Industrial Engineering Journal
ISSN: 0970-2555
Volume: 52, Issue 4, April: 2023

Design and Implementation of AMBA based AHB2APB Bridge using Verilog HDL

Kammari Akanksha, Madduri Jaceintha
D Ishwarya Assistant Professor
Department of Electronics and Communication Engineering, Bhoj Reddy Engineering College for Women, Vinaynagar, Santoshnagar.
kammariakanksha23@gmail.com, m.jaceintha2001@gmail.com

Abstract

The AMBA bus system is used in the development of high-performance SoC architectures. This accomplishes contact by connecting various functional components. It is feasible to create a multiprocessor device by tandardi numerous processors and peripherals. It allows for the reuse of IP from various AMBA buses, bridging the connection gap between high performance and medium speed buses. A system bus connects a CPU, DSP, and high-performance memory drivers in the AMBA Advanced High-performance Bus (AHB), whereas the AMBA Advanced Peripheral Bus (APB) connects Universal AsynchronousReceiver Transmitter. (UART). The AHB2APB Bridge is used to synthesise and model the composite annexation that links the advance high-performance bus and the advance peripheral bus.

This Bridge is a standard bus-to-bus interface that enables Ips connected to various buses to interact in a tandardized manner. Verilog HDL is used to create a synthesizable design of the AHB2APB bridge as well as a test bench for functional testing. The bridge is intended to handle transaction requests from the presently active AHB master. The primary goal of this effort is to convert AHB transactions into APB transactions with no data loss during transfer.

I Introduction

1.1 Introduction

The AMBA is an interconnection standard used in the electronics industry for system on chip (SOC) architecture. The AMBA interconnection standard supports high-speed, high-bandwidth, and pipelined data transmission by utilising a diverse collection of bus signals. Because of its successful application in Application Specification Integrated Circuit (ASIC) design, a number of ARM Partners and IP providers have embraced this specification. In some instances, if AHB side peripherals want to communicate with APB side peripherals, an in-between bridge is needed. A multiplexer interconnection method is used to execute the AHB bus interface.

The communication between master and slave in this method is as follows. The bus masters operate its address and control signals, which show whether the transmission is read or write. When the arbiter chooses the master with the control signal, the address is transmitted to the specific subordinate. The central decoder's primary function is to manage the data read and signal response multiplexer, which chooses the pertinent signals from the slaves that are needed for the transmission. The APB is linked to any peripherals that require limited bandwidth but do not require the high speed of a pipelined bus interface.

1.2 Objective

The AMBA specification was developed to meet four essential criteria. To make it easier to create embedded microprocessor devices with one or more CPUs or signal processors correctly the first time. To be technology-independent, and to guarantee that highly reusable periphery and system macrocells can be moved across a widevariety of IC processes, as well as to be suitable for full-custom, standard cell, and gate array technologies. Encourage modular system design to enhance processor freedom by offering a development roadmap for sophisticated cached CPU cores and peripheral library development. To reduce the silicon infrastructure needed to enable effective on-chip and off-chip communication for operation and production testing.

1.3 Literature Survey

The Advanced Microcontroller Bus Architecture (AMBA) was developed by ARM Ltd in 1996 and is extensively used as the on chip bus in system on chip (SoC) designs. AMBA is a protected brand of ARM Ltd. The first AMBA buses were Advanced System Bus (ASB) and Advanced Peripheral Bus (APB). (APB). In its second iteration, AMBA 2, ARM incorporated the AMBA High-performanceBus (AHB), which is a single clock-edge protocol.

Jaehoon Song et al. have published an effective testable design method for a SoC with an on/off-chip bus bridge for the on-chip advanced high-performance bus and off-chip peripheral-componentinterconnect bus.

The author, Krishna Sekar, developed and reported FLEXBUS, a novel architecture that can effectively adjust the logical connectivity of the communication architecture and the components connectedto it.

The bus bridge is used to connect Open Core Protocol (OCP) and AHB protocols, which are critical in SoC applications to prevent application failure in the event of a problem. G. Geetha Reddy et al. initially model these fundamental protocols independently in VHDL and simulate them. Their simulation findings demonstrate that contact via the bridge between AXI 4.0 and APB4.0 is suitable.

The bridge effectively transferred all instructions and data from AXI 4.0 to APB4.0 protocol, with minimal loss of data or control information discovered by Kalluri Usha and T. Ashok Kumar. The Advanced Microcontroller Bus design (AMBA) is an on-chip bus design that is extensively used to improve the reusability of IP cores and is a widely used interconnection standard for system on chip. (SOC).

