HB0206
Handbook
CoreAHBtoAPB3 v3.2

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

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.

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 4.0

Added PolarFire® SoC support.

1.2 Revision 3.0

Added RTG4™ and IGLOO®2 support.

1.3 Revision 2.0

Updated to correspond with v4.1 of CoreAPB3.

1.4 Revision 1.0

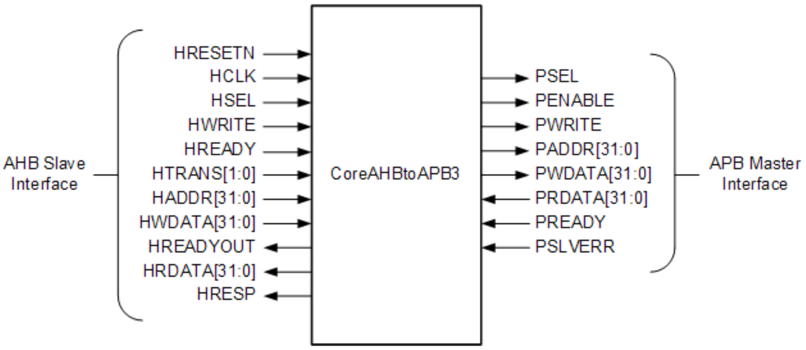
Updated to suit v3.0 of CoreAHBtoAPB3

2 Introduction

2.1 Core Overview

CoreAHBtoAPB3 is made up of an advanced high-performance bus (AHB™) slave and advanced microcontroller bus architecture (AMBA®) 3 advanced peripheral bus (APB) master. The CoreAHBtoAPB3 works as a bridge in between the AHB and the APB domains. CoreAHBtoAPB3 interfaces with CoreAHB or CoreAHBLite through its AHB interface and with CoreAPB3 through its APB interface. Figure 1, page 2 shows an overview of CoreAHBtoAPB3.

Figure 1 • **CoreAHBtoAPB3 Overview**



2.2 Key Features

- Bridges between AHB/AHB-Lite and APB buses
- Connects automatically to CoreAHB/CoreAHBLite and CoreAPB3 in SmartDesign when Auto Connect feature is used
- Compliant with AMBA 3 APB

2.3 Supported Microsemi® FPGA Families

The CoreAHBtoAPB3 v3.2 supports the following families:

- PolarFire® SoC
- PolarFire®
- SmartFusion®2
- SmartFusion®
- Microsemi Fusion®
- IGLOO®, IGLOO®e, IGLOO PLUS
- ProASIC®3, ProASIC®3E, ProASIC®3L
- ProASIC$^{PLUS®}$
- Axcelerator®
- RTAX-S
- IGLOO®2
- RTG4™

2.4 Core Version

This handbook supports CoreAHBtoAPB3 version 3.2.

2.5 Supported Interfaces

CoreAHBtoAPB3 supports an AHB or AHB-Lite slave interface connected to an AHB or AHB-Lite mirrored slave interface (as found on, for example, CoreAHB or CoreAHBLite), as well as an AMBA 3 APB master interface that connects to an AMBA 3 APB mirrored master interface (as found on, for example, CoreAPB3).

2.6 Utilization and Performance

Utilization and performance data for CoreAHBtoAPB3 is given in Table 1, page 3.

Table 1 • **CoreAHBtoAPB3 Device Utilization and Performance**

	Logic Elements			Utilization		Performance (MHz)
Family	**Sequential**	**Combinatorial**	**Total**	**Device**	**%**	
Fusion	126	75	201	M1AFS1500	1%	128
IGLOO	126	80	206	M1AGL1000V2	1%	75
ProASIC3	126	72	198	M1A3P600	1%	134
ProASIC	128	192	320	APA600	1%	80
Axcelerator	127	72	199	AX500	2%	145
RTAX-S	127	72	199	RTAX1000S	1%	97
SmartFusion2	141	63	204	M2S150T	0.14	250
IGLOO2	141	63	204	M2GL150T	0.14	250
RTG4	143	78	221	RT4G150	0.14	242
PolarFire	141	64	205	MPF200T	0.1	300
PolarFire SoC	141	64	205	MPFS250T	0.09	300

Note: Data in this table was gathered based on typical synthesis and layout settings.

