CoreAHBLtoAXI v2.1

Handbook

Microsemi

Table of Contents

Introduction

General Description

The CoreAHBLtoAXI IP core is an advanced high-performance bus lite (AHB-Lite) slave and an advanced extensible interface (AXI) master that provides an interface (bridge) between the AHB domain and AXI domain. The CoreAHBLtoAXI IP core allows an AHBL bus system to be connected to an AXI bus. The AHBL to AHB-Lite Bridge enables the AHB master to communicate with the AHB slave. The AHBL slave interface and AXI master interface allows the AHB bus and AXI bus to access the read/write buffer memory.

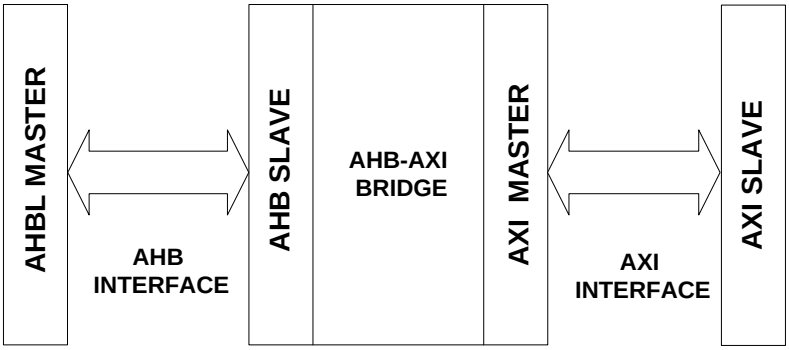

Figure 1 CoreAHBLtoAXI Bridge Block Diagram

Core Version

This Handbook applies to CoreAHBLtoAXI version 2.1.

Supported Families

- SmartFusion®2
- IGLOO®2

Utilization and Performance

Utilization and performance data is listed in Table 1 for the SmartFusion2 (M2S150S) device family. 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	Async Clk	Undefined Burst Length	AXI_AWIDTH	AXI_DWIDTH	AHB_AWIDTH	AHB_DWIDTH	Logic Elements				Frequency (Mhz)
							Sequential	Combinatorial	Total	%	
SmartFusion2	0	0	32	64	32	32	1,202	1,413	2,615	0.89	119.1
SmartFusion2	1	0	32	64	32	32	1,202	1,413	2,615	0.89	112.1
SmartFusion2	0	1	32	64	32	32	1,194	1,393	2,587	0.88	129.1
SmartFusion2	1	1	32	64	32	32	1,194	1,393	2,587	0.88	129.1

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.

Functional Block Description

The CoreAHBLtoAXI Bridge appears as a slave on the AHBL and is a master on the AXI. Read and write transactions on the AHBL are converted into corresponding transfers on the AXI.

The HCLK and ACLK clocks are configurable to synchronous or asynchronous depending on the parameter/generic. This block performs the cross-clock-domain control logic when the AHB clock and AXI clock are asynchronous.

Following are the four major functional blocks of CoreAHBLtoAXI:

- Write memory buffer
- Read memory buffer
- Write/read AHBLite slave controller
- Write/read AXI master Controller

A basic block diagram of the design for CoreAHBLtoAXI is shown in Figure 2.

