Design and Implementation of a Scalable AMBA AHB Interconnect

AET 005

Shriraj Siddhaye, Amrita Bhujbal, Noshi Chopra, Prerana Patil
Vidyalankar Institute of Technology, University of Mumbai, India
Email: {shriraj.siddhaye, amrita.bhujbal, noshi.chopra, prerana.patil}@vit.edu.in

***Abstract*—This paper presents a scalable 4×4 AMBA AHB (Advanced High-performance Bus) interconnect architecture designed for high-speed intra-chip communication in System-on-Chip (SoC) systems. The design supports multiple masters and slaves, with arbitration mechanisms—fixed priority and round-robin—to manage concurrent access. Using Verilog HDL and Vivado 2024.1 for simulation, this work validates the interconnect's correctness, pipelined performance, and protocol compliance. The results demonstrate minimal latency, high throughput, and effective resource utilization, establishing the design's relevance for VLSI and embedded systems.**

***Index Terms*—AMBA AHB, Bus Interconnect, Verilog, SoC, Arbitration, Pipelining, FPGA Simulation, Fixed Priority, Round Robin**

I. Introduction

The evolution of SoC architecture has driven the need for scalable and efficient on-chip communication protocols. The ARM-developed AMBA (Advanced Microcontroller Bus Architecture) protocol suite is widely used to facilitate communication between different IP blocks in SoCs. Among its various buses—APB, ASB, AXI, and AHB—the AHB bus strikes a balance between performance and design complexity. AMBA AHB (Advanced High-performance Bus) supports pipelining, burst transfers, and split transactions. It enables concurrent communication between multiple master and slave devices in a structured and standardized manner. The interconnect's arbitration mechanism plays a crucial role in maintaining fairness and avoiding bus contention.

This project implements a 4×4 scalable AMBA AHB interconnect using Verilog. It includes modules for master, slave, arbiter, and decoder and is simulated using Vivado 2024.1. The interconnect is evaluated through successive configurations: 1×1, 1×2, 2×1, 2×2, and finally 4×4.

II. Background

A. AMBA AHB Protocol Overview

The AHB protocol employs a shared communication bus where transactions are divided into two phases: an **address phase** and a **data phase**. It supports **pipelined** communication, allowing overlap between address and data transfers for increased throughput.

Each transfer is categorized as:

- **IDLE**: No transfer.
- **BUSY**: Bus is retained without transferring data.
- **NONSEQ**: Start of a new burst or single transfer.
- **SEQ**: Continuation of a burst.

Key signals include: 'HADDR', 'HWDATA', 'HRDATA', 'HWRITE', 'HTRANS', 'HSIZE', 'HBURST', 'HREADY', 'HRESP'.

B. Arbitration Mechanisms

Two major arbitration techniques are implemented:

- **Fixed Priority**: Masters are statically ranked. Starvation of lower-priority masters is a risk.
- **Round Robin**: Rotational access among requesting masters, ensuring fairness.

Advanced schemes such as dynamic arbitration are considered for future scope.

C. Literature Review

Prior studies [1]–[3] emphasize performance gains via pipelining and arbitration. Dynamic address mapping, low-latency designs, and FPGA-based verification have been used to optimize AHB systems. This project builds on such works by scaling to a 4×4 interconnect with practical simulation results.

III. Interconnect Architecture

A. 1×1 Master-Slave

Basic configuration with one master and one slave is demonstrated in fig. 1. It demonstrates a complete read/write transaction in two clock cycles (10 ns each). Signals are monitored for protocol adherence.

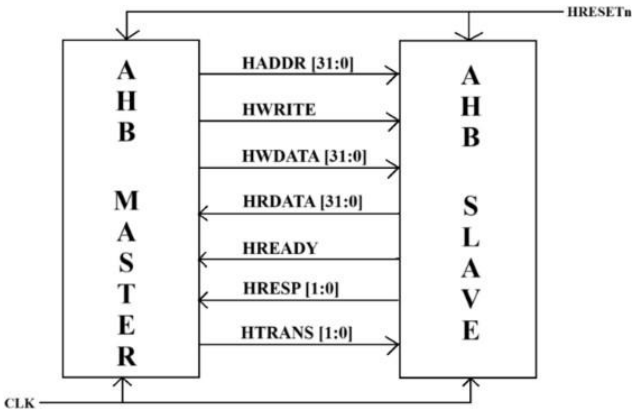

Fig. 1. 1x1 configuration

B. 1×2 Master-Slaves with Decoder

A decoder is introduced to select between two slaves based on address ranges:

- `0x000--0x3FF` → Slave 0
- `0x400--0x7FF` → Slave 1

Decoder asserts ‘HSEL0‘ or ‘HSEL1‘ accordingly.