3 Interface Description

3.1 Ports

The ports present on CoreAHBtoAPB3 are listed in Table 2, page 4.

Table 2 • **CoreAHBtoAPB3 Ports**

Port Name	Type	Description
HRESETN	Input	AHB reset, active low asynchronous reset
HCLK	Input	AHB clock signal
HSEL	Input	AHB slave select
HWRITE	Input	APB write indication
HREADY	Input	AHB ready input
HTRANS[1:0]	Input	AHB transfer type: 00: Idle 01: Busy 10: Non-sequential 11: Sequential
HADDR[31:0]	Input	AHB address bus
HWDATA[31:0]	Input	AHB write data
HREADYOUT	Output	AHB ready output
HRDATA[31:0]	Output	AHB read data
HRESP	Output	AHB transfer response: 00: Okay 01: Error 10: Retry 11: Split
PSEL	Output	APB select
PENABLE	Output	APB enable
PWRITE	Output	APB write indication
PADDR[31:0]	Output	APB address bus
PWDATA[31:0]	Output	APB write data
PRDATA[31:0]	Input	APB read data
PREADY	Input	APB ready indication
PSLVERR	Input	APB slave error indication

Note: All signals in this table are active high unless otherwise stated.

4 Design Description

CoreAHBtoAPB3 acts like a bridge between an AHB/AHB-Lite bus and an APB bus. Read and write transfers on the AHB bus are converted to corresponding transfers on the APB bus. High bandwidth peripherals, such as memory controllers, are typically connected to the AHB bus, whereas the APB bus is used for less demanding peripherals such as general purpose Input/Output (GPIO). Unlike the AHB bus, transfers on the APB bus are not pipelined.

4.1 AHB Interface Timing

Figure 2, page 5 and Figure 3, page 5 illustrate the AHB transfer timing. An AHB transfer can be described in terms of an address phase and a data phase. The data relevant to the address of interest follows in the clock cycle(s) after the address. The data phase can extend over a number of cycles if the HREADY signal is held low. Figure 2, page 5 shows a basic AHB transfer with no wait states as HREADY is high during the transfer. Figure 3, page 5 shows an AHB transfer with one wait state. HREADY is low for one cycle during the data phase in the waited transfer.

Figure 2 • **Basic AHB Transfer**

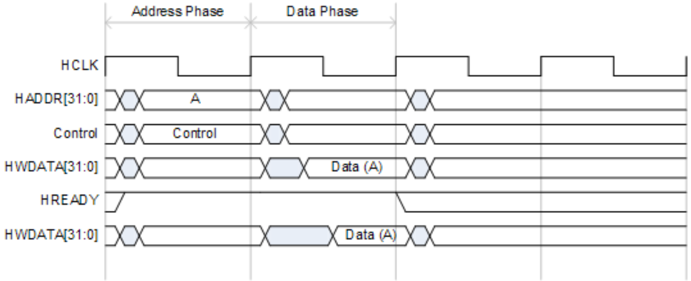
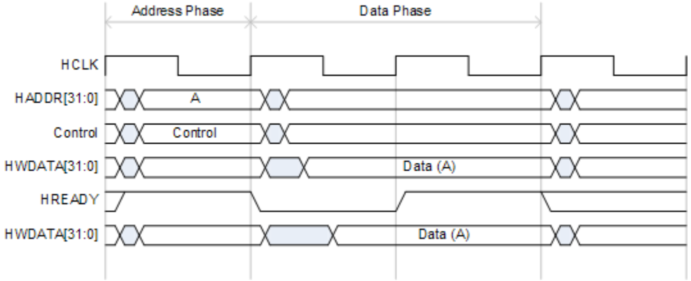


Figure 3 • **AHB Transfer with One Wait State**



4.2 APB Interface Timing

Figure 4, page 5 and Figure 5, page 6 show the timing of APB write and read transfers with no wait states. Figure 6, page 6 and Figure 7, page 6 show APB write and read transfers with one wait state.

