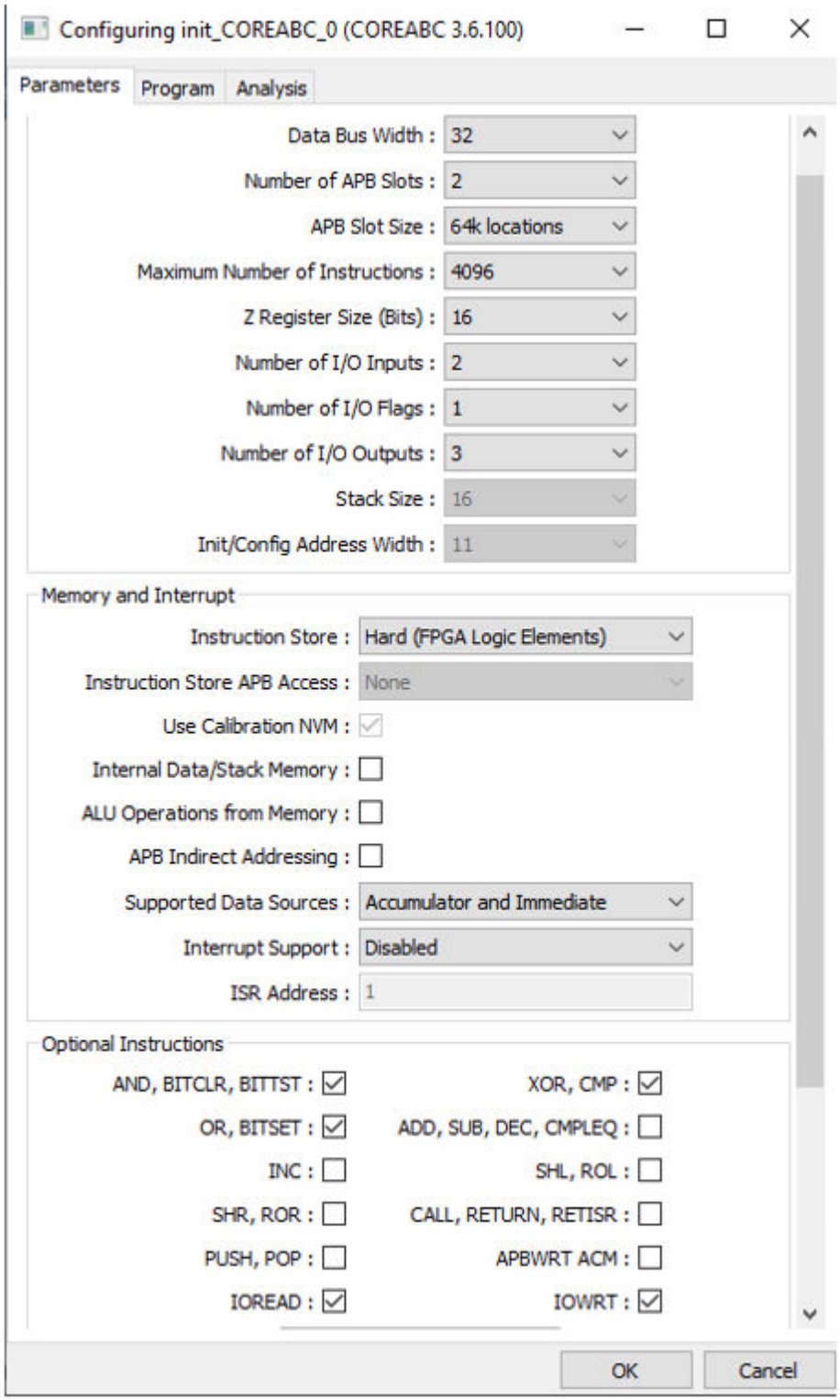
For PCIe cores, use the following procedure to build the initialization circuitry after configuring and generating the SERDES component.