1.4 A Typical AMBA based Microcontroller

An AMBA-based microcontroller usually consists of a high-performance system backbone bus (AMBA AHB or AMBA ASB) capable of supporting external memory capacity, which houses the CPU, on-chip memory, and other Direct Memory Access (DMA) devices.

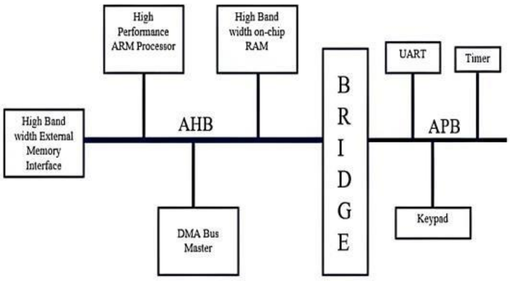

High Performance ARM Processor
High Band width on-chip RAM
UART
Timer
B R I D G E
High Band width External Memory Interface
AHB
APB
DMA Bus Master
Keypad

Figure 1 A Typical AMBA based Microcontroller

This bus connects the components engaged in the bulk of transactions by providing a high-bandwidth link. A connection to the lesser bandwidth APB, where the majority of the system's peripheral devices are situated, is also placed on the high-performance network. (see Figure 1).

AMBA AHB is a bus interface designed for high-performance synthesizabledesigns. It specifies the link betweencomponents such as controllers, interconnects, and slaves. AMBA AHB includes the characteristics needed for high-performance, high-clock-frequencysystems, such as:

- Burst transfers.

- Single clock-edge operation;

non-tristate design;

and wide data bus combinations of 64, 128, 256, 512, and 1024 bits.

Internal memory devices, external memory connections, and high-bandwidth accessories are the most commonly used AHB slaves. Although low-bandwidth peripherals can be included as AHB slaves, for system efficiency reasons, they are usually located on the AMBA Advanced

Peripheral Bus. (APB). An AHB slave, also known as an APB bridge, is used to connect the better performance AHB and APB.

II AMBA AHB and APB

2.1 Introduction

The Advanced Microcontroller Bus Architecture (AMBA) specifications defines an on-chip communication standard for designing high-performance embedded microcontroller.

- Advanced High-Performance Bus (AHB).
- Advanced System Bus (ASB).
- Advanced Peripheral Bus (APB).

2.2 AHB (AMBA High-performance Bus).

AMBA AHB is a bus interface suitable for high-performance synthesizable designs. It defines the interface between components, such as masters, interconnects, and slaves. AMBA AHB implements the features required for high-performance, high clock frequency systems including:

- Burst transfers.
- Single clock-edge operation.
- Non-tristate implementation.
- Wide data bus configurations, 64, 128, 256, 512, and 1024 bits.

The most common AHB slaves are internal memory devices, external memory interfaces, and high-bandwidth peripherals. Although low-bandwidth peripherals can be included as AHB slaves, for system performance reasons, they typically reside on the AMBA Advanced Peripheral Bus (APB). Bridging between the higher performance AHB and APB is done using an AHB slave, known as an APB bridge.

A standard AMBA AHB system architecture includes the following components:

AHB master: A bus master can receive and write data by giving a location and control information. Only one bus master may operate the vehicle at any given moment.

AHB slave: A bus slave is a device that reacts to read or write actions within a specific address-space range. The bus slave reports the success, failure, or writing of the data transmission to the current master.

2.3 APB (Advanced Peripheral Bus)

The APB is a bus in the AMBA system that is optimised for low power usage and low interface complexity. The AMBA APB shows as a local auxiliary bus that is encapsulated as a single AHB or ASB slave device. APB is a low-power extension to the system network thatimmediately builds on AHB or ASB signals. The APB bridge shows as a subordinate module that manages the bus handshake and control signal retiming on behalf of the local peripheral bus. The advantages of system diagnostics and test methods can be leveraged by defining the APB interface from the system bus's beginning point. The AMBA APB should be used to connect to any low bandwidth

peripherals that do not require the high speed of a pipelined bus interface.

An AMBA APB application usually includes a single APB bridge, which is needed to convert AHB or ASB transfers into a format appropriate for the slave devices on the APB. The bridge latches all address, data, and control signals and provides a second level of decoding to produce slave select signals for the APB devices.

All other components on the APB are APB slaves. The APB slaves have the following protocol specifications:

- Address and authority are good for the duration of the entry. (unpipelined).
- Zero-power link when not using an auxiliary bus (peripheral bus is staticwhen not in use).
- Decode with flash timing can provide timing. (unclocked interface).
- Write data that is good for the duration of the access (allowing glitch-free transparent latch implementations).

