AXI2AHB
(Beta Release)

Version 1.0

August 21, 2024

Copyright

Trademarks

Disclaimers

Contents

IP Summary

Introduction

The AXI2AHB bridge, also known as the AXI (Advanced eXtensible Interface) to AHB (Advanced High-Performance Bus) bridge, is a critical component in digital system design, especially when integrating different IP cores or subsystems that use different bus protocols. This bridge serves as an interface between the AHB bus and the AXI bus, which is widely adopted in modern designs. Block diagram of AXI2AHB4 IP is shown in figure below:

Features

- AXI interface is based on the AXI4-Full specification
- AHB interface is based on the AHB5 specification
- AHB slave 32 bit interface with Burst support
- AXI4 master 32 interface with burst support

Overview

AXI2AHB

In AXI2AHB4 bridge the AXI Interface is slave interface on AHB bus side. It accepts the control signals when a transfer is initiated by AXI master and generates AHB transactions based on the transfer progress on the AHB side.

The Control logic is the main controlling unit which generates the respective signals depending upon the transfer type and progress of that transfer. It detects the properties of a transfer on AXI side (Read/Write, Burst, Single, transfer type) and upon this information generate the AHB transfer signals appropriately. Below is the functional level flow diagram:1.

Figure 1: AXI2AHB Block Diagram

IP Specification

The working of an AXI2AHB bridge involves several key operations and considerations:

Bus Protocol Translation: One of the primary functions of the bridge is to convert AHB transactions into AXI transactions and vice versa. This translation ensures that data and control signals are correctly interpreted and processed by devices on both sides of the bridge.

Address Mapping: The AXI memory map and the AHB memory map are one single complete 32-bit (4 GB) memory space. The AXI to AHB Bridge core does not modify the address for AXI; hence, the address that is presented on the AXI is exactly as received on the AHB interface.

Control Signals: The bridge also translates control signals between the AXI and AHB buses. This includes signals like read and write enables, burst types, and response codes. Proper translation and synchronization of control signals are essential for maintaining the integrity of data transfers.

Error Handling: Robust error handling mechanisms are essential in any bridge design. The AXI2AHB bridge detects and manage errors, such as bus contention, address mapping faults, or data transfer errors, to ensure system reliability.

Configuration Options: Many AXI2AHB bridges offer configuration options to adapt to various system requirements. Designers can often configure parameters like address mapping, data width, and burst sizes to match the specific needs of their system.

The AXI2AHB module has two main parts: the write path and the read path. The write path takes data from the AXI4 slave interface and stores it in a write data FIFO. The read path takes data from a read data FIFO and provides it to the AXI4 master interface.

In addition, the AXI2AHB is equipped with the flexibility to support various burst types and parametrizable data and address interface widths. Moreover, it offers the option to delay the address channel until either the write data is entirely shifted into the FIFO or the read data FIFO can accommodate the entire burst.

Standards

The AXI4-Full interface is compliant with the AMBA® AXI Protocol Specification and AHB interface is compliant with AMBA® AHB5 Protocol Specification.

IP Support Details

The Table 1 gives the support details for AXI2AHB.

Compliance		IP Resources					Tool Flow	
Device	Interface	Source Files	Constraint File	Testbench	Simulation Model	Analyze and Elaboration	Simulation	Synthesis
GEMINI	AHB & AXI4 Full	System Verilog	SDC	Python	System Verilog	Raptor	Raptor Icarus	Raptor

Table 1: IP Details

Parameters

Table lists the parameters of the AXI2AHB.

Parameter	Values	Default Value	Description
DATA WIDTH	8,16,32,...,1024	32	Bridge Data width of data being transferred.
ADDR WIDTH	1-64	32	Bridge address width.
ID WIDTH	1-32	8	AXI ID width.

AXI2AHB Parameters

Port List

Table 2 lists the top interface ports of the AXI2AHB.

