HB0946
Handbook
CorePCIe_AHBLtoAXI v2.0

Microsemi
a Microchip company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

©2021 Microsemi, a wholly owned subsidiary of Microchip Technology Inc. All rights reserved. Microsemi and the Microsemi logo are registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

50200946. 1.0 4/21

Contents

1 Revision History 1
1.1 Revision 1.0 1
2 Introduction 2
2.1 Overview 2
2.2 Features 2
2.3 Core Version 2
2.4 Supported Families 2
2.5 Device Utilization and Performance 3
3 Core Description 4
3.1 Burst Support 4
3.2 Control Signal Handling 4
3.3 Write Transactions 5
3.4 Read Transactions 5
3.5 Response Handling 5
4 Interface 6
4.1 Configuration Parameters 6
4.2 Ports 6
5 Tool Flow 10
5.1 License 10
5.2 Using core in Libero SmartDesign 10
5.3 Simulation Flows 10
5.4 Synthesis in Libero 10
5.5 Place-and-Route in Libero 10
6 Clocking and Reset 12
7 Integration with SERDES in Libero SmartDesign 13

Figures

Figure 1 CorePCIe_AHBLtoAXI Bridge Block Diagram 2
Figure 2 Core Instance View in SmartDesign 10
Figure 3 High Speed Serial Interface Configurator 13
Figure 4 PCIE Configuration for Protocol 1 14
Figure 5 AHB and AXI Bus Connections 15

Tables

Table 1 Device Utilization and Performance 3
Table 2 AHBL and AXI Burst Support 4
Table 3 Input and Output Signals 6

1 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.

1.1 Revision 1.0

The first publication of this document. Created for CorePCIe_AHBLtoAXI v2.0.

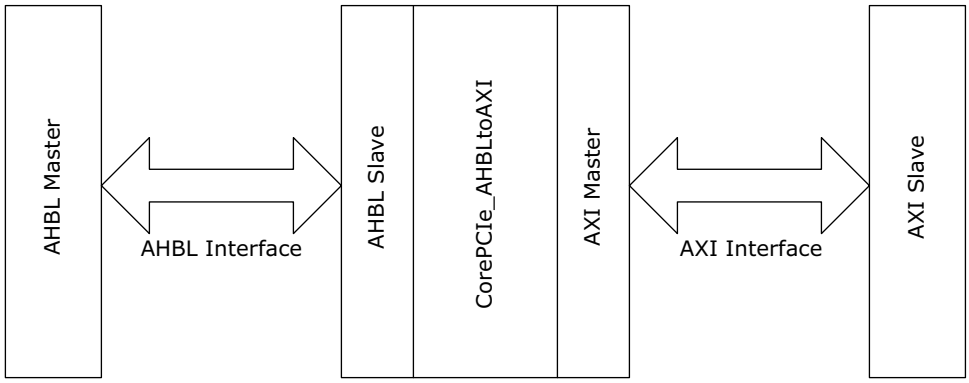
2 Introduction

2.1 Overview

The CorePCIe_AHBLtoAXI IP core provides an interface (bridge) between the AHBL domain and AXI domain. The IP core is an AHBL slave and an AXI master. The core allows an AHBL bus system to be connected to an AXI bus enabling an AHBL master to communicate with an AXI slave.

The IP core is recommended to be used with SmartFusion®2, IGLOO®2, and RTG4™ PCIe AXI Slave to connect to an AHBL master or AHBL sub-system.

Figure 1 • **CorePCIe_AHBLtoAXI Bridge Block Diagram**



2.2 Features

The following features are supported in CorePCIe_AHBLtoAXI:

- Compliant to AMBA 3 AHB-Lite and AMBA 3 AXI specifications
- Provides an interface between the AHBL domain and the AXI domain
- AHBL and AXI domains are synchronous (common clock for both AHBL and AXI)
- Generates only single-beat increment burst AXI transactions
- Converts 32-bit AHBL write/read transactions into 64-bit AXI write/read transactions, respectively
- Controls the bus responses from AXI-Slave to AHBL-Master

The following features are not supported in CorePCIe_AHBLtoAXI:

- Protection (HPROT) and lock (HMASTLOCK) signal is not supported on the AHB interface
- Protection (AWPROT/ARPROT) and cached (AWCACHE/ARCACHE) transfers are not supported on the AXI interface
- Locked transfers are not supported on the AXI interface (ARLOCK and AWLOCK outputs are tied low)

2.3 Core Version

This handbook is for CorePCIe_AHBLtoAXI version 2.0.