To build the initialization circuitry for PCIe cores:

1. Configure and generate the SERDES component.
2. Create a SmartDesign component with the name pcie_init.
3. From the Catalog, drag and drop CoreABC (v3.6.100 or later) to the SmartDesign component pcie_init.
4. Optional: Enter the tasks to be performed.
5. Specify a name to the CoreABC component and click **OK** to open its configurator.
6. Configure the CoreABC component as shown in the following figure. Make sure the following parameters are selected in the Parameters tab:
    - Data Bus Width: 32
    - Maximum Number of Instructions: at least 256
    - Number of I/O Inputs: 2
    - Number of I/O Flags: 1

- Number of I/O Outputs: 3
- Instruction Store: Hard (FPGA Logic Elements)
- Optional Instructions: AND, OR, IOREAD, and IOWRT

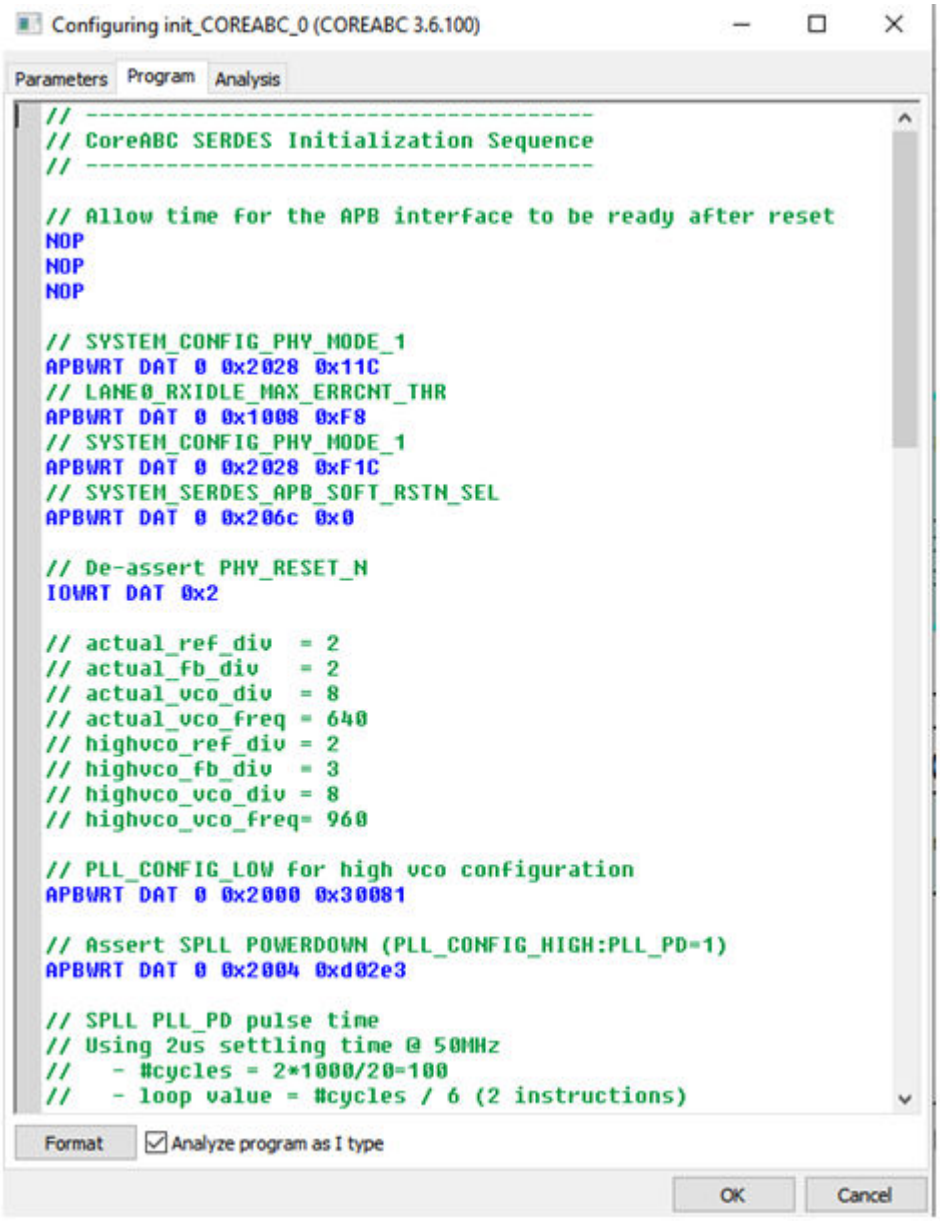
Figure 6-7. CoreABC Parameter Configuration