III AHB to APB Bridge

3.1 Introduction

The AHB to APB bridge interface is an AHB slave. When accessed (during regular operation or a system test), it starts an access to the APB. The length of APB usage varies. (three HCLK cycles in the EASY for a read, and two cycles for a write). They also have a set width of one word, which implies that only an 8-bit portion of a 32-bit APB register can be written. APB devices do not require a PCLK input because APB access is scheduled with an enable signal produced by the AHB to APB bridge link. Because APB peripherals are only strobed when requested, they have a minimal power usage.

3.2 AHB to APB Bridge Block Diagram

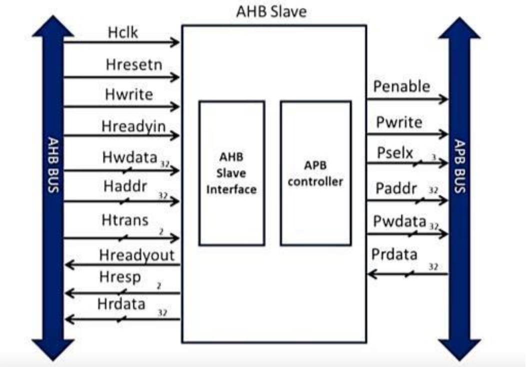


Figure 2 AHB to APB bridge blockdiagram

The AHB to APB bridge is an AHB slave, providing an interface between the high speed AHB and the low-power APB. Read and write transfers on the AHB are converted into equivalent transfers on the APB. As the APB is not pipelined, then wait states are added during transfers to and from the APB when the AHB is required to wait for the APB.

It is required to bridge the communication gap between low bandwidth peripherals on APB with the high bandwidth ARM Processors and/or other high-speed devices on AHB. This ensures that there is no data loss between AHB to APB or APB to AHB data transfers. AHB2APB interfaces AHB and APB. It buffers address, controls and data from the AHB, drives the APB peripherals and return data along with response signal to the AHB.

The AHB bus protocol is implemented with a multiplexer interconnection technique. In this technique the communication of master and arbiter is as follows, The bus masters drives its address and control signal which indicates the transfer type i.e. whether it is a read or write transfer. Address routed to the particular slave when the arbiter selects the master which having its control signal. Main role of the central decoder is to control the data read and signal response multiplexer, which selects the relevant signals from the slaves that is required for the transfer. The APB is connected to any peripherals which are low bandwidth and do not need high performance of a pipelined bus interface.

The AHB2APB interface is designed to operate when AHB and APB clocks have the any combination of frequency and phase. TheAHB2APB performs transfer of data from AHB to APB for write cycle andAPB to AHB for Read cycle. Interface between AMBA high performance bus (AHB) and AMBA peripheral bus (APB). It provides latching of address, controls and data signals for APB peripherals.

3.3 Architecture

The architecture of the AHB2APB bridge shown in Figure 3 contains three modules: bridge top, AHB slave interface, and APB controller, which are further described.

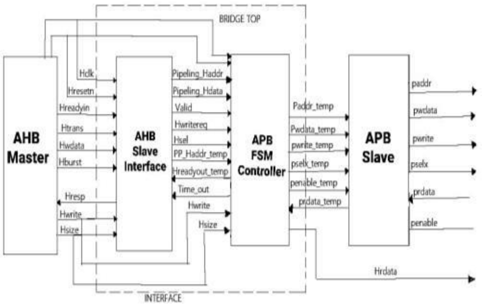
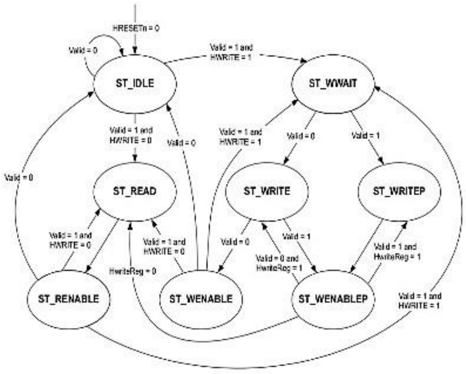


Figure 3 Architecture of AHB to APB bridge

Bridge Top: It consists of twofundamental modules: the AHB slave interface and the APB FSM driver. These modules are instantiated into the primary bridge top and connected via signal flow, which is specified in the port list of both the AHB interface and the APB controller's submodules.



AHB Slave Interface: This module contains the logic required to implement TSELx, which is used to select the peripheral memory map, valid and pipelined address, data and control channel, as well as all the input and output signals that define the module for further instantiation and connection.

