DESIGN AND IMPLEMENTATION OF THE AHB TO APB BRIDGE

Project Reference No.: *49S_BE_5363*

College : VSM's Somashekhar R Kothiwale Institute of Technology, Nipani, Belagavi
Branch : Department of Electronics and Communication Engineering.
Guide : Prof. Madhavi J.A
Student(s) : Mr. Shankar M Indi
Mr. Pranit Kore
Mr. Shivaraj Ambare
Mr. Vishal Patil

Keywords:
AHB (Advanced High-performance Bus), APB (Advanced Peripheral Bus), AMBA Protocol, Bridge Design, VLSI / RTL Design, Verilog HDL, System-on-Chip (SoC).

Introduction:

1. In modern System-on-Chip (SoC) architectures, efficient communication between various IP cores and peripherals is critical to overall system performance. ARM's Advanced Microcontroller Bus Architecture (AMBA) is a widely adopted on-chip bus standard that defines the communication protocols between components within an SoC. Among the many protocols defined under AMBA, the Advanced High-performance Bus (AHB) and the Advanced Peripheral Bus (APB) are two of the most commonly used.
2. The AHB is a high-bandwidth, pipelined bus primarily used for connecting high-performance components such as processors, DMA controllers, and on-chip memories. It supports burst transfers, split transactions, and multiple bus masters, making it suitable for high-speed data transfer. In contrast, the APB is a simpler, low-power, non-pipelined bus used for connecting slow, low-bandwidth peripheral devices such as UARTs, timers, GPIO controllers, and similar components.
3. Since AHB and APB operate differently — in terms of timing, signal protocols, and data transfer mechanisms — a direct connection between them is not possible. A bridge circuit is therefore required to serve as an interface between the two buses. The AHB-to-APB bridge translates transactions from the AHB side into APB-compatible transactions, ensuring seamless data transfer between high-speed and low-speed components within the SoC.
4. This project focuses on the design and implementation of a functional AHB-to-APB bridge using hardware description language (Verilog HDL). The design is simulated, verified functionally, and synthesized to validate its correctness and performance within a typical SoC environment.

Objectives:

1. **Design and Develop the AHB–APB Bridge.**
   To design and implement a bridge that connects the high-speed AHB bus with the low-speed APB bus for efficient system integration.

2. **Implement Protocol Conversion Mechanism.**
   To enable accurate translation of AHB transactions into APB-compatible signals, ensuring smooth communication between different bus protocols.
3. **Ensure Efficient and Reliable Data Transfer.**
   To manage timing, control signals, and address decoding for stable, error-free, and optimized data transfer between AHB and APB.
4. **Optimize Power and Performance.**
   To reduce power consumption by utilizing APB for peripheral communication while maintaining overall system performance.
5. **Validate Design with AMBA Standards.**
   To verify and test the design using HDL simulations, ensuring compliance with AMBA standards and proper functionality.

Methodology:

1. The methodology of designing an AHB to APB bridge begins with a thorough understanding of the AMBA AHB and APB protocols, including their signal definitions, timing characteristics, and operational differences. AHB is a high-speed, pipelined bus used for communication between processors and high-performance components, whereas APB is a low-speed, simple bus intended for peripheral devices. Understanding these differences is essential to ensure proper communication between the two buses.
2. Once the protocols are understood, the system architecture of the bridge is designed by dividing it into key modules such as the AHB interface, APB interface, and a control unit. The AHB interface handles communication with the master, while the APB interface communicates with peripheral devices. The control unit manages the overall operation and coordination between these two interfaces.
3. The next step involves signal mapping and conversion, where AHB signals like address, write control, and data are translated into corresponding APB signals. Since AHB supports pipelining and APB does not, proper care is taken to align the timing and ensure correct sequencing of operations. This conversion is crucial for maintaining data integrity during transfers.
4. A finite state machine (FSM) is then designed to control the entire data transfer process. The FSM typically includes states such as idle, setup, and enable, which follow the APB protocol sequence. It ensures proper synchronization between AHB and APB and controls when data should be read or written, thereby avoiding timing mismatches.
5. The design also incorporates logic for handling read and write operations efficiently. In write operations, data from the AHB side is captured and forwarded to the APB side, while in read operations, data from APB peripherals is sent back to the AHB master. Temporary registers or buffers may be used to hold data during these transfers.
6. Finally, the complete design is implemented using hardware description languages like Verilog or VHDL, followed by simulation and verification using tools such as ModelSim or Vivado. After successful verification, the design is synthesized and tested on hardware platforms like FPGA to validate its functionality, timing performance, and reliability in real-world applications.