2.4 Supported Families

- IGLOO2
- SmartFusion2
- RTG4

2.5 Device Utilization and Performance

Device utilization and performance data is provided in the following table for the supported device families. The data listed in this table is indicative only. The overall device utilization and performance of the core is system-dependent.

Table 1 • **Device Utilization and Performance**

Family (Device)	Utilization (Logic Elements)				Performance (MHz)
	Sequential (DFF)	Combinatorial (4LUT)	Total	%	HCLK Frequency
SmartFusion2 (M2S150) / IGLOO2 (M2GL150)	59	250	309	0.21	152
RTG4 (RT4G150)	59	247	306	0.20	146

Note: The data in this table is achieved using typical synthesis and layout settings. Frequency (in MHz) was set to 100, and speed grade was -1.

3 Core Description

The core translates incoming AHBL read/write operations into AXI read/write transactions, respectively. The AHBL 32-bit transactions will result in AXI3 64-bit compliant transactions.

3.1 Burst Support

The following table illustrates the burst type in AHBL and its corresponding translation into AXI.

Table 2 • **AHBL and AXI Burst Support**

HBURST	Burst Type	Description	ARLEN/AWLEN [3:0]	ARBURST/AWBURST [1:0]	Corresponding AXI Transactions
AHBL Side			AXI Side		
000	SINGLE	Single	0000 - Single	01 - INCR	1 single-beat transaction
001	INCR	Undefined length burst	0000 - Single	01 - INCR	1 or multiple single-beat transactions
010	WRAP4	4-beat WRAP	0000 - Single	01 - INCR	4 single-beat transactions
011	INCR4	4-beat INCR	0000 - Single	01 - INCR	4 single-beat transactions
100	WRAP8	8-beat WRAP	0000 - Single	01 - INCR	8 single-beat transactions
101	INCR8	8-beat INCR	0000 - Single	01 - INCR	8 single-beat transactions
110	WRAP16	16-beat WRAP	0000 - Single	01 - INCR	16 single-beat transactions
111	INCR16	16-beat INCR	0000 - Single	01 - INCR	16 single-beat transactions

Note: Irrespective of the AHBL burst type input, the core generates only single-beat (AxLEN = 4’b0000) increment burst (AxBURST = 2’b01) transactions on the AXI interface.

3.2 Control Signal Handling

For AHBL single transactions, the command is latched whenever HSEL and HREADY are high, and HTRANS is NON-SEQ.

For AHBL burst transactions, the core converts all AHBL burst transactions into AXI single-beat increment burst transactions. Hence, the command is latched whenever HSEL and HREADY are high, and HTRANS is NON-SEQ or SEQ.

Then the address and control signals are driven onto the appropriate AXI channel (Write/Read).

3.3 Write Transactions

AWVALID is asserted when AHBL write command (HSEL, HREADY, and HWRITE are high, and HTRANS is NON-SEQ or SEQ) is captured and kept high until the slave responds with AWREADY.

WVALID is asserted when AHBL write command (HSEL, HREADY, and HWRITE are high, and HTRANS is NON-SEQ or SEQ) is captured and kept high until the slave responds with WREADY for each cycle of valid data. The dependency of WVALID on HTRANS stems from the fact that AHBL Master can generate a busy cycle in between a burst transfer when the data is not available.

BREADY is asserted when AHBL write command (HSEL, HREADY, and HWRITE are high, and HTRANS is NON-SEQ or SEQ) is captured and de-asserted when BVALID is high (Response received from the AXI Slave channel).

HREADYOUT is de-asserted when AHBL write command (HSEL, HREADY, and HWRITE are high, and HTRANS is NON-SEQ or SEQ) is captured and asserted when BVALID is high (Response received from the AXI Slave channel).

3.4 Read Transactions

ARVALID is asserted when AHB read command (HSEL and HREADY are high, HTRANS is NON-SEQ or SEQ, and HWRITE is low) is captured and kept high until the slave responds with ARREADY.

RREADY is asserted AHBL read command (HSEL, HREADY, and HWRITE are high, and HTRANS is NON-SEQ or SEQ) is captured and is de-asserted when RVALID and RLAST are high (Read Data received from the AXI Slave channel). RREADY is de-asserted when HTRANS goes to Idle/Busy in-between a read burst transfer indicating that Master is not yet ready to receive the next beat of read data.