Figure 4 • **APB Write Transfer, No Wait States**

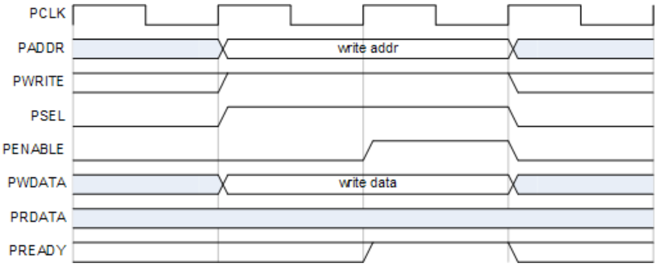


Figure 5 • **APB Read Transfer, No Wait States**

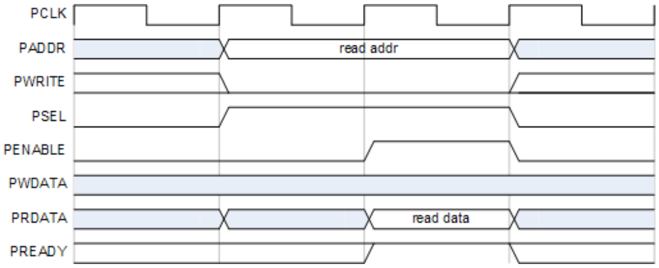


Figure 6 • **APB Write Transfer with One State**

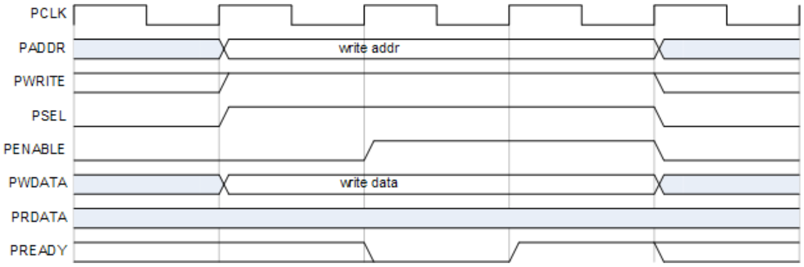
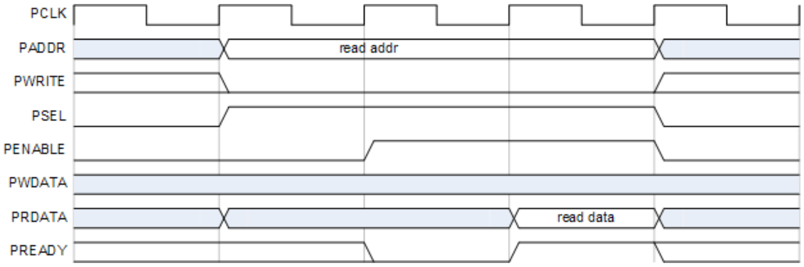


Figure 7 • **APB Read Transfer with One State**



5 Tool Flows

5.1 Licensing

CoreAHBtoAPB3 is licensed in two ways, Obfuscated or register transfer level (RTL).

5.1.1 Obfuscated

Complete RTL code is provided for the core, allowing the core to be instantiated with SmartDesign. Simulation, synthesis, and layout can be performed within Libero® System-on-Chip (SoC). The RTL code for the core is obfuscated and some of the testbench source files are not provided; instead, they are precompiled into the compiled simulation library.

5.1.2 RTL

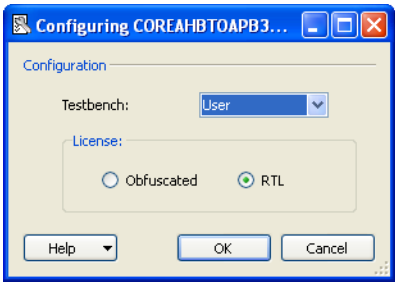
Complete RTL source code is provided for the core and testbenches.