APB Controller: It is the heart of the design. This module is helpful for making decisions that are needed for the correct operation of the design based on the current state, next state logic, and output logic.

IV Bridge Design

4.1 Introduction

The AHB to APB bridge is made up of a state machine that controls the production of the APB and AHB output signals, as well as address decoding circuitry that generates the APB peripheral select lines. All registers in the system are timed from the rising edge of the system clock HCLK and use the asynchronous reset HRESETn.This module employs the standard AHB slave bus interface, which includes valid transfer detection logic that determines when a legitimate transfer is reaching the slave. The address and control files are used to save information from thetransfer's address phase for use in the data phase. Because read and write transfers have distinct AHB to APB timing, either the current or prior address input value is required to generate the APB transfer properly. A multiplexor is thus used to pick between the current address input and the recorded address for read and write transfers, respectively.

4.2 APB Transfer State Machine

Figure 4 Finite State Machine

The transfer state machine is used to manage the application of APB transfers depending on the AHB inputs. The state diagram in Figure 4 depicts the function of the state machine, which is governed by its present state and the AHB slave interface.

1. AHB transaction with HREADYout indication under state machine management
2. Production of APB product indications.

According to Haddr, a state machine is used to monitor distinct Pslex signals. Then APB begins to progress by authorised production. In the event of an unknown location, no specific peripheral get is favoured.

V Results

5.1 Introduction

In this section we will observe the simulated results for single read transfer and single write transfer of AHB2APB bridge and synthesis of bridge which includes AHB slave and finite state machine.

5.2 Simulation of Single Write

A single write transmission to the APB canoccur with zero wait conditions. The bridge is in charge of sampling the location and data of the transfer and storing these values on the APB for the length of the write transfer. While the first transfer can be completed with zero wait states, future transfers to the periphery bus will require a single wait state for eachtransfer to complete.

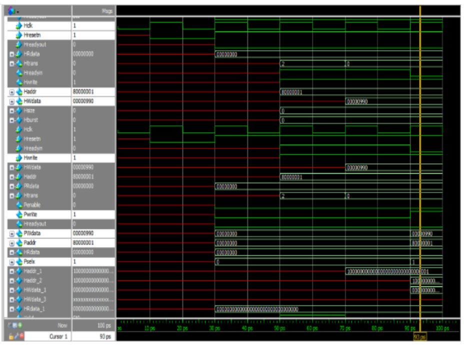


Figure 5 Simulation of Single Write

5.3 Simulation of Single Read

The transmission begins on the AHB at time T1, and the address is sampled by the APB bridge in the following cycle. If the transfer is for the peripheral bus, this location is transmitted and the proper peripheral select signal is produced. The first cycle on the peripheral bus is known as the SETUP cycle, and it is followed by the ENABLE cycle, which occurs when the PENABLE signal is affirmed. During the ENABLE stage, the peripheral mustprovide the read data. Normally, this read data can be routed straight back to the AHB, where the bus master can sample it on the rising edge of the clock at the conclusion of the ENABLE cycle, as shown in figure 6

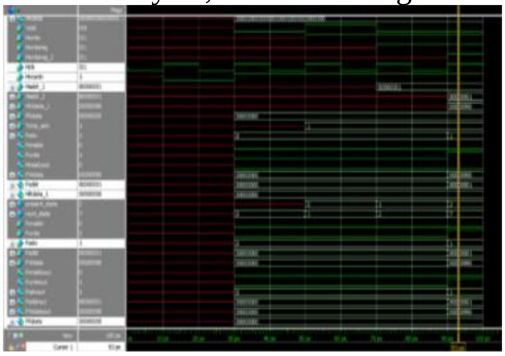


Figure 6 Simulation of Single Read

5.4 AHB to APB Bridge Synthesis

Figure 7 depicts the composition of an AHB2APB bridge. It includes two modules: AHB Slave and APB Controller. The primary purpose of AHB Slave is to pipeline the input signals transmitted from AHB Master. The APB Controller is a Finite State Machine that generates output impulses. The AHB to APB bridge connects the high-speed AHB and the low-power APB.

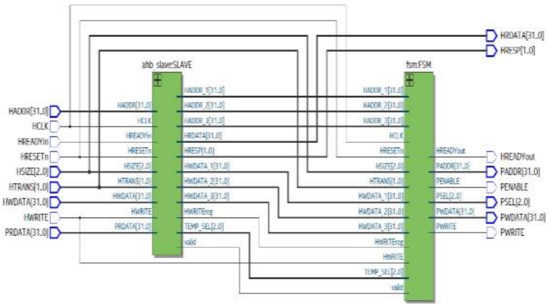


Figure 7 Synthesis of AHB to APB bridge

5.5 AHB Slave Synthesis

The figure 8 shows the internal design of AHB Slave.