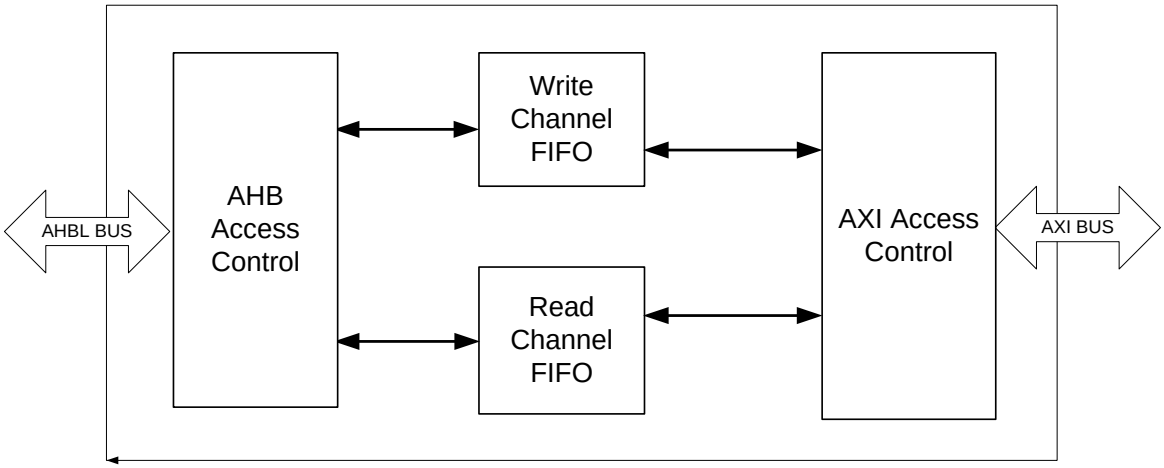

Figure 2 CoreAHBLtoAXI Block Diagram

AHB Access Control

The AHB Access Control block is the AHBL Slave interface of the bridge. This block contains Write/Read finite state machine (FSM) which converts the AHB Write/Read transactions into corresponding AXI write or read transactions. The AHB master drives write transaction when the HREADY is asserted. It triggers the AHB Access Control FSM. During AHB Write transaction, the FSM asserts write enable to the Write Channel FIFO. It also drives the address, burst length, and size control information to the AXI access control block.

This block, depending on the AHB burst length and size, calculates the corresponding AXI transactions of the bursts, size, and length on the AXI side. It also provides the write strobe information to the AXI access control block. The Write Response from the AXI slave is latched and sent to the AHB Master.

AHB Access Control block also generates the read enable to the Read Channel FIFO. The read data received from the Read Channel FIFO is driven to the AHB master through the AHB Access Control.

Write Channel FIFO Control

The Write Channel Control FIFO block is 16 deep x 64 wide asynchronous FIFO.

It stores the write data from the AHB Access Control in the Write Channel FIFO. The AXI Access Control block generates the read enable to the FIFO, and it reads the data when the write to the Write Channel FIFO is completed.

Read Channel FIFO Control

The Read Channel Control FIFO block is 16 deep x 64 wide asynchronous FIFO. It stores the write data from the AXI Access Control. The AHB Access Control block generates the read enable to the FIFO and reads the data.

AXI Access Control logic

The AXI Access Control block is the AXI master interface of the bridge. This block generates the AXI write/read transactions on the AXI bus when the AWREADY and WREADY are asserted. The AXI start address is calculated from the AHB start address based on its size and burst.

The block controls the read enable to the Write Channel FIFO and the write enable to the Read Channel FIFO. For Write transactions on the AXI slave, the write enable is asserted to the Write Channel FIFO. Similarly, for Read transactions, the read enable is asserted to the Read Channel FIFO. The data read from the AXI slave through the AXI master interface of the CoreAHBLtoAXI bridge is stored in the Read Channel FIFO.

Clock Domains

The CoreAHBLtoAXI Bridge consists of two clock domains: the AXI clock and the AHB clock. The AXI Access Control logic block works on AXI clock and the AHB Access Control logic block works on the AHB clock. The synchronization is achieved through the use of asynchronous FIFO's (Write and Read Channel FIFO).

Licensing

CoreAHBLtoAXI requires a register transfer level (RTL) license to be used and instantiated.

RTL

Complete RTL source code is provided for the core and testbenches.

SmartDesign

CoreAHBLtoAXI is preinstalled in the SmartDesign IP Deployment design environment. An example instantiated view is shown in Figure 3. The core c

an be configured using the configuration GUI within the SmartDesign, as shown in Figure 4.

For information on using the SmartDesign to instantiate and generate cores, refer to the *Using DirectCore in Libero® System-on-Chip (SoC) User Guide* or consult the *Libero SoC online help*.