HREADYOUT is de-asserted when the read command (HSEL and HREADY are high, and HTRANS is NON-SEQ or SEQ) is captured and asserted when RVALID and RLAST are high (Read Data received from the AXI Slave channel).

3.5 Response Handling

In the case of AXI OKAY response, the core uses BRESP from the AXI side to generate HRESP for each beat of AHB transaction.

In the case of AXI SLVERR response, HRESP is asserted for two cycles, and HREADYOUT is asserted during the second cycle of the response.

4 Interface

4.1 Configuration Parameters

There are no configurable parameters available in CorePCIe_AHBLtoAXI.

4.2 Ports

All the input and output ports of the core are listed in the following table.

Table 3 • **Input and Output Signals**

Port	Width	Direction	Description
			Global Signal Ports (Clocks and Resets)
HCLK	1	Input	Core clock. All input signals are required to be clocked on rising edge of this clock. All the output signals are clocked on rising edge of this clock.
HRESETN	1	Input	Core reset signal - Active low reset signal. The reset input is required to be synchronous to clock HCLK.
			AHBL Interface Ports
HADDR	32	Input	AHBL address. 32-bit address on the AHB-Lite interface
HSEL	1	Input	AHBL slave select. When high, indicates the AHB to AXI bridge is selected.
HBURST	3	Input	Type of burst generated by the AHBL master Supported burst types: 3'b000: Single burst 3'b001: Incrementing burst of undefined length 3'b010: 4-beat wrapping burst 3'b011: 4-beat incrementing burst 3'b100: 8-beat wrapping burst 3'b101: 8-beat incrementing burst 3'b110: 16-beat wrapping burst 3'b111: 16-beat incrementing burst
HSIZE	2	Input	AHBL transfer size - Indicates the size of the AHBL transfer Supported transfer sizes: 2'b00: 8-bit (byte) transaction 2'b01: 16-bit (half word) transaction 2'b10: 32-bit (word) transaction
HTRANS	2	Input	AHBL transfer type - Indicates the transfer type of the current transaction: Supported transfer types: 2'b00: Idle 2'b01: Busy 2'b10: Non-Sequential 2'b11: Sequential
HWRITE	1	Input	AHBL write - When high, indicates that the current transfer is a write transfer. When low, indicates that the current transfer is a read transfer.
HWDATA	32	Input	AHBL write data - Write data from the AHB-Lite master to the AHBL to AXI bridge.
HREADY	1	Input	AHBL ready input - When HIGH, the HREADY signal indicates the slave that the previous transfer is complete.

Table 3 • **Input and Output Signals *(continued)***

Port	Width	Direction	Description
HREADYOUT	1	Output	AHBL ready output - When high for a write indicates the slave is ready to accept write data, and when high for a read indicates that read data is valid.
HRDATA	32	Output	AHBL read data - Read data from the AHBL slave to the AHBL master
HRESP	1	Output	AHBL response status - When driven low at the end of a transaction, indicates that the transaction has completed with errors. When driven high at the end of a transaction indicates that the transaction has completed successfully.
			AXI Interface Ports
			AXI Write Address Channel
AWID	4	Output	Write Address ID - The identification tag for the write address group of signals. This output is tied low always.
AWADDR	32	Output	Write address - Gives the address of the first transfer in a write transaction.
AWLEN	4	Output	Burst length - Denotes the number of transfers in a transaction. Supports generation of the following AXI burst lengths: 4’b0000: Single-beat transaction
AWSIZE	2	Output	Burst size - Indicates the size of each transfer in the transaction Supports generation of following AXI burst sizes: 2'b00: 8-bit (byte) transactions 2'b01: 16-bit (half word) transactions 2'b10: 32-bit (word) transactions
AWBURST	2	Output	Burst type - Signals the type of burst transfer performed. Supports generation of following AXI burst types: 2'b01: Incrementing address burst
AWLOCK	2	Output	Lock type This output is tied low always.
AWVALID	1	Output	Write address valid - Indicates that valid write address and control information are available: 1'b1: Address and control available 1'b0: Address and control not available
AWREADY	1	Input	Write address ready - Indicates that the slave is ready to accept an address and associated control signals: 1'b1: Slave ready 1'b0: Slave busy
			AXI Write Data Channel
WID	4	Output	Write Data ID tag - The identification tag for the write data group of signals. The WID must match the AWID value of the write transaction. This output is tied low always.
WDATA	64	Output	Write data bus is 64-bits wide.
WSTRB	8	Output	Write strobes. - Indicates the byte lanes of the WDATA signal that contain valid write data. There is one write strobe for each 8 bits of the write data bus. WSTRB[n] corresponds to WDATA [(8×n)+7:(8×n)].