7. Copy the CoreABC program generated for your SERDES (pcie_init) block from the *<SERDES_location_name>init_abc.txt* file created under the *<project_location>/component/ work/../ component_name/component_name_<0>* folder and paste it to the **CoreABC Program** tab (see the following figure). The program code loads the SERDES configuration registers with the values you configured and starts the initialization sequence.

   Note: Depending on the number of instructions generated in the *<SERDES_location_name>_init_abc.txt* file, you may have to increase the maximum number of instructions.

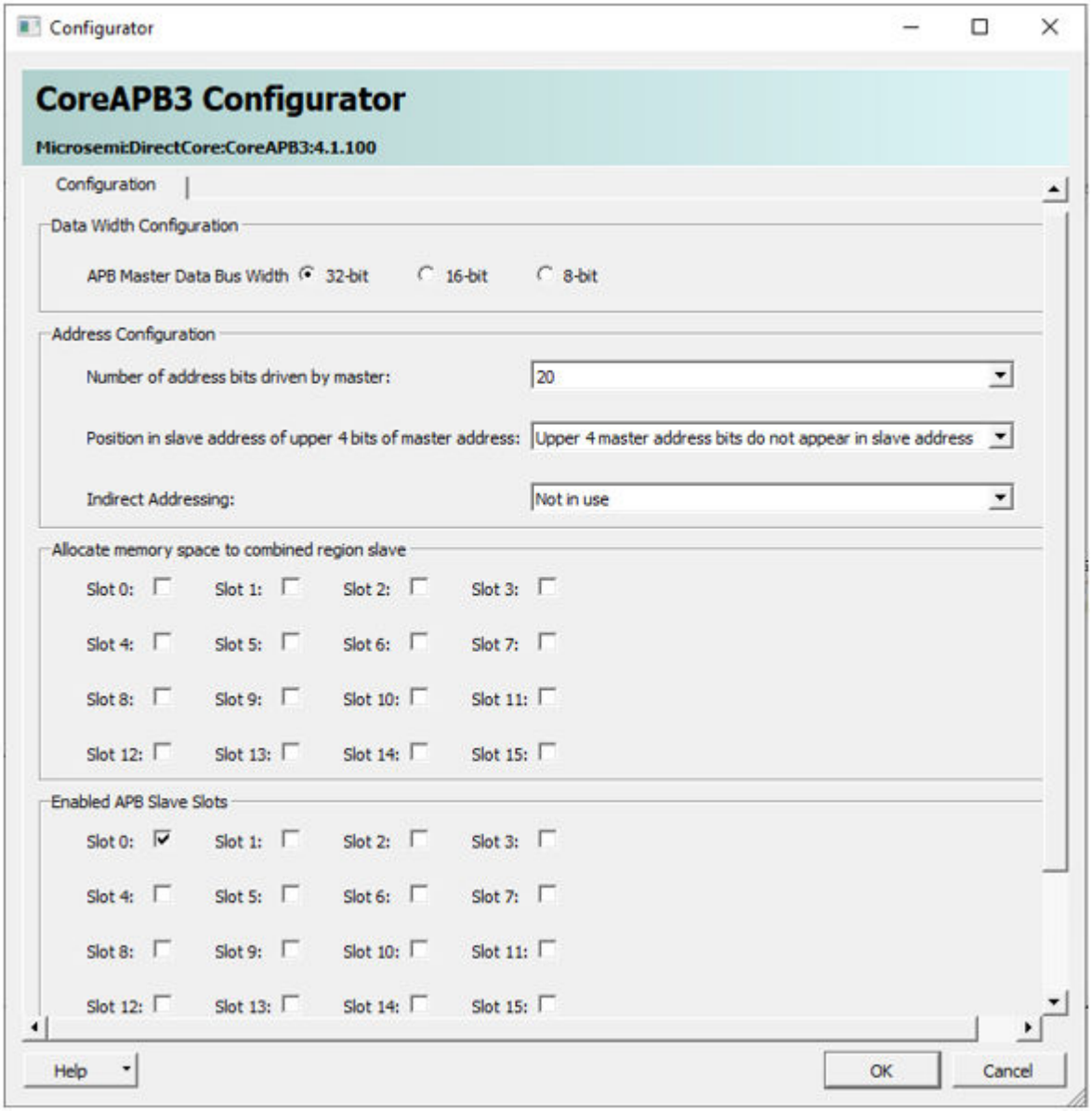
   Figure 6-8. CoreABC Program Code for PCIe SERDES



