HB0945
Handbook
CorePCIe_AXItoAHBL v2.0

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

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.

50200945. 1.0 4/21

Contents

Figures

Tables

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_AXItoAHBL v2.0.

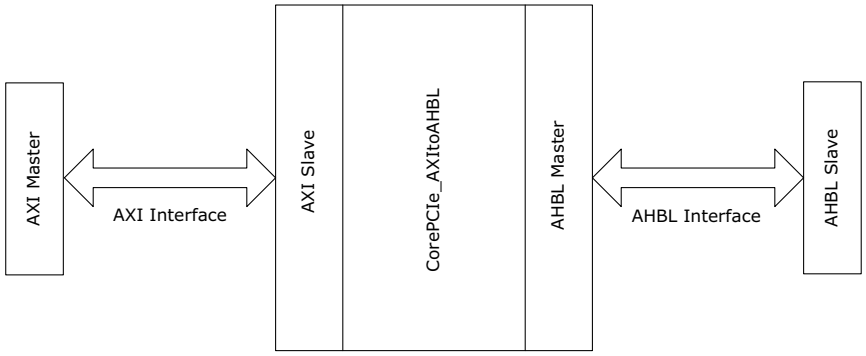
2 Introduction

2.1 Overview

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

The IP core is recommended to be used with G4 PCIe AXI Master to connect to an AHBL slave or AHBL sub-system.

Figure 1 • **CorePCIe_AXItoAHBL Bridge Block Diagram**



2.2 Features

CorePCIe_AXItoAHBL has the following features:

- Compliant to AMBA3 AXI and AMBA3 AHB-Lite specifications
- Provides an interface between the AXI domain and the AHBL domain
- AXI and AHBL domains are synchronous (common clock for both AXI and AHBL interfaces)
- Converts 64-bit AXI write/read transactions into 32-bit AHBL write/read transactions respectively
- Supports only 64-bit AXI transactions
- Supports only INCR type AXI burst transactions
- Generates only SINGLE AHB transfers using undefined length burst
- Generates only IDLE and NON-SEQUENTIAL AHB transfer type
- Controls the bus responses from AHBL-Slave to AXI-Master

The following features are not supported in CorePCIe_AXItoAHBL:

- Locked and exclusive transactions that is AxLOCK
- Auxiliary signals that is AxCACHE and AxPROT
- Write interleaving transactions
- FIXED and WRAP burst transactions
- Does not generate HMASTLOCK and HPROT

2.3 Core Version

This handbook is for CorePCIe_AXItoAHBL version 2.0.

2.4 Supported Families

- IGLOO®2
- SmartFusion®2
- 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**

	Utilization (Logic Elements)				Performance (MHz)
Family (Device)	**Sequential (DFF)**	**Combinatorial (4LUT)**	**Total**	**Percentage (%)**	**ACLK Frequency**
SmartFusion2 (M2S150)/IGLOO2 (M2GL150)	313	297	610	0.41	211
RTG4 (RT4G150)	313	269	582	0.39	168

Note: The data in this table is achieved using typical synthesis and layout settings. Frequency (in MHz) was set to 166, and speed grade was -1.

3 Core Description

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

3.1 Outstanding Transactions

The core only accepts a single AXI read or write transaction at a time.

3.2 Arbitration

The core will prioritize a read over a write operation when idle. After every read, if a write is pending, it will complete before the next read starts that is read and write will alternate.

If a read and write is pending to the same AXI 4K aligned address range, then the write will be prioritized over the read to ensure the data reaches the AHB system before a read to the same address range.

3.3 Write Channel AXI IDs

The core will store the AXI ID during the write address transaction to be used during the write response transaction. The core does not check the WID during the write data cycles.

3.4 Read Channel AXI IDS

The core will store the AXI ID during the read address transaction to be used during the read response cycles.