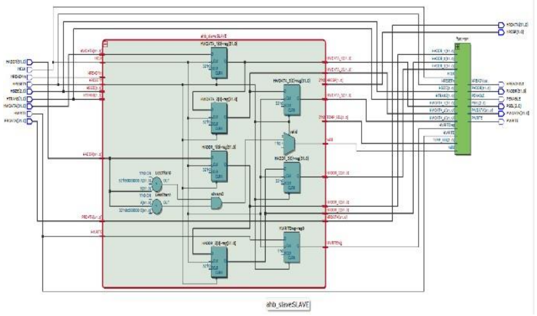


Figure 8 Enhanced Synthesis of AHB Slave

5.6 Finite State Machine Synthesis

The figure 9 shows the internal design of Finite State Machine.

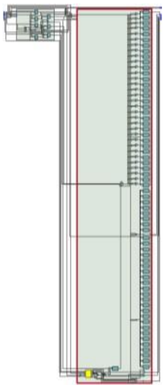


Figure 9 Synthesis of Finite State Machine

VI Conclusion

In this paper, a synthesizable design of an AHB2APB bridge and a testbench for functional verification are created using verilog HDL. The RTL Simulation of the AHB2APB Bridge was validated and proven using suitable test platforms such as the AHB Module and the APB Module. The AHB2APB bridge design is developed in Verilog HDL for single read and single write transfers, and all of these designs are simulated using the ModelSim programme. The aim of protocol AHB2APB is to achieve maximum code and functional coverage in order to increase design efficiency. As a consequence, the AHB2APB Bridge is a stand-alone solution that bridges the dividebetween the recently developed ARM- based AMBA AHB bus and the current APB bus. It has the following benefits: high speed, pipelined operation, and high broadband operation. Many schedulingissues are resolved by bus adjudication. It's simple to introduce new components. It has a minimal battery consumption and great latency. It provides less expensive boards. It is used in the creation of high- performance SoC systems. (system onchip).

References

[1] Prof. Shiva Kumar P R "AHB2APB Bridge Design", ARM limited 2020.

[2] M Kiran Kumar, Amrita Sajja, Dr, Fazal Noorbasha, "Design and FPGA Implementation of AMBA APB bridge with clock skew minimization Technique" (IOSRJVSP), Vol.7, Issue.3, June 2017.

[3] N.G.N. Prasad, "Development and Verification of AHB2APB Bridge Protocolusing UVM Technique". IJSRR, Vol.6, Issue.12, 2017

[4] Prof. Ravi Mohan Sriram and Prof. Sumith Sharma, "FSM and Handshaking Based AHB to APB Bridge for highspeed systems", International journal of Engineering Research and Technology (IJERT). Vol.2 issue 11, November 2019.

[5] Aparna Kharade and V. Jayashree, "VLSI Design of AMBA based AHB2APB Bridge", International Journal of VLSI design & Communication Systems (VLSICS) Vol.9, No.3, June 2018.

[6] Vani R M and N Roopa, "Design of AMBA based AHB2APB Bridge", IJSCNS International Journal of Computer Science And Network Security, Vol.10 no.11 November 2019.

[7] Prof. Shiva Kumar P R "AHB2APB Bridge Design", ARM limited 2020.

[8] M Kiran Kumar, Amrita Sajja, Dr, Fazal Noorbasha, "Design and FPGA Implementation of AMBA APB bridge with clock skew minimization Technique" (IOSRJVSP), Vol.7, Issue.3, June 2017.

[9] N.G.N. Prasad, "Development and Verification of AHB2APB Bridge Protocolusing UVM Technique". IJSRR, Vol.6, Issue.12, 2017

[10]Prof. Ravi Mohan Sriram and Prof. Sumith Sharma, "FSM and Handshaking Based AHB to APB Bridge for highspeed systems", International journal of Engineering Research and Technology (IJERT). Vol.2 issue 11, November 2019.

[11] Aparna Kharade and V. Jayashree, "VLSI Design of AMBA based AHB2APB Bridge", International Journal of VLSI design & Communication Systems (VLSICS) Vol.9, No.3, June 2018.

[12] Vani R M and N Roopa, "Design of AMBA based AHB2APB Bridge", IJSCNS International Journal of Computer Science And Network Security, Vol.10 no.11 November 2019.