

Design and Implementation of AMBA based AHB2APB Bridge

T. Koundinya

Electronics and Communications Department,
Sri Chandrasekarendra Saraswathi Viswa Maha Vidyalaya [SCSVMV],
Kanchipuram, Tamil Nadu.

Abstract: The Advanced Microcontroller Bus Architecture (AMBA) is an on-chip bus architecture that is designed to improve IP core reusability and is a commonly used interconnection standard for system on chip (SOC). The examination of AMBA-based embedded systems is a difficult task. The goal of this study is to synthesise and simulate the AHB2APB Bridge, a complex interface bridge between the Advanced High-Performance Bus (AHB) and the Advanced Peripheral Bus (APB). The Synthesized Net List of Bridge Module is generated in this paper. Using Xilinx and Modelsim softwares and Verilog HDL, to run functional and timing simulations.

Keywords: Bus Architecture, System on Chip, Pipelined Data, SOC, Simulation, Verilog HDL, Handshaking Signal, AMBA, APB, AHB.

1. INTRODUCTION:

The mission of AMBA is to assist embedded system designers in overcoming issues such as low power design, facilitating right-first-time development of Embedded Microcontroller Products with one or more CPUs or signal processors, being technology-independent, and encouraging modular systems. To reduce the amount of silicon infrastructure needed to support efficient on-chip and off-chip connectivity for both operation and manufacturing test.

The focus of this research paper is on the synthesis and simulation of an AMBA-based AHB2APB Bridge. AHB2APB Bridge is a bridge that connects the AHB and APB buses. The communication gap between low bandwidth peripherals on APB and high bandwidth ARM Processors and/or other high-speed devices on AHB must be bridged. This is to ensure that no data is lost during data transfers from AHB to APB or APB to AHB.

The RTL (Register Transfer Level) code was designed using Verilog HDL (Hardware Description Language). Xilinx and Modelsim are used for synthesis and simulation.

2. TYPICAL AMBA BASED MICROCONTROLLER:

A typical AMBA-based microcontroller comprises of a high-performance system backbone bus (AMBA AHB or AMBA ASB) capable of supporting external memory bandwidth, on which the CPU, on-chip memory, and other Direct Memory Access (DMA) devices are located. This bus connects the elements that are involved in the majority of transfers and provides a high-bandwidth interface. A bridge to the lower bandwidth APB, which houses the majority of the system's peripheral devices, is also located on the high-performance bus. As a supplemental bus to the higher bandwidth pipelined main system bus, AMBA APB offers the basic peripheral macro cell communications architecture.

These peripherals generally:

- (i) have memory-mapped register interfaces.
- (ii) do not have high-bandwidth interfaces.
- (iii) are accessed under programmed control.