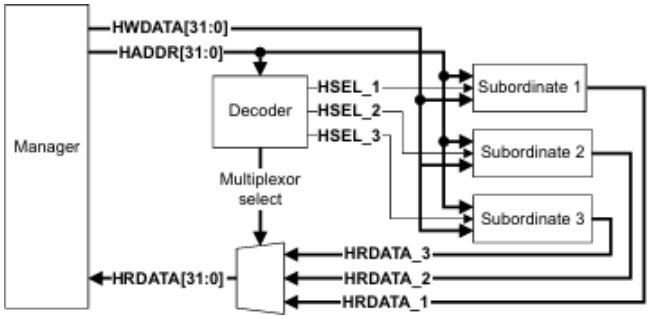

Fig. 2. 1x2 configuration

C. 2×1 Masters-Slave with Arbiter

Two masters communicate with a single slave. An arbiter handles access:

- **Fixed Priority**: Master0 > Master1.
- **Round Robin**: Alternating access with a priority counter (e.g., every 3 cycles).

A multiplexer routes the selected master’s address/data to the slave.

D. 2×2 Masters-Slaves with Arbiter and Decoder

Combines arbitration and decoding. Each master may attempt to communicate with either slave. This validates bus sharing, concurrent data flow, and proper address decoding.

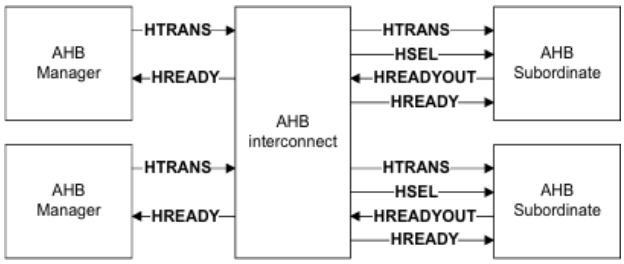

Fig. 3. 2x2 configuration

E. 4×4 Full-Scale Configuration

The system scales to four masters and four slaves:

- Slave 0: `0x000--0x3FF`
- Slave 1: `0x400--0x7FF`
- Slave 2: `0x800--0xBFF`
- Slave 3: `0xC00--0xFFF`

This tests the interconnect’s scalability, stress-handling, and fairness under load.

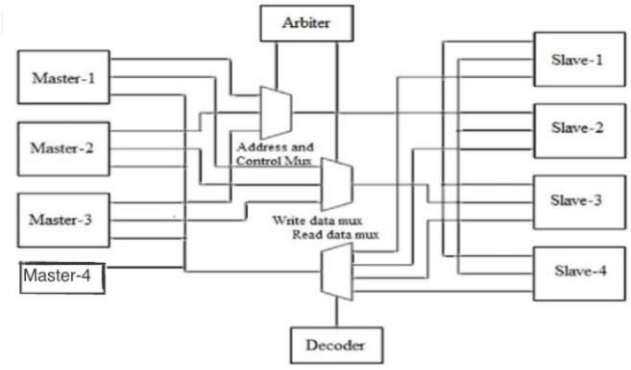

Fig. 4. 4x4 configuration

IV. Verilog Implementation

Each module is written in Verilog:

- **Masters**: Generate random read/write instructions with ‘HTRANS‘, ‘HADDR‘, ‘HWDATA‘.
- **Slaves**: Store and return data, using ‘HRESP‘ and ‘HREADY‘.
- **Decoder**: Monitors ‘HADDR‘ and activates the correct slave (‘HSELx‘).
- **Arbiter**: Implements logic for both arbitration schemes using FSMs.
- **Testbenches**: Simulate scenarios with pipelined reads/writes, simultaneous access, and edge cases.

The design is modular and synthesizable, except for the testbench.