3.5 Burst Support (AXI and AHBL)

The core will convert AXI burst transfers to NONSEQ AHB transfers. This means it will only generate HTRANS cycles off IDLE and NONSEQ. It will not generate BUSY or SEQ transactions. All reads will be 32-bit. For write, AHB will use 32, 16, and 8-bit depending on the AXI WSTRBS. The HBURST output will be hard wired to 3'b001 Incrementing burst of undefined length. The core uses ARLEN to determine how many reads to complete, setting RLAST on the last data beat. For write, the core ignores AWLEN and used WLAST to detect the end of a burst.

3.6 Write Strobes

For AXI write transactions the core looks at the byte strobes and breaks the write into multiple AHB writes with appropriate HADDR and HSIZE settings as AHB does not support write strobes. The core supports all 256 possible WSTRB values.

3.7 AXI Response Errors

During read transactions, an AHB response error will be converted to an AXI response error for the associated AXI 64-bit read. The error response will be set with read data if either of the two 32-bit AHB reads indicates a response error. For write transactions, any AHB response errors will be accumulated and reported at the end of the transaction on the AXI write response channel.

4 Interface

4.1 Configuration Parameters

There are no configurable parameters available in CorePCIe_AXItoAHBL.

4.2 Ports

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

Table 2 • **Input and Output Signals**

Port	Width	Direction	Description
Global Signal Ports (Clocks and Resets)			
ACLK	1	Input	Core clock. All input signals are required to be clocked on the rising edge of this clock. All the output signals are clocked on the rising edge of this clock.
ARESETN	1	Input	Core reset signal - Active low reset signal. The reset input is required to be synchronous to clock ACLK.
AXI Interface Ports			
AXI Write Address Channel			
AWVALID	1	Input	Write address valid - Indicates that valid write address and control information are available: 1: address and control available 0: address and control not available
AWREADY	1	Output	Write address ready - Indicates that the slave is ready to accept an address and associated control signals: 1: slave ready 0: slave busy
AWID	4	Input	Write Address ID - The identification tag for the write address group of signals.
AWADDR	32	Input	Write address - Gives the address of the first transfer in a write transaction. The associated control signals are used to determine the addresses of the remaining transfers in the burst. The lower 3-bits of this signal is not used.
AWLEN	4	Input	Burst length - Denotes the number of transfers in a transaction. This input is not used, the core uses WLAST to terminate a write sequence.
AWSIZE	2	Input	Burst size - Indicates the size of each transfer in the transaction. Supported AXI burst sizes: 2'b11: 64-bit (double word) transactions. This input is not used. However, the RTL code includes an assertion to trigger when AXI master issues an unsupported size transaction.

Table 2 • **Input and Output Signals *(continued)***

Port	Width	Direction	Description
AWBURST	2	Input	Burst type - Signals the type of burst transfer performed. Supported AXI burst types: 2'b01: Incrementing address burst. This input is not used. However, the RTL code includes an assertion to trigger when AXI master issues an unsupported burst transaction.
AXI Write Data Channel			
WVALID	1	Input	Write valid - Indicates that valid write data and strobes are available: 1: write data and strobes available 0: write data and strobes unavailable
WREADY	1	Output	Write ready - Indicates that the slave will register the write data and strobes on the next ACLK rising edge, at which point the write data can be updated/removed. 1: slave ready 0: slave not ready
WID	4	Input	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. The signal is not used.
WDATA	64	Input	Write data bus is 64-bits wide.
WSTRB	8	Input	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)].
WLAST	1	Input	Write last - Indicates that the current transfer is the last transfer in the write transaction.
AXI Write Response Channel			
BVALID	1	Output	Write response valid - Indicates that a valid write response is available: 1: write response available 0: write response not available
BREADY	1	Input	Response ready - Indicates that the AXI master will register the AXI slave write response on the next ACLK rising edge, at which point the slave write response can be removed. 1: master ready 0: master not ready
BID	4	Output	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.
BRESP	2	Output	Write response - Indicates the status of the write transaction. Responses provided by the IP: 2'b00: OKAY 2'b10: SLVERR