II. DESIGN OF AMBA BASED AHB2APB BRIDGE

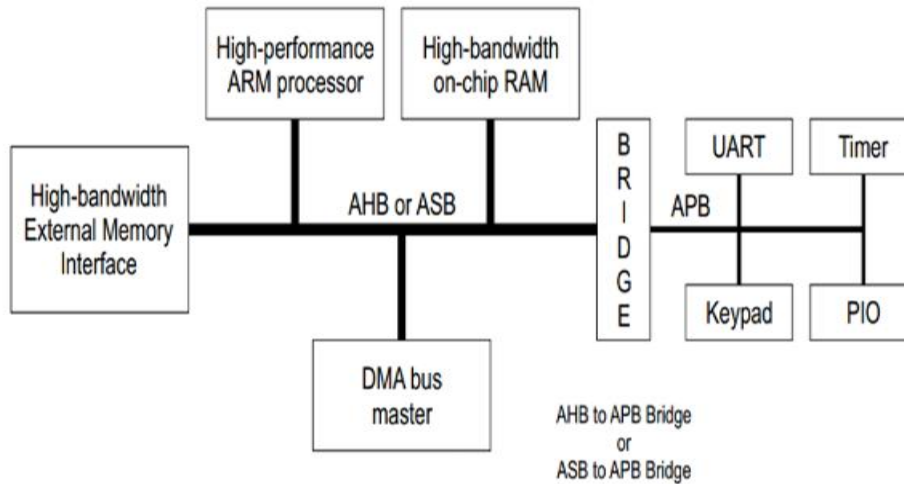


Fig.1. AMBA based AHB2APB Bridge

3. AMBA BUSES OVERVIEW:

The Advanced Microcontroller Bus Architecture (AMBA) is an open, no-cost specification developed by ARM that defines an on-chip communications standard for high-performance Embedded Microcontrollers. Within the AMBA specification, three separate buses are defined: The Advanced High-Performance Bus (AHB), the Advanced System Bus (ASB), and the Advanced Peripheral Bus (APB) are the three types of advanced buses (APB)

Advanced high-performance bus (AHB): AHB is a new AMBA bus generation designed to meet the needs of high-performance synthesizable designs. It's a high-performance system bus with numerous bus masters and high-bandwidth capabilities. AMBA AHB includes the following capabilities for high-speed, high-clock-frequency systems:

- High Performance
- Pipelined operation
- Multiple bus masters
- Burst transfers
- Single-cycle bus master handover
- Non-tri state implementation
- Configurations with a larger data bus (64/128 bits). Bridging between this higher level of bus and the current ASB/APB may be accomplished quickly, allowing any existing designs to be simply integrated. One or more bus masters may be included in an AMBA AHB design. In most cases, a system will include at least the CPU and the test interface. Bus masters can also be Direct Memory Access (DMA) or Digital Signal Processor (DSP) devices. The most frequent AHB slaves are the external memory interface, APB Bridge, and any internal memory. As an AHB slave, any other peripheral in the system could be used. Low, on the other hand bandwidth peripherals typically reside on the APB.

The AMBA Advanced System Bus (ASB) is designed for high-performance modules. It's a different system bus that can be used when the AHB's high-performance characteristics aren't needed. ASB also enables low-power peripheral macro cell functionalities to connect CPUs, on-chip memories, and off-chip external memory interfaces. ASB's characteristics include:

- Burst transfers
- Pipelined transfer operation
- Multiple bus masters.

The Advanced Peripheral Bus (APB) is part of the Advanced Microcontroller Bus Architecture (AMBA) hierarchy of buses and is geared for low power consumption and decreased interface complexity. The AMBA APB is intended for low-bandwidth peripherals that do not require the high performance of a pipelined bus interface. The APB's most recent modification assures that all signal transitions are tied to the clock's rising edge.

As a result of this enhancement, the APB peripherals may now be readily included into any design flow.

APB's features include:

- Low power

- Latched address and control
- Simple interface
- Suitable for a wide range of peripherals The APB has been updated to make it easier to connect it to the new Advanced High-performance Bus (AHB). [3]

4. AHB2APB BRIDGE SPECIFICATIONS:

The AHB2APB bridges the gap between AHB and APB. It buffers the AHB's address, controls, and data, drives the APB peripherals, and sends data and a response signal back to the AHB [4]. The AHB2APB interface is intended to work when the AHB and APB clocks have any frequency and phase combination. The AHB2APB transfers data from the AHB to the APB for the write cycle and from the APB to the AHB for the read cycle.

AMBA high-performance bus (AHB) and AMBA peripheral bus (APB) interface [2]. Address, control, and data signals are latched for APB peripherals. Supports the following initiatives:

- Slaves and peripherals that are APB compatible.
- Peripherals which require additional wait states.