V. Simulation and Results

Simulations were performed in Vivado 2024.1. The testbench verifies:

- **Correct timing**: Read/write operations finish in two clock cycles.
- **Protocol correctness**: Valid transitions through NONSEQ/SEQ, proper use of ‘HREADY‘/‘HRESP‘.
- **Arbitration effectiveness**:
  - Fixed priority showed starvation.
  - Round robin ensured equitable access.

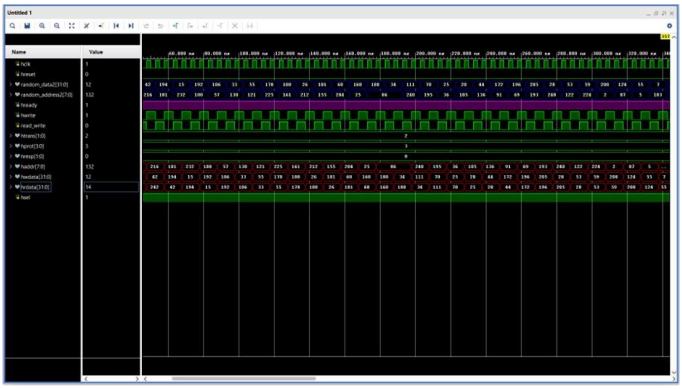

Fig. 5. 1 master 1 slave

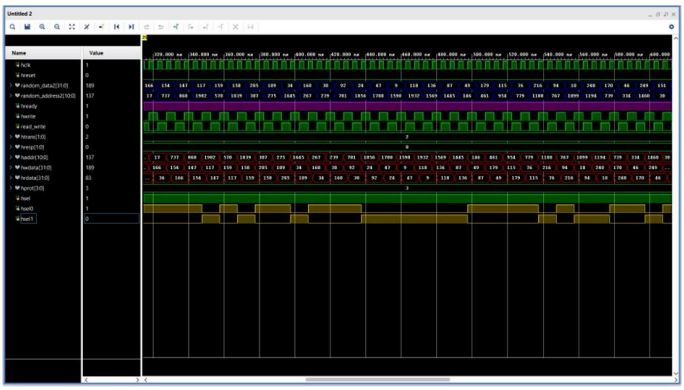

Fig. 6. 1 master 2 slaves

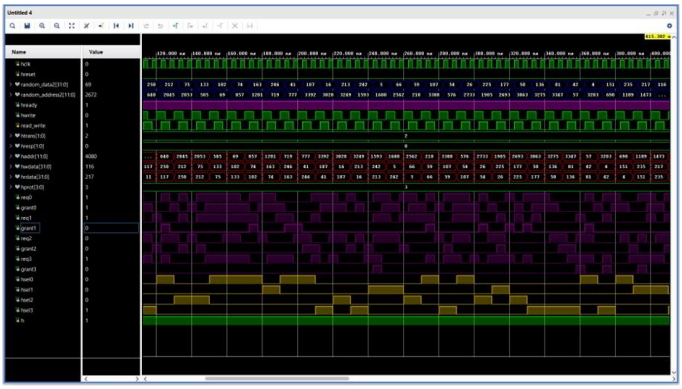

Fig. 7. 4 masters 4 slaves with fixed priority arbitration

Sample Observations:

- 1x1: Data write followed by correct readback.
- 1x2: Decoder activates appropriate slave.
- 2x1: Only one master granted access; contention resolved.
- 2x2: Multiple concurrent transactions; no collisions.
- 4x4: Continuous pipelined operations; no signal conflicts.

VI. Conclusion and Future Work

This work successfully implements a scalable 4×4 AMBA AHB interconnect for intra-chip communication. The system supports concurrent, pipelined transactions with efficient arbitration and address decoding. Simulation results affirm the design's correctness and throughput.

Future Enhancements:

- Add **split and burst transfers**.
- Incorporate **dynamic arbitration**.
- Implement on FPGA for real-time validation.
- Explore **AXI integration** for higher performance.

References

[1] S. Aithal et al., "FPGA Implementation of AHB to APB Protocol," *IJESC*, vol. 5, no. 5, 2016.

[2] S. Divekar, A. Tiwari, "Multichannel AHB with Arbitration," *IEEE ICGCCEE*, 2014.

[3] Y. Godhal et al., "Synthesis of AMBA AHB from Formal Spec.," *arXiv*, 2010.