Signal Name	I/O	Description
AXI Clock and Reset		
clk	I	Bridge clock Clock
rst_l	I	Bridge clock RESET
Slave INTERFACE		
AXI WRITE ADDRESS CHANNEL		
axi_awvalid	I	AXI Write address valid
axi_awready	O	AXI Write address ready
axi_awaddr	I	AXI Write address
axi_awburst	I	AXI Burst
axi_awlen	I	AXI Burst length
axi_awsize	I	AXI Burst Size
axi_awprot	I	AXI Protection type
axi_awid	I	AXI write address ID
AXI WRITE DATA CHANNEL		
axi_wvalid	I	AXI Write valid
axi_wready	O	AXI Write ready.
axi_wdata	I	AXI Write data
axi_wstrb	I	AXI Write strobes
axi_wlast	O	AXI Burst last beat
AXI WRITE RESPONSE CHANNEL		
axi_bvalid	O	AXI Write response valid
axi_bready	I	AXI Response ready

axi_bresp	O	AXI Write response
axi_bid	O	AXI ID
AXI READ ADDRESS CHANNEL		
axi_arvalid	I	AXI Read address valid
axi_arready	O	AXI Read address ready
axi_araddr	I	AXI Read address
axi_arprot	I	AXI Protection type
axi_arburst	I	AXI Burst
axi_arlen	I	AXI Burst length
axi_arsize	I	AXI Burst Size
axi_arprot	I	AXI Protection type
axi_arid	I	AXI Read address ID
AXI READ DATA CHANNEL		
axi_rvalid	I	AXI Read valid
axi_rready	O	AXI Read ready
axi_rresp	I	AXI Read data
axi_rdata	O	AXI Read response
axi_rid	O	AXI Read channel ID
Master INTERFACE		
ahb_haddr	I	AHB Write address valid
ahb_hburst	I	AHB Write address ready
ahb_hmastlock	I	AHB Write address
ahb_hprot	I	AHB Protection type
ahb_hsize	I	AHB Write valid
ahb_htrans	I	AHB Write ready.
ahb_hwrite	I	AHB Write data
ahb_hwdata	I	AHB Write strobes
ahb_hsel	I	AHB Write response valid
ahb_hreadyin	I	AHB Response ready
ahb_hnonsec	I	AHB Write response
ahb_hrdata	O	AHB Read address valid
ahb_hreadyout	O	AHB Read address ready
ahb_hresp	O	AHB Read address

Table 2: AXI2AHB Interface

Resource Utilization

The parameters for computing the maximum and minimum resource utilization are given in Table 3, remaining parameters have been kept at their default values.

Tool	Raptor Design Suite			
FPGA Device	GEMINI			
Configuration			**Resource Utilization**	
Minimum Resource	**Options**	**Configuration**	**Resource**	**Utilized**
	DATA WIDTH	32	BRAMs	3
	ADDR WIDTH	16	LUTs	209
	WRITE FIFO DEPTH	32	Registers	293
Maximum Resource	**Options**	**Configuration**	**Resource**	**Utilized**
	DATA WIDTH	1024	BRAMs	62
	ADDR WIDTH	32	LUTs	255
	WRITE FIFO DEPTH	512	Registers	2426

Table 3: Resource Utilization

Design Flow

IP Customization and Generation

AXI2AHB IP core is a part of the Raptor Design Suite Software. A customized AXI2AHB can be generated from the Raptor's IP configurator window as shown in Figure 2.

Figure 2: IP list

Parameters Customization

From the IP configuration window, the parameters of the AXI2AHB can be configured and AXI2AHB features can be enabled for generating a customized AXI2AHB IP core that suits the user application requirement as shown in Figure 3. After IP Customization, all the source files are made available to the user with a top wrapper that instantiates a parameterized instance of the AXI2AHB.

Figure 3: IP Configuration

Synthesis and PR

Raptor Suite is armed with tools for Synthesis along with Place and Route capabilities and the generated post-synthesis and post-route and place net-lists can be viewed and analyzed from within the Raptor. The generated bit-stream can then be uploaded on an FPGA device to be utilized in hardware applications.

Test Bench

Currently testbench is not available.

Release

Release History

Date	Version	Revisions
August 21, 2024	1.0	Initial version AXI2AHB Bridge User Guide Document