AHB to APB Bridge - Verilog Implementation

- Project Objective

This project implements an AHB to APB Bridge in Verilog using Vivado. The goal is to design, simulate, and verify a bridge that connects a high-performance AHB master to low-power APB peripherals, ensuring protocol compliance and correct data transfer.

- AHB & APB Protocol Overview

AHB: High-speed, pipelined, burst-capable bus with signals like HADDR, HWRITE, HTRANS, etc.

APB: Simple, low-power, non-pipelined bus with signals like PADDR, PWRITE, PENABLE, etc.

- Block Diagram Description

The system includes an AHB Slave Interface, APB Controller with FSM, APB Interface as peripheral, Bridge Top module, and a Testbench AHB Master for verification.

- Design Details

FSM States: ST_IDLE, ST_READ, ST_WRITE, ST_WWAIT, ST_WENABLE, ST_RENABLE, ST_WRITEP, ST_WENABLEP.

Modules: AHB_M.v, AHB_slave_interface.v, APB_controller.v, APB_interface.v, Bridge_top.v, tb.v.

- Transaction Phases Explained

AHB initiates transaction using HTRANS, HADDR, etc.

APB side decodes and handles PSEL, PENABLE, etc.

HREADYOUT stalls AHB if needed. PRDATA is returned during the enable phase.

- Waveform Snapshots

Include screenshots for:

- Single Write

AHB to APB Bridge - Verilog Implementation

- Single Read

- Burst Write

(Screenshots not included in PDF)

- Tools Used

Vivado 2022.2, XSim, Verilog HDL, GTKWave (optional).

- Project Status

- RTL Design Complete

- Simulation Verified

- Protocol Compliant

- Debugged and Documented

- Author

Jaideep Patel

BTech in ECE | VLSI Design Intern @ Maven Silicon

LinkedIn: https://www.linkedin.com/in/YOUR_PROFILE

- Folder Structure

AHB_to_APB_Bridge/

- rtl/

- tb/

- doc/

- scripts/

- README.md

- .gitignore

AHB to APB Bridge - Verilog Implementation

- License

This project is for educational purposes only and does not include any proprietary IP from ARM or AMBA specifications.

Debug Log & Fix Summary

- Debug Log & Fix Summary

- 1. Port Redeclaration Error

Error: ansi port pwrite_out cannot be redeclared in the header in APB_interface.v

Fix: Cleaned up the port declaration to ensure each signal is declared only once.

- 2. Signal Name Mismatch

Error: Hwritereg1 used in Bridge_top.v but not declared.

Fix: Declared Hwritereg1 properly or removed the unused connection.

- 3. Clock Signal Inconsistency

Error: In tb.v, AHB_M was instantiated with Hclk instead of hclk.

Fix: Unified signal naming across modules to avoid mismatches.

- 4. Missing APB Enable Phase

Error: PENABLE was not asserted during the enable phase in APB transactions.

Fix: Updated FSM logic to assert penable_temp correctly during enable phases.

- 5. Incorrect Read Data Timing

Error: PRDATA was not showing expected value (0x19) during read.

Fix: Corrected the logic to return prdata = 25 only during valid read enable phase.

- 6. FSM Timing Misalignment

Error: State transitions were correct, but signal outputs (like PENABLE) were delayed.

Fix: Ensured each FSM phase (setup - enable - complete) lasted exactly one clock cycle and signals were updated synchronously.

- 7. Simulation Launch Failure

Error: 'launch_simulation' failed due to earlier errors

Fix: Resolved all syntax and structural issues, recompiled successfully.

- 8. Waveform Signal Absence

Error: Simulation object /tb/hedata was not found in the design

Fix: Updated waveform configuration and ensured all monitored signals were present in the design.

- Outcome
- Simulation ran successfully
- Waveforms showed correct protocol behavior
- FSM transitions aligned with APB protocol
- Read/write operations were verified
- Burst and pipelined transfers worked as expected