Table 2 • **Input and Output Signals *(continued)***

Port	Width	Direction	Description
AXI Read Address Channel			
ARVALID	1	Input	Read address valid - Indicates that valid read address and control information are available: 1: address and control available 0: address and control not available
ARREADY	1	Output	Read address ready - Indicates that the slave is ready to accept an address and associated control signals: 1: slave ready 0: slave busy
ARID	4	Input	Read Address ID - This signal is the identification tag for the read address group of signals.
ARADDR	32	Input	Read address - Gives the address of the first transfer in a read transaction. The associated control signals are used to determine the addresses of the remaining transfers in the burst. The lower 3-bits of this signal are not used.
ARLEN	4	Input	Burst length - Denotes the number of transfers in a transaction.
ARSIZE	2	Input	Burst size - Indicates the size of each transfer in the transaction. Supported AXI burst sizes: 2'b11: 64-bit (double word) transactions This input is not used. However, the RTL code includes an assertion to trigger when AXI master issues an unsupported size transaction.
ARBURST	2	Input	Burst type - Signals the type of burst transfer performed. Supported AXI burst types: 2'b01: Incrementing address burst This input is not used. However, the RTL code includes an assertion to trigger when AXI master issues an unsupported burst transaction.
AXI Read Data Channel			
RVALID	1	Output	Read Valid - Indicates to the AXI master that CorePCIe_AXItoAHBL is presenting valid read data. 1: read data available 0: read data not available
RREADY	1	Input	Read ready - Indicates that the AXI master will register the read data on the next ACLK rising edge, at which point the read data can be updated/removed. 1: slave ready 0: slave not ready
RID	4	Output	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 it is responding.
RDATA	64	Output	Read data bus is 64-bits wide

Table 2 • **Input and Output Signals *(continued)***

Port	Width	Direction	Description
RRESP	2	Output	Read response - Indicates the status of the read transaction. Responses provided by the IP: 2'b00: OKAY 2'b10: SLVERR
RLAST	1	Output	Read Last - Indicates that the current transfer is the last transfer in the read transaction.
AHBL Interface Ports			
HADDR	32	Output	AHBL address. 32-bit address on the AHB-Lite interface
HSEL	1	Output	AHBL Slave Select. Not required on AHB master interface but provided to aid direct connection to single AHB slave. This signal is asserted when HTRANS is active.
HWRITE	1	Output	AHBL write - When high, indicates that the current transfer is a write transfer. When low, indicates that the current transfer is a read transfer.
HBURST	3	Output	Type of burst generated by the AHBL master Supported burst types: 3'b001: Incrementing burst of undefined length
HSIZE	2	Output	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	Output	AHBL transfer type - Indicates the transfer type of the current transaction: Supported transfer types: 2'b00: IDLE 2'b10: NON-SEQUENTIAL
HWDATA	32	Output	AHBL write data - Write data from the AHB-Lite master to the AHB-Lite slave
HREADYIN	1	Input	AHBL ready input - 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	Input	AHBL read data - Read data from the AHBL slave to the AHBL master
HRESP	1	Input	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.

5 Clocking and Reset

The core uses a single synchronous clock (ACLK) to clock both the AXI and AHB sides of the interface. All registers are rising edge clocked that is AXI and AHB are synchronous to each other. Likewise, a single active low asynchronous reset ARESETN is used to reset the core. The reset de-assertion should be synchronized to ACLK.

6 Tool Flow

6.1 License

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

6.2 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*.

6.3 Simulation Flows

Test bench is not provided along with the core.