8. Click **OK** to exit the configurator and generate the CoreABC component.
9. From the Catalog, drag and drop CoreAPB3 (v4.1.100 or later) to the SmartDesign component pcie_init.
10. Specify a name to the CoreAPB3 component and click **OK** to open its configurator.
11. Configure the CoreAPB3 component as shown in the following figure. Make sure it has the following selections:
    - APB Master Data Bus Width: 32

- Number of address bits driven by master: 20
- Enabled APB Slave Slots is Slot 0. All other Save Slots are disabled. If you plan to implement Hot Reset and L2 Exit detection logic as a part of the PCIe initialization circuitry, enable Slave Slot 1.

Figure 6-9. CoreAPB3 Configuration



12. Click **OK** to exit the configurator and generate the CoreAB3 component.
13. Make the necessary port tie-offs, promotions, and connections between the CoreABC and CoreAPB3 components in the pcie_init SmartDesign canvas (see the following table).

Table 6-2. Port Tie-offs, Promotions, and Connections in pcie_init

Port/Bus Interface (BIF) Name/Block	Action
NSYSRESET/CoreABC	Promote to top and rename to INIT_RESET_N.
PCLK/CoreABC	Promote to top and rename to INIT_CLK.

.....continued	
Port/Bus Interface (BIF) Name/Block	**Action**
IO_IN[1:0]/CoreABC	Slice the bus into two single bit inputs IO_IN[0] and IO_IN[1]. Promote IO_IN[0] to top and rename to SPLL_LOCK. This will be used to poll for SPLL_LOCK output of the SERDES block. IO_IN[1] is for Hot Reset and L2 Exit detection logic. If your design does not use the Hot Reset and L2 Exit detection logic, tie-off IO_IN[1] to GND.
PRESETN/CoreABC	Mark as unused.
IO_OUT[2:0]/CoreABC	Slice the bus into three single bit outputs IO_OUT[0], IO_OUT[1] and IO_OUT[2]. Promote all three to top and rename them as follows: • IO_OUT[0] to INIT_DONE • IO_OUT[1] to PHY_RESET_N • IO_OUT[2] to CORE_RESET_N
APB3master BIF/CoreABC	Connect to APB3mmaster BIF of CoreAPB3.
APBmslave0/CoreAPB3	Promote to top and rename to INIT_APB.

14. Click the **Generate** button in the SmartDesign canvas to generate the pcie_init component.

Proceed to the next section to interface the SERDES (PCIe) core with the initialization logic.