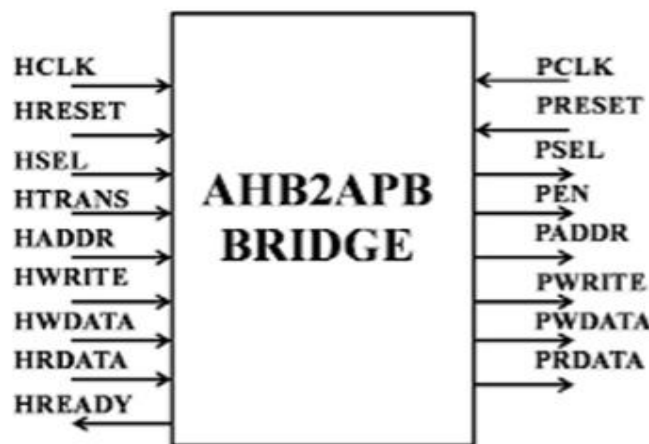


Figure 2. Pin details of AHB2APB Bridge

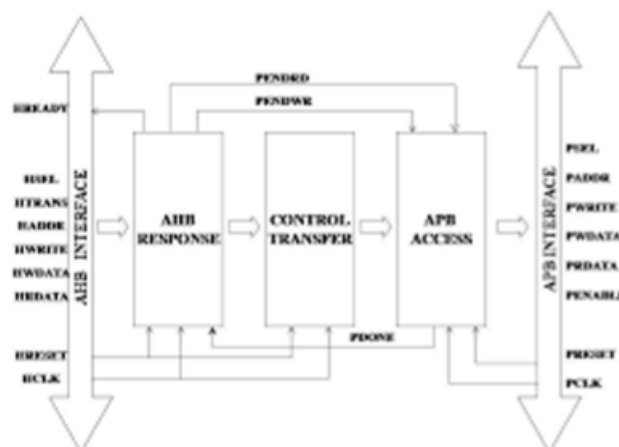


Figure 3. Internal architecture of the bridge

5. IMPLEMENTATION

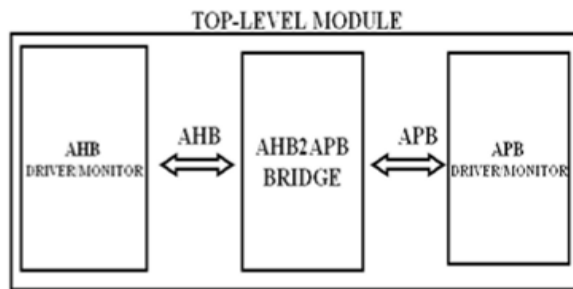


Figure 4. Design of AHB2APB Bridge

5.1 Design of AHB2APB Bridge

The AHB2APB Bridge uses HCLK, while the APB access sub module uses PCLK. The AHB response and control transfer are referred to as the AHB interface, whereas the APB access is referred to as the APB interface. We use three internal signals in the bridge module to assure the accurate creation of appropriate control signals and addresses: PENDWR (Pending Write), PENDRD (Pending Read), and PDONE (Peripheral operation done). When HREADY, HTRANS, and HSEL are valid, the address and control are captured for Write or Read operations. There is a necessity for linking these clock domains because the sub modules run on distinct clock domains, namely HCLK and PCLK. At the interface's boundary, the design becomes asynchronous, resulting in setup and hold time violations, metastability, and faulty data transfers. As a result, we'll need to seek for design and interface strategies. If we need to transfer data in this situation, there are just a few options, including

- Handshake signalling method
- Asynchronous FIFO

Both have their own set of benefits and drawbacks. We employed the Handshake Signalling Method in our study. Based on the handshake signals PENDWR (or PENDRD) and PDONE, the AHB interface delivers data to the APB interface using the Handshake signalling mechanism. The protocol for this is based on the same handshake signals Request and Acknowledge as the 8155-chip used with the 8085.

The PENDWR (or PENDRD) signal is asserted by the AHB interface, instructing the APB interface to receive or send data on the data bus. The PDONE signal is asserted by the APB interface, indicating that it has accepted or transferred the data. This method is simple, but it has flaws: when the APB interface samples the AHB interface's PENDWR (or PENDRD) line and the AHB interface samples the APB interface's PDONE line, they do so use their internal clocks, causing setup and hold time violations. To avoid this, we use two stage synchronizers, which are rather resistant to metastability.