6.4 Timing Constraints

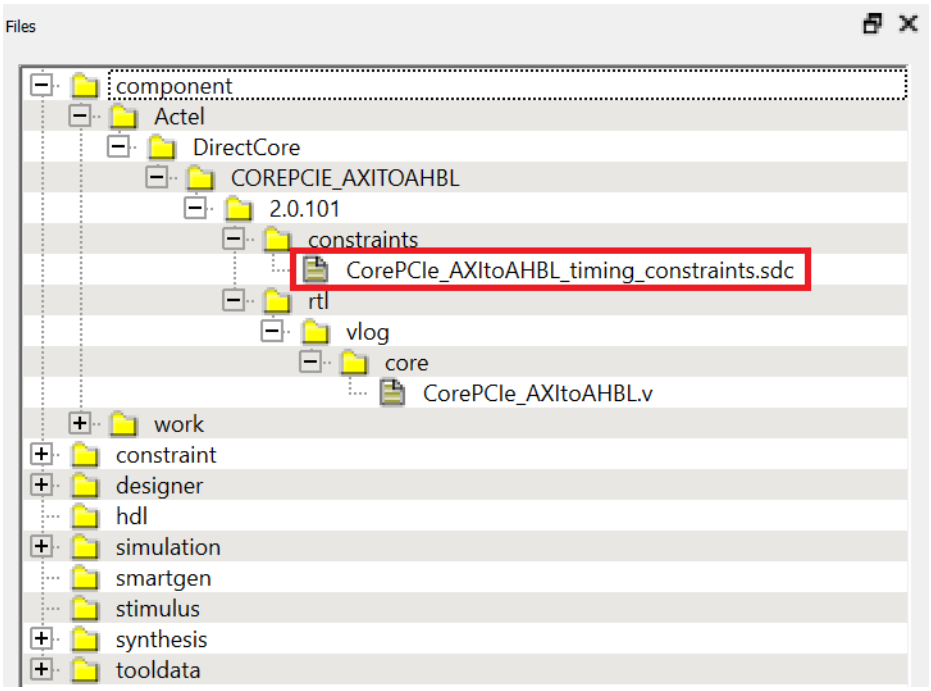
The CorePCIe_AXItoAHBL_timing_constraints.sdc file contains timing constraints recommended to use when the core is connected to SmartFusion2, IGLOO2, or RTG4 SERDES PCIe AXI Master.

This sdc file is packaged with the IP Core files, which will be available after the core is generated in the Libero project. These constraints can be used while running Synthesis, Place-and-Route, and Verify Timing tools. The constraints provided in this sdc file are listed for user reference as follows:

- false path constraints
  - set_false_path -through [get_pins {*/AWID*}]
  - set_false_path -through [get_pins {*/ARID*}]

The following figure shows the path where the timing constraint file is generated in the Libero project.

Figure 3 • **Timing Constraints Path Libero Project**



6.5 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.

6.6 Place-and-Route in Libero

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

7 Performance

COREPCIE_AXITOAHBL converts 64-bit AXI transactions to a 32-bit AHB transaction. The throughput will be affected by the HREADY assertion by the user design. The following table provides the number of clock cycles from AWVALID/ARVALID assertion to the BVALID/RLAST cycles for AXI transfers of 1, 2, and 4 beats, with 0, 1, and 2 waits states being inserted by the AHB system.

Table 3 • **Throughput values for AXI transfers of 1, 2, and 4 beats, with 0, 1, and 2 AHB waits states**

AXI Burst Length + 1	Number of AHB Transfers	AHB Wait States					
		0		1		2	
		Write	Read	Write	Read	Write	Read
1	2	8	6	10	8	12	10
2	4	12	10	15	14	18	18
4	8	20	18	25	26	30	34