Figure 3 SmartDesign CoreAHBLtoAXI Instance View

Figure 4 SmartDesign CoreAHBLtoAXI Configuration Window

Simulation Flows

The User Testbench for CoreAHBLtoAXI is included in all releases.

To run simulations, select the User Testbench flow within the SmartDesign and click **Save and Generate on the Generate pane**. The User Testbench is selected through the Core Testbench Configuration GUI.

When the SmartDesign generates the Libero SoC project, it installs the User Testbench files.

To run the User Testbench, set the design root to the CoreAHBLtoAXI instantiation in the Libero SoC design hierarchy pane and click **Simulation** in the Libero SoC design flow window. This invokes ModelSim® and automatically runs the simulation.

User Testbench

Figure 5 shows an example of User Testbench is included with CoreAHBLtoAXI.

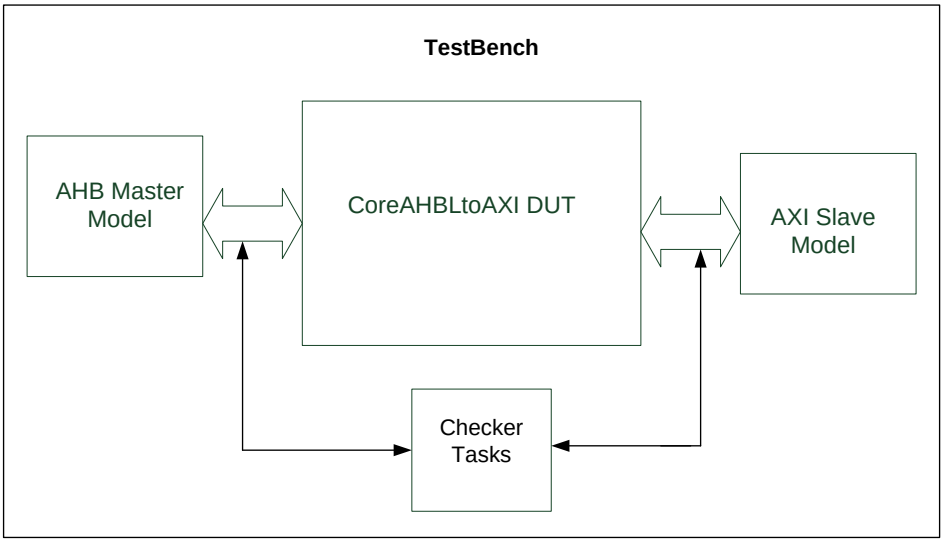

Figure 5 User Testbench

As shown in Figure 5, the User Testbench instantiates a Microsemi® DirectCore CoreAHBLtoAXI design under test (DUT). The CoreAHBLtoAXI Testbench environment consists of the AHB master model, AXI slave model, and the checker model. The AHB master model drives the write/read AXI transactions to the DUT. These transactions are converted by the DUT in corresponding AXI write/read transactions. The checker model checks and determines whether or not the transaction is successful and displays the result.

Synthesis in Libero SoC

Click **Synthesis** in Libero SoC. The synthesis window appears and displays the Synplicity® project. Set **Synplicity** to use the Verilog 2001 standard, if Verilog is being used. To run synthesis, select **Run**.

Place-and-Route in Libero SoC

Click **Layout** in the Libero SoC to invoke **Designer**. CoreAHBLtoAXI does not require any special place-and-route settings.

Signal Descriptions for CoreAHBLtoAXI

Signal descriptions for CoreAHBLtoAXI are defined in Table 2.