Table 3 • **Input and Output Signals *(continued)***

Port	Width	Direction	Description
WLAST	1	Output	Write last - Indicates that the current transfer is the last transfer in the write transaction.
WVALID	1	Output	Write valid - Indicates that valid write data and strobes are available: 1'b1: Write data and strobes available 1'b0: Write data and strobes unavailable
WREADY	1	Input	Write ready - Indicates that the slave will register the write data and strobes on the next HCLK rising edge, at which point the write data can be updated/removed. 1'b1: Slave ready 1'b0: Slave not ready
			AXI Write Response Channel
BID	4	Input	Response ID - The identification tag for the write response group of signals. The BID must match the AWID value of the write transaction to which the slave is responding. The signal is not used.
BRESP	2	Input	Write response - Indicates the status of the write transaction. 2'b00: OKAY 2'b01: EXOKAY 2'b10: SLVERR 2'b11: DECERR The LSB bit, BRESP[0] is unused. The MSB bit, BRESP[1] is mapped on to HRESP for AHBL write transactions.
BVALID	1	Input	Write response valid - Indicates that a valid write response is available: 1'b1: Write response available 1'b0: Write response not available
BREADY	1	Output	Response ready - Indicates that the AXI master will register the AXI slave write response on the next HCLK rising edge, at which point the slave write response can be removed. 1'b1: Master ready 1'b0: Master not ready
			AXI Read Address Channel
ARID	4	Output	Write Address ID - The identification tag for the read address group of signals. This output is tied low always.
ARADDR	32	Output	Read address - Gives the address of the first transfer in a read transaction
ARLEN	4	Output	Burst length - Denotes the number of transfers in a transaction. Supports generation of the following AXI burst lengths: 4’b0000: Single-beat transaction
ARSIZE	2	Output	Burst size - Indicates the size of each transfer in the transaction Supports generation of following AXI burst sizes: 2'b00: 8-bit (byte) transactions 2'b01: 16-bit (half word) transactions 2'b10: 32-bit (word) transactions

Table 3 • **Input and Output Signals *(continued)***

Port	Width	Direction	Description
ARBURST	2	Output	Burst type - Signals the type of burst transfer performed. Supports generation of following AXI burst types: 2'b01: Incrementing address burst
ARLOCK	2	Output	Lock type This output is tied low always.
ARVALID	1	Output	Read address valid - Indicates that valid read address and control information are available: 1'b1: Address and control available 1'b0: Address and control not available
ARREADY	1	Input	Read address ready - Indicates that the slave is ready to accept an address and associated control signals: 1'b1: Slave ready 1'b0: Slave busy
			AXI Read Data Channel
RID	4	Input	Read Data ID tag - This signal is the ID tag of the read data group of signals. The RID must match the ARID value of the read transaction to which slave is responding. The signal is not used.
RDATA	64	Input	Read data bus is 64-bits wide
RRESP	2	Input	Read response - Indicates the status of the read transaction. 2'b00: OKAY 2'b01: EXOKAY 2'b10: SLVERR 2'b11: DECERR The LSB bit, BRESP[0] is unused. The MSB bit, BRESP[1] is mapped on to HRESP for AHBL read transactions.
RLAST	1	Input	Read Last - Indicates that the current transfer is the last transfer in the read transaction.
RVALID	1	Input	Read Valid - Indicates to the AXI master that CorePCIe_AHBLtoAXI is presenting valid read data. 1'b1: Read data available 1'b0: Read data not available
RREADY	1	Output	Read ready - Indicates that the AXI master will register the read data on the next HCLK rising edge, at which point the read data can be updated/removed. 1'b1: Slave ready 1'b0: Slave not ready

5 Tool Flow

5.1 License

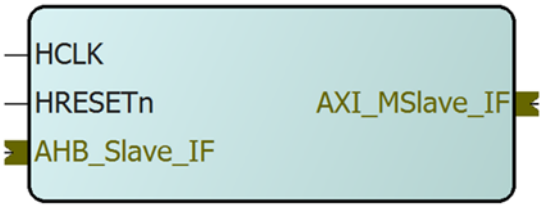
The core is freely available. Complete clear RTL source code is provided.