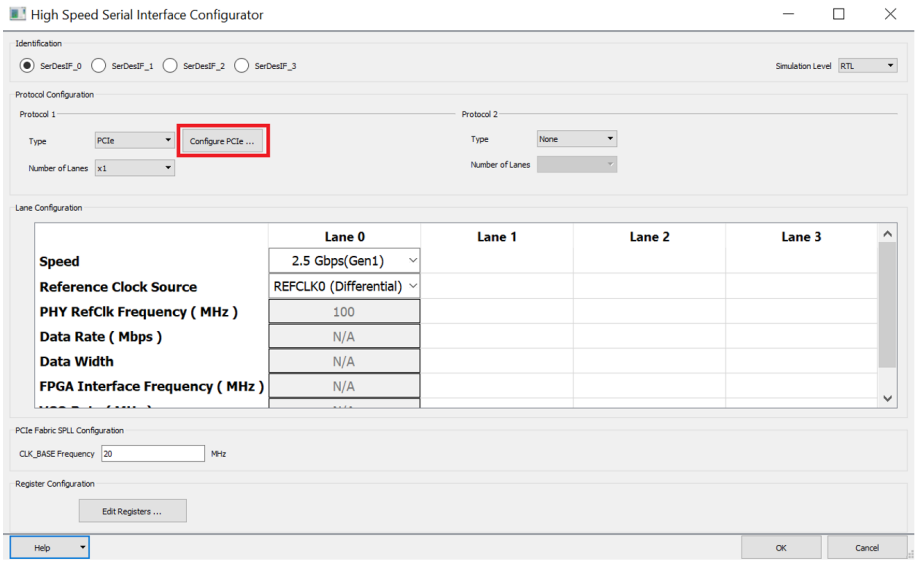
8 Integration with SERDES in Libero SmartDesign

CorePCIe_AXItoAHBL is recommended to be used with G4 PCIe AXI Master to connect to an AHBL slave 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_AXItoAHBL.

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

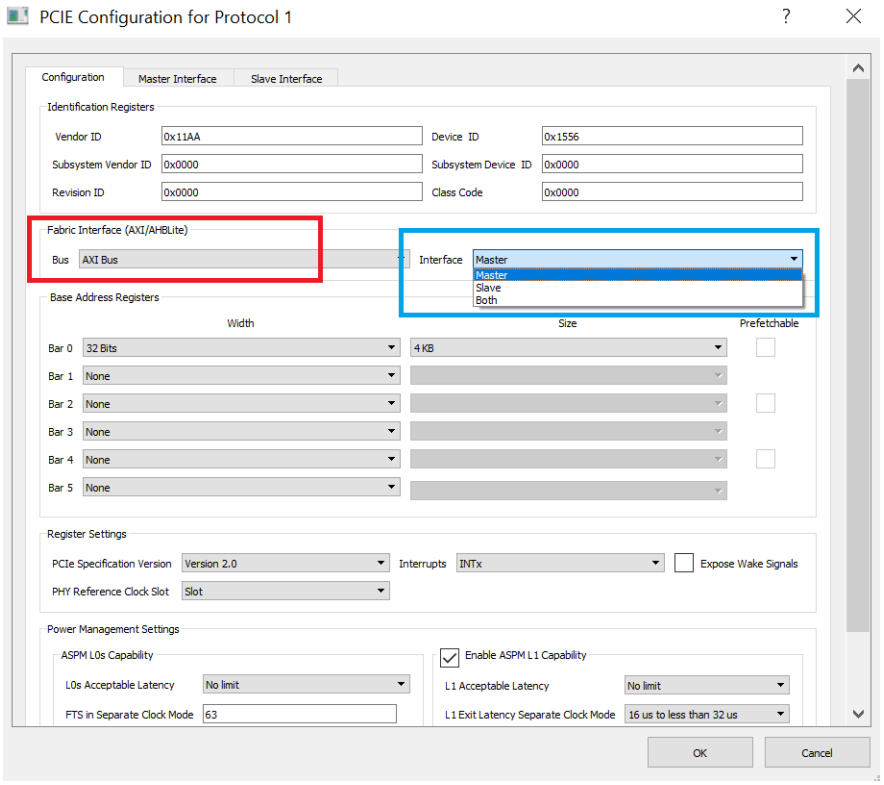
Figure 4 • **High Speed Serial Interface Configurator**



The PCIE Configuration for Protocol 1 window pops up.

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

Figure 5 • **PCIE Configuration for Protocol 1**



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

- The AXI_MMaster_IF of CorePCIe_AXItoAHBL must be connected to the AXI_MASTER interface of SERDES.
- The AHB_Master_IF of CorePCIe_AXItoAHBL must be connected to AHB_mmasterx (x represents the master interface selected on CoreAHBLite) interface of CoreAHBLite.

Figure 6 • **AXI and AHB Bus Connections**