6.2.3.2 Interfacing SERDES (PCIe) with the Initialization Logic

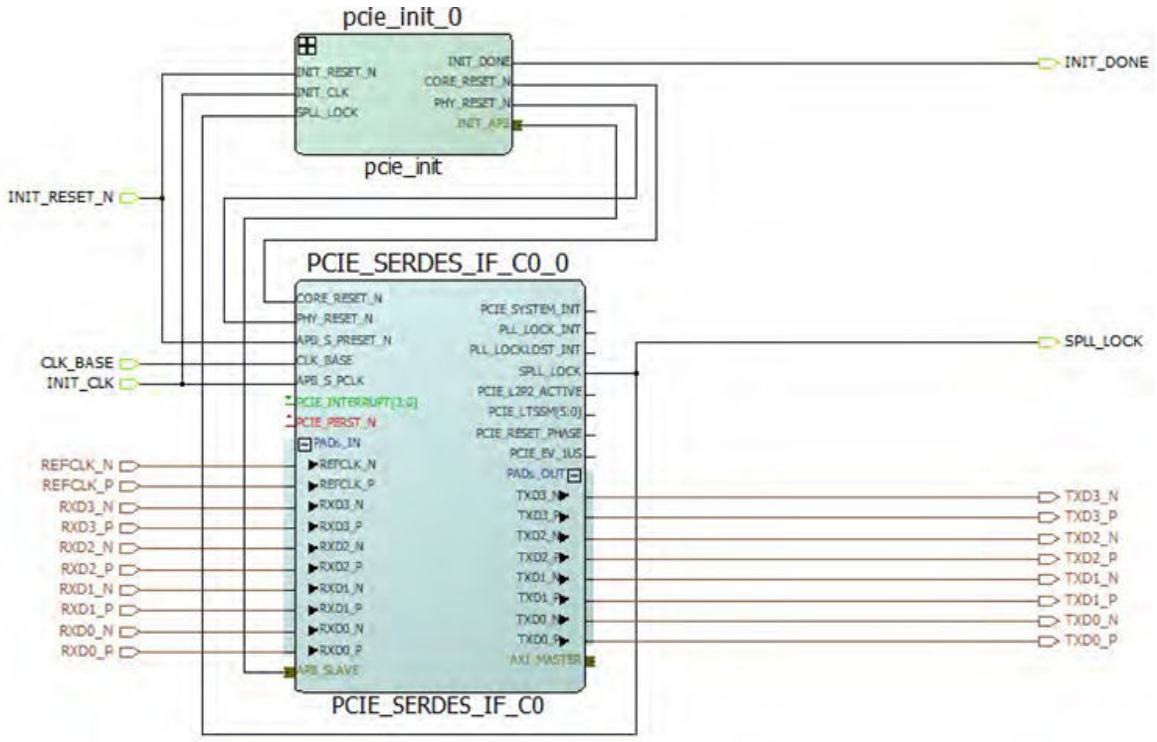
To interface SERDES with the initialization logic:

1.
2. Build the initialization circuitry for the SERDES (PCIe) core.
3. Drag and drop the generated SERDES (PCIe) component from the Design Hierarchy window into the top level SmartDesign canvas.
4. Drag and drop the SmartDesign component pcie_init from the Design Hierarchy window into the same SmartDesign canvas where you instantiated the SERDES component.
5. To interface the NPSS_SERDES to the initialization logic block pcie_init, make the following connections between the pcie_init block and your PCIe SERDES block in the SmartDesign.

Port/Bus Interface (BIF) of SERDES Block	Port/Bus Interface (BIF) Name/Block
APB_SLAVE/SERDES block	INIT_APB/Init Block
APB_S_PCLK/SERDES block	INIT_CLK/Init Block
PHY_RESET_N/SERDES block	PHY_RESET_N/Init Block
CORE_RESET_N/SERDES block	CORE_RESET_N/Init Block
APB_S_PRESET_N/ SERDES block	INIT_RESET_N/Init Block
SPLL_LOCK/SERDES block	SPL_LOCK/Init Block
PCIE_L2P2_ACTIVE/SERDE S block	Can be used to connect to Hot Reset and L2 Exit detection logic if implemented.

6. When completed, click the **Generate** button in SmartDesign to generate the PCIe SERDES subsystem.

Figure 6-10. PCIe Subsystem Initialization Circuitry



This completes the procedure for the SERDES (PCIe) high-speed serial interface initialization.