5.2 Using core in Libero SmartDesign

The core is pre-installed in the SmartDesign IP deployment design environment or can be downloaded from the online repository.

An example of the core instantiated in Libero SmartDesign is shown in the following figure.

Figure 2 • **Core Instance View in SmartDesign**



The core does not have a configurator as there are no configurable parameters.

For more information on using the SmartDesign to instantiate and generate cores, refer to the *Using DirectCore in Libero® SoC User Guide*.

5.3 Simulation Flows

Testbench is not provided along with the core.

5.4 Synthesis in Libero

To run synthesis on the core, set the design root to the IP component instance and run the **Synthesis** tool from the Libero **Design Flow** pane.

5.5 Place-and-Route in Libero

After the design is synthesized, run the **Place-And-Route** tool from the Libero **Design Flow** pane.

6 Clocking and Reset

The core uses a single synchronous clock (HCLK) to clock both the AHB and AXI sides of the interface. That is, the AHB and AXI are synchronous to each other. Likewise, a single active low asynchronous reset HRESETN is used to reset the core. The reset de-assertion should be synchronized to HCLK.

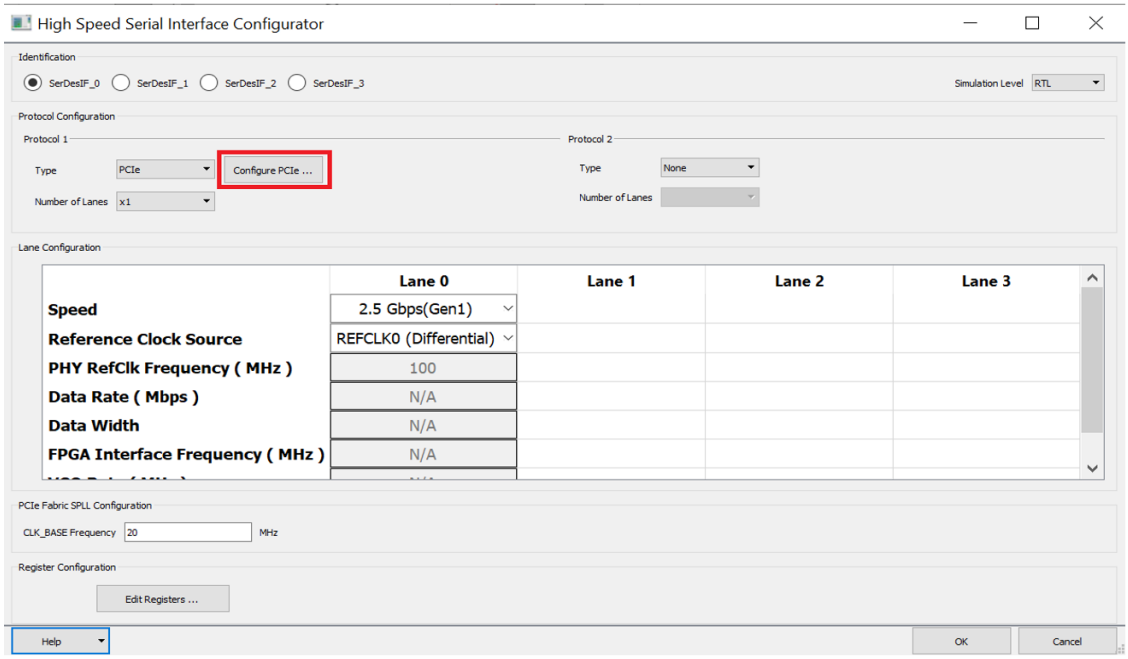
7 Integration with SERDES in Libero SmartDesign

CorePCIe_AHBLtoAXI is recommended to be used with SmartFusion2, IGLOO2, and RTG4 PCIe AXI Slave to connect to an AHBL master or AHBL sub-system.

This section provides the information related to the SERDES configuration selection required to connect the SERDES AXI interface to the AXI interface of CorePCIe_AHBLtoAXI.