The below figure shows the process

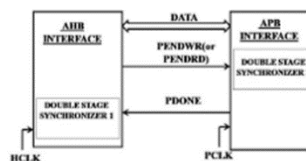


Figure 5. Handshake signalling method

The internal blocks of a double stage synchronizer for PENDWR are shown in the diagram below.

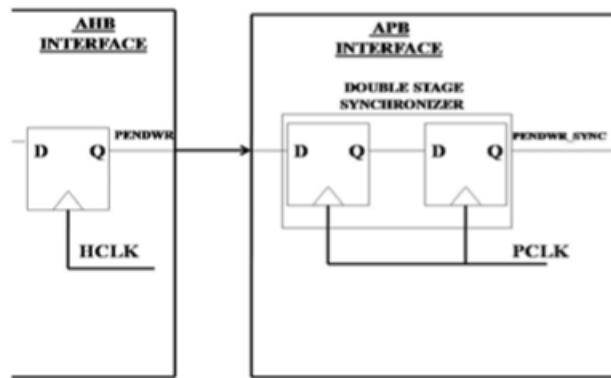


Figure 6. Double Stage Synchronizer

The double synchronizers for PENDRD and PDONE will be the same as the double synchronizers for PENDWR, with the exception that the PDONE synchronizer will work on HCLK rather than PCLK like PENDWR and PENRD. Because many clock cycles are wasted just handshaking when we do the double synchronisation, the transfer rate decreases.

5.2 STATE MACHINE CAUSE AHB TO APB INTERFACE

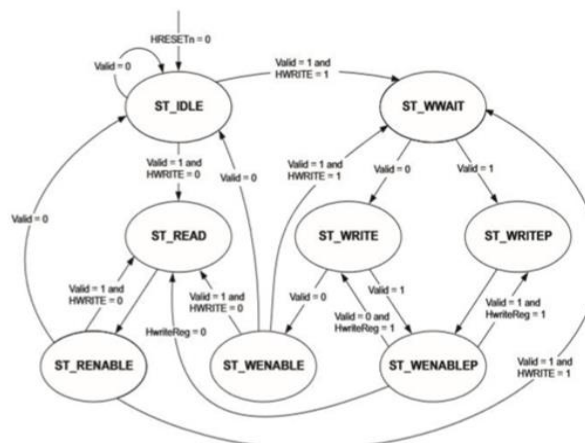


Fig. 3. FSM of AHB2APB

State machine control:

- AHB transaction with HREADY out signal.
- Generation of each product signals of APB.

According to Haddr, a state machine is utilized to track different Pselx signals. After that, APB begins to proceed with permitted output. [3] There is no single peripheral get selected in the situation of an unknown location.

5.2.A. Execution Steps of State Machine

Present State	Input	Next State	Output
ST_IDLE	Valid = 1 Hwrite = 0	ST_READ	AHB have valid APB Read transfer
	Valid = 1 Hwrite = 1	ST_WAIT	AHB have valid APB Write transfer
ST_READ		ST_REENABLE	Enable Current APB transfer
ST_REENABLE	Valid = 1 Hwrite = 1	ST_WAIT	
	Valid = 1 Hwrite = 0	ST_READ	AHB have valid APB Read transfer
ST_WAIT	Valid = 1	ST_WRITEP	Address decoded with presence of wait state to perform pending transfer.

	Valid = 0	ST_WRITE	Address decoded with no wait State
ST_WRITE	Valid = 0	ST_WENABLE	Enable current APB transfer.
	Valid = 1	ST_WENABLEP	Enable wait state for pending transfer
ST_WENABLE	Valid = 1 Hwrite = 0	ST_READ	AHB have valid APB Read transfer.
	Valid = 1 Hwrite = 1	ST_WAIT	AHB have valid APB Write trans
	Valid = 0	ST_IDLE	No transfer is to perform
ST_WRITEP		ST_WENABLEP	Enable wait state
ST_WENABLEP	Valid = 0 Hwritereg = 1	ST_WRITE	Address decoded with no wait State
	Valid = 1 Hwritereg = 1	ST_WRITEP	Address decoded with presence of wait state to perform pending transfer
	Hwritereg = 0	ST_READ	AHB have valid APB Read transfer

6. APPLICATION OF BRIDGE:

- This IP is used to create affiliate and consistent systems on chips.
- Allow for a swappable Soc unit.
- For rephrasing various IP.
- Enabling the design of SoCs with multiple CPUs.
- Enable high-speed, high-performance transmission with low-power consumption.