7. Port Descriptions

7.1 Common Ports

The following topics describe the ports applicable to all RTG4 high-speed serial interface cores covered in this guide.

7.1.1 APB Ports

The following table provides a list of APB ports.

Table 7-1. APB Ports

Port	Direction	Port Group
APB_S_PRDATA[31:0]	OUT	APB_SLAVE
APB_S_PREADY	OUT	
APB_S_PSLVERR	OUT	
APB_S_PADDR[13:2]	IN	
APB_S_PENABLE	IN	
APB_S_PSEL	IN	
APB_S_PWDATA[31:0]	IN	
APB_S_PWRITE	IN	
APB_S_PCLK	IN	
APB_S_PRESET_N	IN	

7.1.2 XAUI Control Ports

The following table provides a list of XAUI control ports.

Table 7-2. XAUI Control Ports

Port	Direction
CORE_RESET_N	IN
PHY_RESET_N	IN
SPLL_LOCK	OUT
PLL_LOCK_INT	OUT
PLL_LOCKLOST_INT	OUT

7.1.3 XAUI Ports

The following table provides a list of XAUI ports.

Table 7-3. XAUI Ports

Port	Direction
XAUI_RXD[63:0]	OUT
XAUI_RXC[7:0]	OUT
XAUI_VNDRESLO[7:0]	IUT
XAUI_VNDRESHI[7:0]	OUT

.....continued	
Port	**Direction**
XAUI_MMD_MDC	IN
XAUI_MMD_MDI	IN
XAUI_MMD_MDI_EXT	IN
XAUI_MMD_MDOE_IN	IN
XAUI_MMD_PRTAD[4:0]	IN
XAUI_MMD_DEVID[4:0]	IN
XAUI_LOOPBACK_IN	IN
XAUI_MDC_RESET	IN
XAUI_TX_RESET	IN
XAUI_RX_RESET	IN
XAUI_TXD[63:0]	IN
XAUI_TXC[7:0]	IN
XAUI_MMD_MDO	OUT
XAUI_MMD_MDOE	OUT
XAUI_LOWPOWER	OUT
XAUI_LOOPBACK_OUT	OUT
XAUI_TX_CLK_OUT	OUT
XAUI_RX_CLK_OUT	OUT
XAUI_PHY_NOT_READY	OUT
XAUI_RX_CLK_IN	IN
XAUI_FB_CLK	IN
XAUI_LANE01_RX_ERR	IN
XAUI_LANE23_RX_ERR	IN
XAUI_PWRDN	IN **Note:** When this input is driven high, it powers down the XAUI core. Tie Low for normal operation.
XAUI_TX_OOB	IN **Note:** Not used for XAUI. Tie Low for normal operation.

7.1.4 EPCS Ports

The following table provides a list of EPCS ports.

Table 7-4. EPCS Ports

Port	Direction	Port Group
EPCS_<n>_PWRDN	IN	EPCS_<n>_IN where n can be 0, 1, 2, or 3, depending on the number of configured lanes.
EPCS_<n>_TX_VAL	IN	
EPCS_<n>_TX_OOB	IN	
EPCS_<n>_RX_ERR	IN	
EPCS_<n>_RESET_N	IN	
EPCS_<n>_TX_DATA[<wd>:0]	IN	
ECPS_<n>_ARXSKIPBIT	IN	
EPCS_<n>_RXFWF_RCLK	IN	
EPCS_<n>_TXFWF_WCLK	IN	
EPCS_FAB_REF_CLK when Fabric is selected as the Reference Clock Source in the Configurator	IN	
INIT_RESET_N	IN	Active Low signal. Assert/de-assert this signal to start the initialization of the SERDES.
INIT_CLK_50MHZ	IN	Initialization clock. Must be connected to a 50MHz clock source.
EPCS_<n>_READY	OUT	EPCS_<n>_OUT where n can be 0, 1, 2, or 3, depending on the number of configured lanes.
EPCS_<n>_TX_CLK_STABLE	OUT	
EPCS_<n>_TX_CLK	OUT	
EPCS_<n>_RX_CLK	OUT	
EPCS_<n>_RX_VAL	OUT	
EPCS_<n>_RX_IDLE	OUT	
EPCS_<n>_TX_RESET_N	OUT	
EPCS_<n>_RX_RESET_N	OUT	
EPCS_<n>_RX_DATA[<wd>:0]	OUT	
GLOBAL_0_OUT	OUT	
GLOBAL_1_OUT	OUT	
INIT_DONE	OUT	Asserted high when the initialization of the SERDES block is complete.