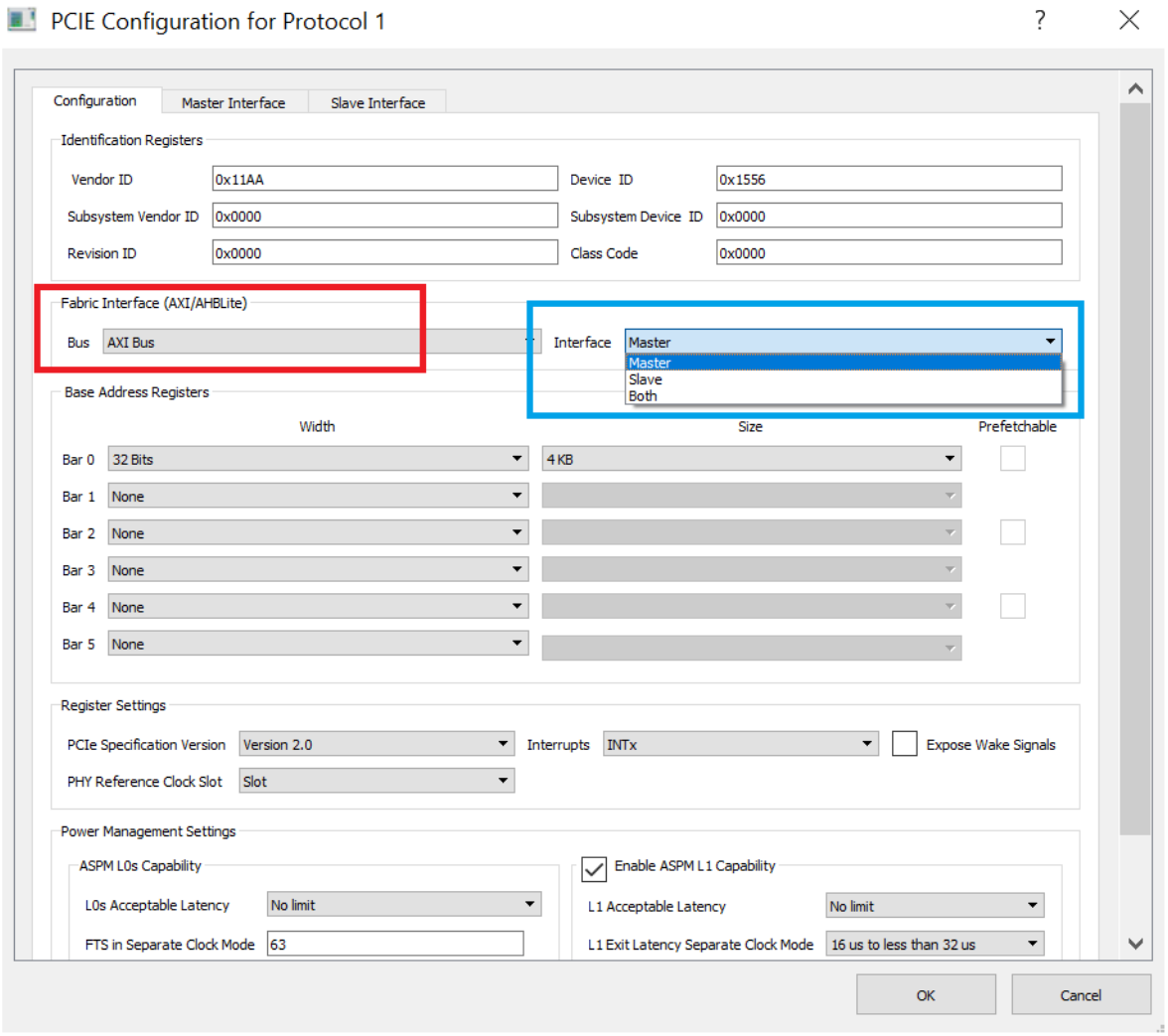
1. In High Speed Serial Interface Configurator, under Protocol Configuration, click **Configure PCIe,** as shown in the following figure.

Figure 3 • **High Speed Serial Interface Configurator**



2. Select the **AXI Bus** option under Fabric Interface (AXI/AHBLite) in the **Configuration** tab, as shown in the following figure.
3. To make the AXI Slave interface available, under Interface, select **Slave** or **Both**, as shown in the following figure.

Figure 4 • **PCIE Configuration for Protocol 1**



The following figure shows the recommended AHB and AXI bus connections for CorePCIe_AHBLtoAXI. The core is connected to the CoreAHBLite on AHBL interface and SERDES on the AXI interface.

- The AHB_Slave_IF of CorePCIe_AHBLtoAXI must be connected to AHBmslave**x** (**x** represents the slave interface selected on CoreAHBLite) interface of CoreAHBLite.
- The AXI_MSlave_IF of CorePCIe_AHBLtoAXI must be connected to the AXI_SLAVE interface of SERDES.

Figure 5 • **AHB and AXI Bus Connections**