Table 2 CoreAHBLtoAXI I/O Signals

Port Name	Width	Direction	Description
AHBL Slave Interface Ports			
HCLK	1	In	AHB clock. All the AHB signals inside the block are clocked on the rising edge.
HRESETn	1	In	AHB Reset. The signal is active low. Asynchronous assertion and synchronous de-assertion. This is used to reset AHB registers in the block.
HSEL	1	In	AHBL slave select – When asserted this AHBL-to-AXI Bridge is the currently selected AHBL slave.
HADDR	AHB_AWIDTH	In	AHBL address – 32 bit address on the AHBL interface
HWRITE	1	In	AHBL write – When high, it indicates that the current transaction is a write. When low, indicates that the current transaction is a read.
HREADY	1	In	AHBL ready input – Used to monitor the AHBL bus ready
HTRANS	2	In	AHBL transfer type – Indicates the transfer type of the current transaction b00: IDLE b01: BUSY b10: NONSEQUENTIAL b11: SEQUENTIAL
HSIZE	3	In	AHBL transfer size – Indicates the size of the current transfer (8/16/32/64 bit transactions only) bx00: 8 bit (byte) transaction bx01: 16 bit (half word) transaction bx10: 32 bit (word) transaction bx11: 64 bit(double word) transaction
HWDATA	AHB_DWIDTH	In	AHBL write data – Writes data from the AHBL master to the AXI slave
HBURST	3	In	AHBL Burst – Indicates the size and type of burst from the AHB master
HREADYOUT	1	Out	When HIGH, the HREADYOUT signal indicates that a transfer has finished on the bus. This signal can be driven LOW to extend a transfer.

Port Name	Width	Direction	Description
HRESP	1	Out	AHBL response status – When driven high at the end of a transaction indicates that the transaction is completed with errors. When driven low at the end of a transaction, indicates that the transaction is completed successfully.
HRDATA	AHB_DWIDTH	Out	AHBL read data – Read data from the AXI slave to the AHBL master
AXI Master Interface Ports			
Global Signal Ports (Clocks)			
ACLK	1	In	AXI clock– All the AXI signals inside the block are clocked on the rising edge.
ARESETn	1	In	AXI reset signal – The signal is active low. The signal is asynchronously asserted and synchronously de-asserted. This is used to reset all the AXI registers in the block.
AXI Write ADDRESS CHANNEL			
AWID	4	Out	Write Address ID. Is the identification tag for the write address group of signals
AWADDR	AXI_AWIDTH	Out	Write address – Gives the address of the first transfer in a write burst. The associated control signals are used to determine the addresses of the remaining transfers in the burst.
AWLEN	4	Out	Burst length – Gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
AWSIZE	3	Out	Burst size – Indicates the size of each transfer in the burst.
AWBURST	2	Out	Burst type – Coupled with the size information, details how the address for each transfer within the burst is calculated.
AWLOCK	2	Out	Lock type – Provides additional information about the atomic characteristics of the transfer.
AWVALID	1	Out	Write address valid – Indicates that the valid write address and control information are available: 1: address and control available 0: address and control not available
AWREADY	1	In	Write address ready – Indicates that the slave is ready to accept an address and associated control signals: 1: slave ready 0: slave not ready
AXI Write DATA CHANNEL			
WID	4	Out	Write Data ID tag – The Identification tag for the write data transfer. The WID must match the AWID value of the write transaction.
WDATA	AXI_DWIDTH	Out	Write data bus is 64 bits wide.

Port Name	Width	Direction	Description
WSTRB	8	Out	Write strobes – Indicates which byte lanes to update in the memory. There is one write strobe for each eight bits of the write data bus. WSTRB[n] corresponds to WDATA [(8 × n) + 7 :(8 × n)].
WLAST	1	Out	Write last- Indicates the last transfer in a write burst.
WVALID	1	Out	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	In	Write ready – Indicates that the slave can accept the write data: 1: slave ready 0: slave not ready
AXI Write RESPONSE CHANNEL			
BREADY	1	Out	Response ready – Indicates that the master can accept the response information. 1: master ready 0: master not ready
BID	4	In	Response ID – Is the Identification tag for the write response. The BID must match the AWID value of the write transaction to which the slave is responding.
BRESP	2	In	Write response – Indicates the status of the write transaction. The allowable responses are: 00: OKAY 01: EXOKAY 10” SLVERR DECERR is not supported.
BVALID	1	In	Write response valid – Indicates that a valid write response is available: 1: write response available 0: write response not available
AXI Read ADDRESS CHANNEL			
ARID	4	Out	Read Address ID – Is the identification tag for the read address group of signals.
ARADDR	AXI_AWIDTH	Out	Read address – Gives the initial address of a read burst transaction. Only the start address of the burst is provided.
ARLEN	4	Out	Burst length – Gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
ARSIZE	3	Out	Burst size – Indicates the size of each transfer in the burst
ARBURST	2	Out	Burst type – Coupled with the size information, details how the address for each transfer within the burst is calculated.
ARLOCK	2	Out	Lock Type – Provides additional information about the atomic characteristics of the transfer.
ARVALID	1	Out	Read address valid- When HIGH, indicates that the read address and control information is valid. 1: address and control valid 0: address and control not valid