7. QUALITIES COVERED BY AHB2APB BRIDGE:

- Connection establishment between the Advance high-performance bus and the Advance peripheral bus hooks low-power peripheral information signals (Address, data, and control)
- For best performance, synchronisation between the user (client) and the attendant (AHB and APB) is also required.

5.2 Design of AHB module

The AHB module provides appropriate control signals, address, and data to the AHB2APB Bridge [5]. Monitors the input data from the bridge so that an AHB master environment can be created.

This module also contains the HCLK generating block, which is distributed to module AHB2APB Bridge, and the RESET generation block, which is distributed to two other modules, AHB2APB Bridge and APB module, and these two signals are evidently used in this module as well.

5.3 Design of APB module

The APB module is the module that provides appropriate data to the AHB2APB Bridge and also monitors the control signals, address, and data received from the bridge in order to build the APB Slave environment [6]. Depending on whether it is a write or a read operation, the control signals, address, and data received from the bridge are appropriately employed for the data transaction from bridge to this module or vice versa. This module has a PCLK generation block that is distributed to the AHB2APB bridge module, and the PCLK generated is clearly utilised in this module as well.

5.4 Design of top module

This module is the quickest and most prominent of all the ones available. All signals are used as wires to connect many modules that make up this top module. All three modules are covered in this module:

- AHB Module
- AHB2APB Bridge
- APB Module.

Positional assignments, which are more straightforward than naming assignments, are used to instantiate all of these modules..

RTL SCHEMATIC OF TOP MODULE

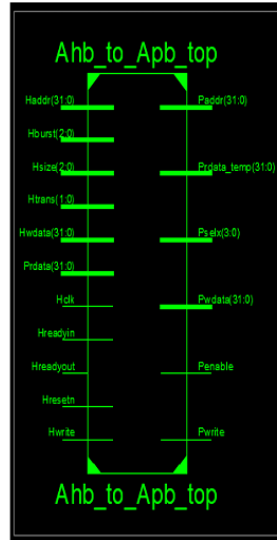


Fig. 5. RTL Schematic of TOP Module

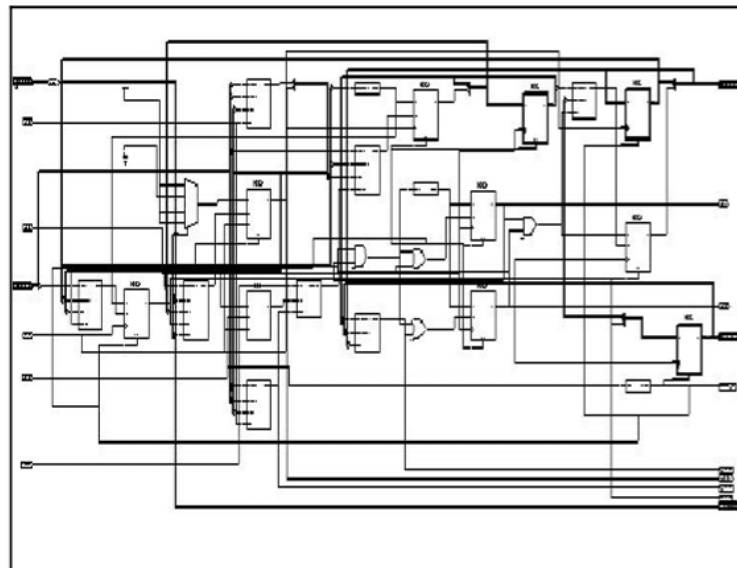


Figure 7. Synthesized Netlist of Bridge Module

6. SIMULATION RESULTS:

Back annotation is the process of converting a routed or fitted design into a timing simulation Netlist. On the AHB2APB Bridge module synthesis file provided by Xilinx, back annotation was performed. In our paper, only the Bridge module was synthesised; the AHB driver and APB monitor were test bench modules.