Observe the following guidelines:

- <n> indicates the lane on which EPCS is configured.
- Valid values for <wd> are 19,15, 9, 7, 4, and 3.
- GLOBAL_0_OUTand GLOBAL_1_OUT ports are available only if they have been configured in the configurator. For more information, see 3.3.1. EPCS Lane TX/RX Clock Selection.

7.1.5 PAD Ports

The following table provides a list of PAD ports and description.

Table 7-5. PAD Ports

Port	Direction	Port Group	Description
RXD0_P, RXD0_N	IN	PADs_IN	Differential input pair for lane 0 (Rx data).
RXD1_P, RXD1_N	IN		Differential input pair for lane 1 (Rx data).
RXD2_P, RXD2_N	IN		Differential input pair for lane 2 (Rx data).
RXD3_P, RXD3_N	IN		Differential input pair for lane 3 (Rx data).
REFCLK_P, REFCLK_N	IN		Differential input reference clock pair.
TXD0_P, TXD0_N	OUT	PADs_OUT	Differential output pair for lane 0 (Tx data).
TXD1_P, TXD1_N	OUT		Differential output pair for lane 1 (Tx data).
TXD2_P, TXD2_N	OUT		Differential output pair for lane 2 (Tx data).
TXD3_P, TXD3_N	OUT		Differential output pair for lane 3 (Tx data).

7.2 PCIe Ports

The following ports apply to PCIe cores.

7.2.1 PCIe Ports

Table 7-6. PCIe Ports

Port	Direction
CORE_RESET_N	IN
PHY_RESET_N	IN
CLK_BASE	IN
PCIE_INTERRUPT[3:0]	IN
PCIE_PERST_N	IN
PCIE_WAKE_REQ	IN
PCIE_SYSTEM_INT	OUT
SPLL_LOCK	OUT
PLL_LOCK_INT	OUT
PLL_LOCKLOST_INT	OUT
PCIE_EV_1US	OUT
PCIE_LTSSM[5:0]	OUT
PCIE_L2P2_ACTIVE	OUT

.....continued	
Port	**Direction**
PCIE_RESET_PHASE	OUT
PCIE_WAKE_N	OUT

7.2.2 PCIe AXI Master Ports

The following table provides a list of PCIe AXI Master ports.

Table 7-7. PCIe AXI Master Ports

Port	Direction	Port Group
AXI_M_AWID[3:0]	OUT	AXI_MASTER
AXI_M_AWADDR[31:0]	OUT	
AXI_M_AWLEN[3:0]	OUT	
AXI_M_AWSIZE[1:0]	OUT	
AXI_M_AWBURST[1:0]	OUT	
AXI_M_AWVALID	OUT	
AXI_M_AWREADY	IN	
AXI_M_WID[3:0]	OUT	
AXI_M_WSTRB[7:0]	OUT	
AXI_M_WLAST	OUT	
AXI_M_WVALID	OUT	
AXI_M_WDATA[63:0]	OUT	
AXI_M_WREADY	IN	
AXI_M_BID[3:0]	IN	
AXI_M_BRESP[1:0]	IN	
AXI_M_BVALID	IN	
AXI_M_BREADY	OUT	
AXI_M_ARID[3:0]	OUT	
AXI_M_ARADDR[31:0]	OUT	
AXI_M_ARLEN[3:0]	OUT	
AXI_M_ARSIZE[1:0]	OUT	
AXI_M_ARBURST[1:0]	OUT	
AXI_M_ARVALID	OUT	
AXI_M_ARREADY	IN	
AXI_M_RID[3:0]	IN	
AXI_M_RDATA[63:0]	IN	
AXI_M_RRESP[1:0]	IN	
AXI_M_RLAST	IN	
AXI_M_RVALID	IN	
AXI_M_RREADY	OUT	

7.2.3 PCIe AXI Slave Ports

The following table provides a list of PCIe AXI Slave ports.