Result and Conclusion:

1. **Result:** The designed AHB to APB bridge was successfully implemented using HDL and verified through simulation. The bridge effectively converts high-speed AHB transactions into low-speed APB transactions while maintaining proper timing and control. All major operations, including read and write transfers, were tested under different conditions and produced correct results. The finite state machine (FSM) operated as expected, ensuring smooth state transitions between idle, setup, and enable phases.
2. Simulation results confirmed that the bridge handled address decoding, signal mapping, and data transfer accurately without data loss or timing violations. The design showed efficient communication between AHB master and APB slave devices, demonstrating reliable performance. Additionally, resource utilization and timing constraints were within acceptable limits when synthesized on the target platform.
3. **Conclusion:** The AHB to APB bridge design successfully achieves communication between two different bus protocols with varying speeds and complexities. By using proper signal mapping, FSM control, and synchronization techniques, the bridge ensures efficient and error-free data transfer. The project highlights the importance of protocol compatibility and timing management in System-on-Chip (SoC) designs.
4. Overall, the implementation proves that the bridge can be effectively used to connect high-performance components with low-power peripheral devices. The design is scalable and can be further enhanced by adding features like burst transfer support, error handling mechanisms, and power optimization techniques, making it suitable for real-time embedded and VLSI applications.

References:

1. [1] ARM Ltd., *AMBA 3 APB Protocol Specification*, ARM Holdings, 2021.
2. [2]ARM Ltd., *AMBA 5 AHB Protocol Specification*, ARM Holdings, 2022.
3. [3] R. Kumar and S. Patel, “Efficient Implementation of AMBA AHB to APB Bridge Using Verilog,” *IEEE International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA)*, pp. 1–6, 2022.
4. [4] P. Singh and M. Kaur, “Low Power AHB to APB Bridge Design for Embedded Systems,” *IEEE Access*, vol. 11, pp. 45000–45010, 2023.
5. [5] S. Reddy et al., “FPGA Implementation of AMBA Based AHB to APB Bridge,” *IEEE International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 120–125, 2024.
6. [6] K. Sharma and A. Verma, “Design and Verification of AHB to APB Bridge for SoC Applications,” *IEEE International Conference on Electronics and Communication Systems (ICECS)*, pp. 1–5, 2021.

Project outcome & Industry Relevance:

- The AHB to APB bridge project provides an efficient method for communication between high-speed and low-speed buses in System-on-Chip (SoC) designs. It

ensures smooth data transfer by converting AHB protocol signals into APB protocol signals. This bridge helps in reducing power consumption, as APB is mainly used for low-power peripherals like UART, timers, and GPIO.

- In the semiconductor industry, such bridges are widely used in microcontrollers, embedded systems, and processors. It improves system performance by separating high-speed processing tasks from low-speed peripheral operations. The project contributes to VLSI design and embedded systems by demonstrating bus interfacing techniques and protocol conversion.
- In real-world applications, it is used in ARM-based processors, automotive electronics, IoT devices, and consumer electronics. This design can be implemented in FPGA or ASIC for practical hardware applications. Overall, it enhances understanding of digital system design and is highly relevant in modern chip design industries.

Working Model vs. Simulation/Study

- This project is primarily based on simulation and design implementation. The AHB to APB bridge was designed using hardware description languages such as Verilog or VHDL and verified through simulation tools like ModelSim or Xilinx Vivado.
- However, the design can also be extended to a working hardware model by implementing it on FPGA boards for real-time testing and validation.

Project Outcomes and Learnings

Key Outcomes:

- Successfully designed an AHB to APB bridge for protocol conversion.
- Achieved reliable data transfer between high-speed and low-speed buses.
- Verified functionality through simulation results.
- Improved system efficiency and reduced power usage in peripheral communication.

Learnings:

- Gained knowledge of AMBA bus architecture (AHB and APB protocols).
- Understood FSM (Finite State Machine) design for control logic.
- Learned Verilog or VHDL coding and simulation techniques.
- Improved debugging and waveform analysis skills.
- Developed understanding of real-time SoC communication systems.
-

Future Scope:

- **Enhanced Performance & Advanced Features:**
  Implement advanced features such as burst transfer support, pipelining optimization, and high-speed data handling to improve overall system performance and efficiency in complex SoC designs.
- **Low Power & Energy Optimization:**
  Incorporate low-power design techniques like clock gating and power gating to reduce energy consumption, making the bridge suitable for battery-operated and portable embedded systems.

- **Scalability & Multi-Protocol Support:**
  Extend the bridge design to support multiple APB slaves and integrate with other AMBA protocols like AXI, enabling scalability and wider applicability in modern System-on-Chip (SoC) architectures.
- **Real-Time Implementation & FPGA Optimization:**
  Deploy the design on advanced FPGA platforms with optimized resource utilization and timing constraints to achieve real-time performance in industrial applications.
- **Error Handling & Security Enhancement:**
  Add error detection and correction mechanisms along with security features to ensure reliable and safe data transfer, especially in critical embedded and automotive systems.