APB monitor were test bench modules.

- A. With HCLK and PCLK having a ratio of 1:2

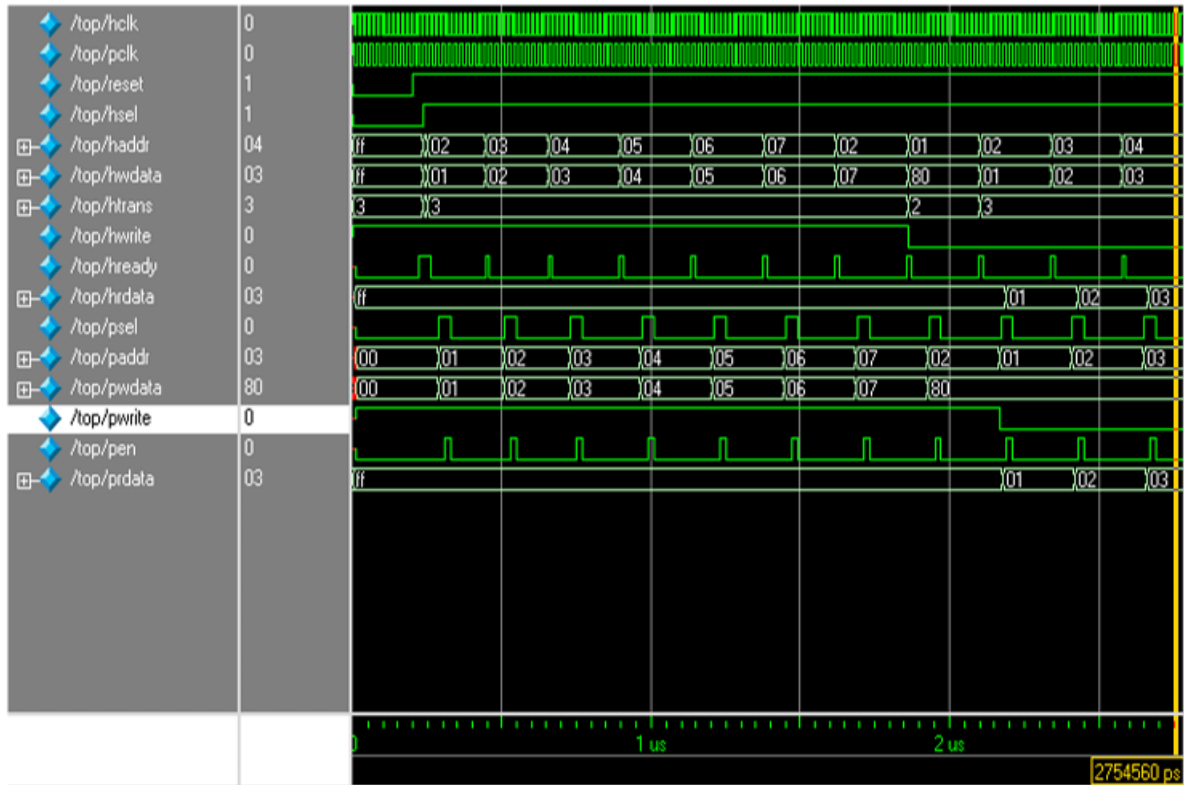


Figure 8. Burst of Write & Read Transfers

B. With a phase difference of 900 and the same frequency, HCLK and PCLK are a good match.

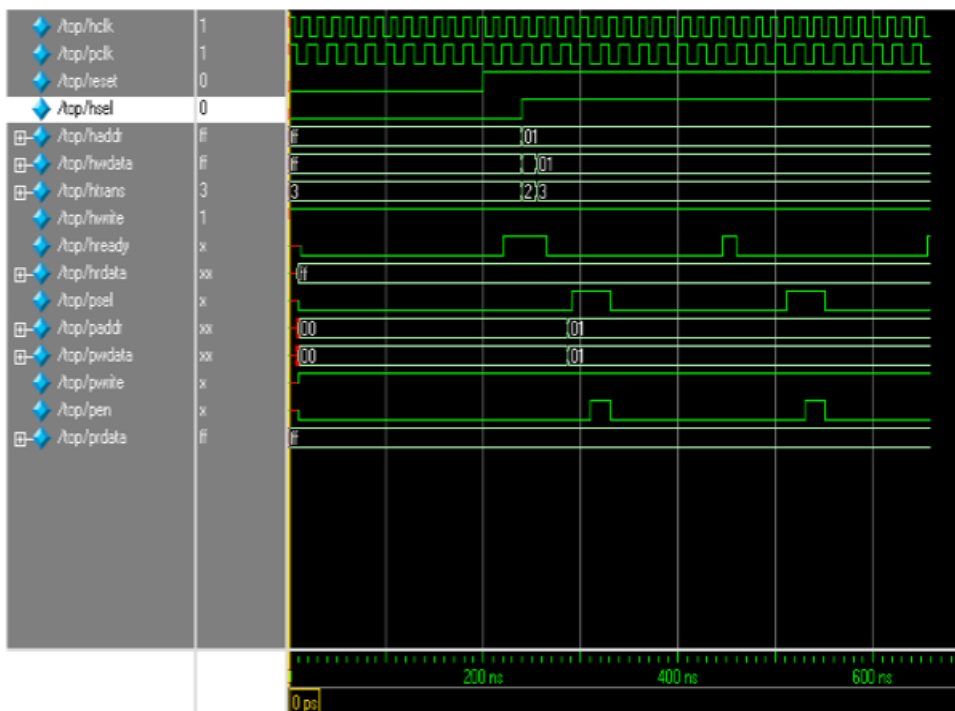


Figure 9. Single Write Transfer

8. CONCLUSION:

Using appropriate test benches, such as AHB Module and APB Module, the RTL Simulation of the AHB2APB Bridge was verified and confirmed.

The AHB2APB bridge design is implemented in Verilog HDL for read, write, read burst, write burst and write transfers, and all these designs are simulated using the Xilinx ISE software. By implementing the timeout concept, data loss can be minimized, and the design can become more extensible. The goal of protocol AHB2APB is to obtain maximal code and functional coverage in order to improve design efficiency.

As a result, the AHB2APB Bridge is a stand-alone solution that bridges the gap between the newly created ARM-based AMBA AHB bus and the current APB bus.

9. REFERENCES:

- [1] A. Suresh and S. Nandi, "Physical Bridge Design for High Performance to Peripheral Bus: An Open Source Approach," *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAIECC)*, 2020, pp. 1-5, doi: 10.1109/ICAIECC50550.2020.9339522.
- [2] Prakash, Peram Bhanu, Panta Nishith Reddy, Maddireddy Sathish Reddy, Rachapalyam Vignesh Kumar, and G. Bharatha Sreeja. *Design of AMBA Based AHB2APB Bridge Protocol*. No. 7890. EasyChair, 2022.
- [3] Roopa, M. "Design of AMBA based AHB2APB bridge." *IJCSNS* 10, no. 11 (2010): 14.
- [4] MITTAL, PULKIT. "BUS CONTROLLED AMBA 2.0 AHB2APB BRIDGE FOR SOC APPLICATION." PhD diss., 2022.
- [5] Pal, Geeta, and Ravi Mohan. "Design of ahb2apb ridge for efficient utilization of ahb and apb." *ISSN: 2320-9984 International Journal of Modern Engineering & Management Research* 1, no. 4 (2012).
- [6] Verma, A. and Bhatt, H., AMBA Based RTL Design of AHB2APB Bridge. __, p.64.
- [7] M. kiran Kumar, Amrita Sajja, Dr, Fazal Noorbasha, "Design and FPGA Implementation of AMBA APB bridge with clock skew minimization Technique" (IOSR-JVSP),Vol.7,Issue.3,June 2017