Table 7-8. PCIe AXI Slave Ports

Port	Direction	Port Group
AXI_S_AWID[3:0]	IN	AXI_SLAVE
AXI_S_AWADDR[31:0]	IN	
AXI_S_AWLEN[3:0]	IN	
AXI_S_AWSIZE[1:0]	IN	
AXI_S_AWBURST[1:0]	IN	
AXI_S_AWVALID	IN	
AXI_S_AWREADY	OUT	
AXI_S_AWLOCK[1:0]	IN	
AXI_S_WID[3:0]	IN	
AXI_S_WSTRB[7:0]	IN	
AXI_S_WLAST	IN	
AXI_S_WVALID	IN	
AXI_S_WDATA [63:0]	IN	
AXI_S_WREADY	OUT	
AXI_S_BID[3:0]	OUT	
AXI_S_BRESP[1:0]	OUT	
AXI_S_BVALID	OUT	
AXI_S_BREADY	IN	
AXI_S_ARID[3:0]	IN	
AXI_S_ARADDR[31:0]	IN	
AXI_S_ARLEN[3:0]	IN	
AXI_S_ARSIZE[1:0]	IN	
AXI_S_ARBURST[1:0]	IN	
AXI_S_ARVALID	IN	
AXI_S_ARLOCK[1:0]	IN	
AXI_S_ARREADY	OUT	
AXI_S_RID[3:0]	OUT	
AXI_S_RDATA[63:0]	OUT	
AXI_S_RRESP[1:0]	OUT	
AXI_S_RLAST	OUT	
AXI_S_RVALID	OUT	
AXI_S_RREADY	IN	

7.2.4 PCIe AHBLite Master Ports

The following table provides a list of PCIe AHBLite Master ports.

Table 7-9. PCIe AHBLite Master Ports

Port	Direction	Port Group
AHB_M_HADDR[31:0]	OUT	AHB_MASTER
AHB_M_HBURST[1:0]	OUT	
AHB_M_HSIZE[1:0]	OUT	
AHB_M_HTRANS[1:0]	OUT	
AHB_M_HWRITE	OUT	
AHB_M_HWDATA[31:0]	OUT	
AHB_M_HREADY	IN	
AHB_M_HRESP	IN	
AHB_M_HRDATA[31:0]	IN	

7.2.5 PCIe AHBLite Slave Ports

The following table provides a list of PCIe AHBLite Slave ports.

Table 7-10. PCIe AHBLite Slave Ports

Port	Direction	Port Group
AHB_S_HSEL	IN	AHB_SLAVE
AHB_S_HADDR[31:0]	IN	
AHB_S_HBURST[1:0]	IN	
AHB_S_HSIZE[1:0]	IN	
AHB_S_HTRANS[1:0]	IN	
AHB_S_HWRITE	IN	
AHB_S_HWDATA[31:0]	IN	
AHB_S_HREADYOUT	OUT	
AHB_S_HRESP	OUT	
AHB_S_HREADY	IN	
AHB_S_HRDATA[31:0]	OUT	

8. Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Description
C	11/2021	Updated 3.5. Signal Integrity Options.
B	08/2021	• Updated 4.1. Configuration. • Added a note in the 3.1. Identification section regarding the dependency of the SERDES block names on the number of blocks present in the device. • Updated the data widths with supported data rates for selected clock frequencies in the 3.2.4. Lane Configuration section.
A	11/2020	Initial Revision. Document is converted to the Microchip template.

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.

- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY,

ISBN: 978-1-5224-9155-2

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820