Port Name	Width	Direction	Description
ARREADY	1	In	Read address ready- Indicates that the slave is ready to accept an address and associated control signals: 1: slave ready 0: slave not ready
AXI Read RESPONSE CHANNEL			
RREADY	1	Out	Read ready – Indicates that the master can accept the read data and response information: 1: master ready, 0: master not ready
RID	4	In	Read ID Tag – The ID tag of the read data group of signals. The RID value is generated by the slave and must match the ARID value of the read transaction to which it is responding.
RDATA	AXI_DWIDTH	In	Read data – Read data bus is 64 bits wide
RRESP	2	In	Read Response – Indicates the status of the read transfer. The allowable responses are: 00: OKAY 01: EXOKAY 10: SLVERR DECERR is not supported
RLAST	1	In	Read Last – Indicates the last transfer in a read burst.
RVALID	1	In	Read Valid – Indicates that the required read data is available and the read transfer can be completed. 1: read data available, 0: read data not available

Core Parameters

CoreAHBLtoAXI Configurable Options

There are a number of configurable options which are applied to CoreAHBLtoAXI as shown in Table 3. If a configuration other than the default is required, the configuration dialog box in the SmartDesign should be used to select appropriate values for the configurable options.

Table 3 CoreAHBLtoAXI Configuration Options

Name	Valid Range	Description
FAMILY	19	Must be set to the required FPGA family 19: SmartFusion2 24: IGLOO2
AHB_AWIDTH	32	A 32-bit System AHB address bus
AHB_DWIDTH	32 or 64	Write/Read data bus on AHB side
AXI_AWIDTH	32	A 32-bit System AXI address bus
AXI_DWIDTH	64	Write/Read data bus on AXI side
CLOCKS_ASYNC	0 or 1	Set to 1 when the AXI clock is asynchronous to the AHB clock. In general, this should be set whenever the two clocks are not on the same global network.
AHB_FREQ		Clock frequency (MHz) on AHB side
AXI_FREQ		Clock frequency (MHz) on the AXI side. It should be set with the below ratio of AHB_FREQ: AHB_FREQ **:** AXI_FREQ 1:1 1:2 1:4 1:8 1:16
UNDEF_BURST	0 or 1	Undefined length Increment bursts on the AHBL are converted into single burst transfer or INCR transactions of 64 bit size and Burst length UPTO 16. 0: SINGLE burst transfer (default) 1: INCR16 burst transfer (Not supported in this release)

Register Map and Descriptions

CoreAHBLtoAXI does not contain any registers.

Ordering Information

Ordering Codes

CoreAHBLtoAXI can be ordered through your local Sales Representative. It should be ordered using the following number scheme: CoreAHBLtoAXI-XX, where XX is listed in Table 4.

Table 4 Ordering Codes

XX	Description
RM	RTL for RTL source — multiple-use license

List of Changes

The following table lists critical changes that were made in each revision of the document.

Date	Change	Page
June 2014	Added a device family under Supported Families section	3
	Added Utilization and Performance section	3
February 2013	Initial Handbook version	N/A

Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

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**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Microsemi SoC Products Group Customer Support website for more information and support (http://www.microsemi.com/soc/support/search/default.aspx). Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on website.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/soc/.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1(949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense and security, aerospace, 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; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.

50200396-1/06.14