5.2 SmartDesign

CoreAHBtoAPB3 is available for downloading to the SmartDesign IP Catalog through the Libero SoC web repository. For information on using SmartDesign to instantiate, configure, connect, and generate cores, refer to *Libero SoC documents page* and use the latest SmartDesign user guide.

The only configuration option available for CoreAHBtoAPB3 allows you to choose whether or not to create a testbench for CoreAHBtoAPB3 when generating a design including CoreAHBtoAPB3 from SmartDesign. Figure 8, page 7 shows the configuration GUI for CoreAHBtoAPB3. Figure 9, page 7 shows how the symbol for CoreAHBtoAPB3 appears on the SmartDesign canvas. The AHB and APB signals are grouped into interfaces that appear on the top and bottom of the symbol.

Figure 8 • **CoreAHBtoAPB3 Configuration GUI**



Configuring COREAHBTOAPB3...

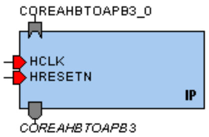
Configuration

Testbench: User

License:

○ Obfuscated ● RTL

Figure 9 • **CoreAHBtoAPB3 Symbol**



5.3 Simulation Flows

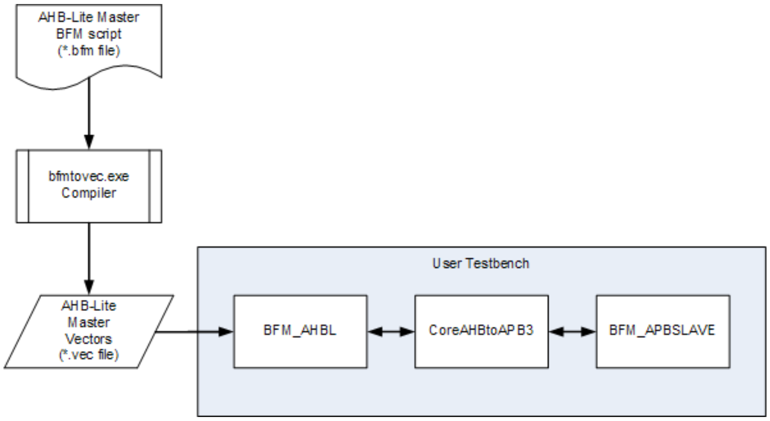
The user testbench for CoreAHBtoAPB3 is included in all releases of the core.

To run simulations, set the testbench configuration option for CoreAHBtoAPB3 to user before generating your design in SmartDesign. To run the user testbench, set the design root to the CoreAHBtoAPB3 instance in the Libero SoC Design Hierarchy pane and double-click the **Simulate** command in the Design Flow pane. You can also right-click on **Simulate** and choose, for example, “**Run**” or “**Open Interactively**”. ModelSim is invoked and the simulation runs automatically.

5.3.1 User Testbench

Figure 10, page 8 shows the simulation environment that includes instantiation of CoreAHBtoAPB3, an AHB-Lite master bus functional model (BFM), and an APB slave. The AHB-Lite BFM drives the testbench and is itself controlled by the commands in an automatically generated script file. This script file has a ".bfm" extension and is a plain text file that can be modified. Refer to the *DirectCore AMBA BFM User's Guide* for more information.

Figure 10 • **CoreAHBtoAPB3 User Testbench**



5.4 Synthesis in Libero SoC

Having set the design root to the CoreAHBtoAPB3 instance, double-click (or right click and choose an option) the **Synthesize** command in the Design Flow pane in Libero SoC. The Synplicity synthesis tool will be invoked. Set Synplicity to use the verilog 2001 standard if a verilog flow is being used. To run synthesis, click **Run**.

5.5 Place and Route in Libero SoC

Having set the design root appropriately and after running synthesis, use the **Compile** command followed by the Place and Route command in the Design Flow pane in Libero SoC to place and route the core.