# DESIGN SPECIFICATION DOCUMENT

ECE-593: Fundamentals of Pre-Silicon Validation

Maseeh College of Engineering and Computer Science

Winter, 2026


Project Name: Design and Verification of AHB to APB Bridge

Members:

Amey Kulkarni - 933959421

Annapoorna Hamsagaru Manjunath - 984436960

Date: 03/31/2026

| | |
|---|---|
| Project Name | Design and Verification of AHB to APB Bridge |
| Location | EB – 92 |
| Start Date | 01/22/2026 |
| Estimated Finish Date | 03/09/2026 |
| Completed Date | 03/09/2026 |


| Prepared by: Team 2 | |
|---|---|
| Prepared for: Prof. Venkatesh Patil | |
| Team Member Name | Email |
| Amey Kulkarni | ameyk@pdx.edu |
| Annapoorna Hamsagaru Manjunath | annapoh@pdx.edu |


| Design Features: |
|---|
| 1. Provides an interface between the high-performance AHB bus and the low-power APB peripheral bus, which handles the necessary protocol conversions and signal timing. |
| 2. Implementation of separate read and write data paths with configurable data widths (32-bit standard), enabling efficient data transfer between the AHB and APB domains. |
| 3. Features an integrated address decoder that generates peripheral select signals (psel) based on the AHB address, simplifying peripheral interfacing and improving design modularity. |
| 4. Supports burst transfers, including incrementing burst operations (INCR4), allowing efficient data movement and improved throughput for sequential transactions. |
| 5. Supports zero-wait state single transfers for write operations and single-wait state transfers for read operations, optimizing the bridge performance while maintaining protocol compliance. |
| 6. Supports burst operation on the AHB side while managing the translation to individual APB transfers, maintaining efficient data throughput for the burst transactions. |
| 7. Includes built-in wait state generation and handling capabilities to manage timing differences between the AHB and APB domains when required for specific peripherals. |

8. Supports CDC for AHB (HClk) to APB (PClk) transfer conversions.

Project Description:

This AHB to APB bridge project comprises of both robust design implementation and comprehensive verification methodologies. It's key features are as follows-

- Features an advanced verification environment developed using SystemVerilog and Universal Verification Methodology (UVM), enabling thorough validation of the bridge functionality through sophisticated test scenarios and coverage metrics.
- Implements a fully compliant AMBA AHB-to-APB bridge interface that handles protocol conversion between the high-performance AHB bus and the low-power APB peripheral bus, supporting efficient system-on-chip integration.
- This design supports optimized transfer modes with zero wait state write operations and single-wait state read operations, ensuring efficient data movement while maintaining protocol integrity.
- Implements an address decoder for generating peripheral select signals (psel), along with separate read and write data paths supporting 32-bit width transfers.
- Has a dedicated Verification IP (VIP) components that facilitate comprehensive testing of all bridge operations, including address decoding, data transfers, and protocol compliance verification.
- The verification environment leverages Synopsys VCS for simulation and DVE for waveform analysis, ensuring thorough validation of all design features and corner cases.
- Project development follows a structured approach, starting from basic functionality verification and progressing to complex test scenarios, ensuring robust operation across various use cases.

Design Signals

Input Signals:

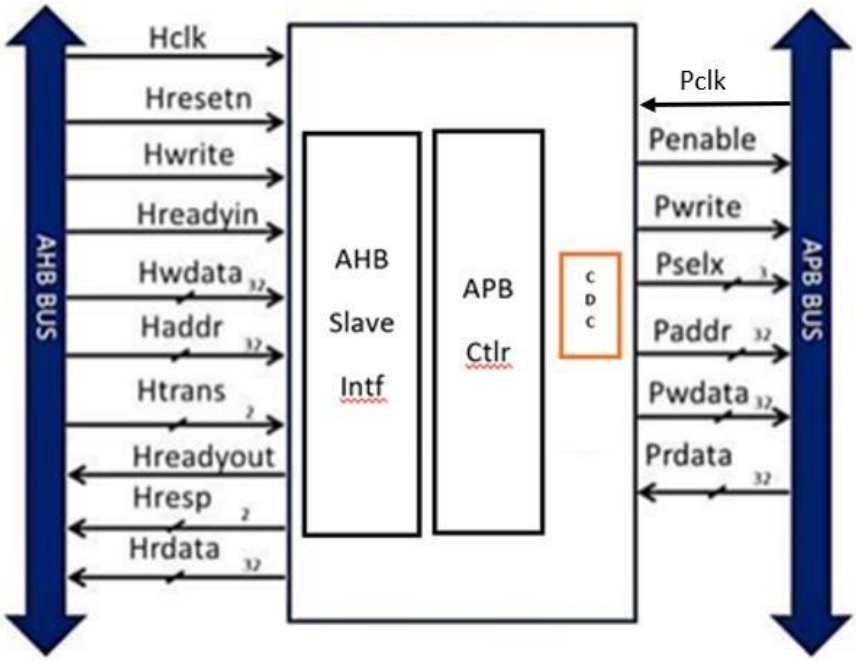
1. hclk- Clock signal for the AHB interface.
2. Pclk - Clock signal for the APB interface.
3. hresetn-Activelow reset signal for the AHB interface.
4. hreadyin - Ready signal indicating the availability of the AHB interface.
5. hsize[2:0] -Size of the transfer on the AHB interface.
6. hburst[2:0] - Specifies the type of burst operation (INCR4).
7. haddr[31:0] -32-bit address for the AHB transfer.
8. htrans[1:0] -Transfer type on the AHB interface (NONSEQUENTIAL, SEQUENTIAL, IDLE or BUSY).
9. hwrite - When HIGH this signal indicates a write transfer, and when LOW, a read transfer.
10. hwdata[31:0] -32-bit write data on the AHB interface.
11. prdata[31:0] - Read data on the APB interface.

Output Signals:

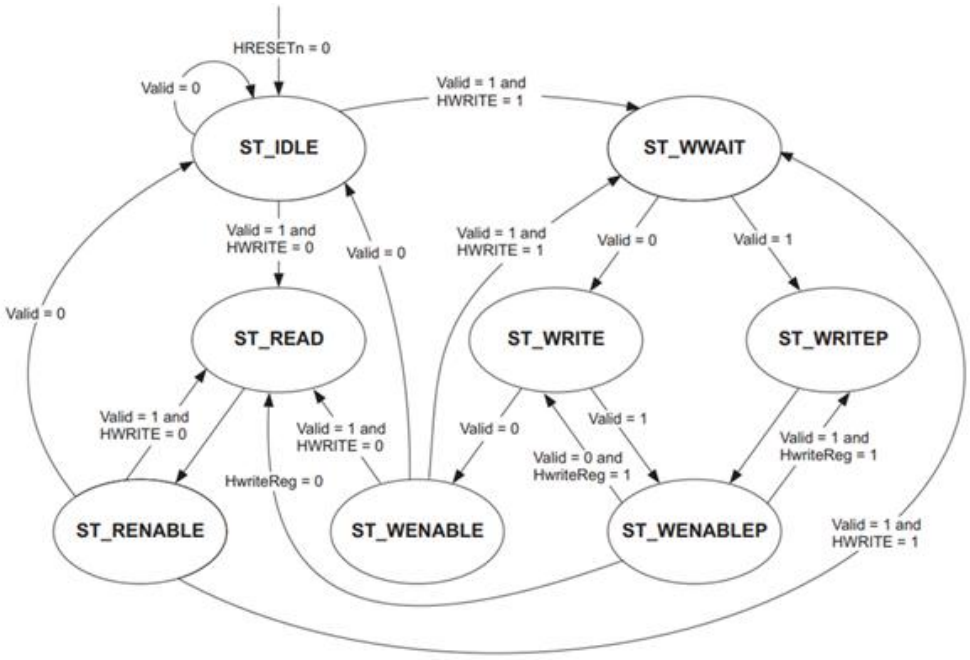
1. psel[2:0] -The signal indicates that the slave device is selected, and that a data transfer is required. It has the same timing as the peripheral address bus. It becomes HIGH at the same time as PADDR, but will be set LOW at the end of the transfer.
2. penable - Enable signal for the APB interface.
3. paddr[31:0]-32-bit address for the APB transfer.
4. pwrite -This signal indicates a write to a peripheral when HIGH, and a read from a peripheral when LOW. It has the same timing as the peripheral address bus.
5. pwdata[31:0]–32-bit write data on the APB interface.
6. hrdata[31:0]-32-bit read data from the AHB interface.
7. hreadyout - Ready signal indicating the readiness of the AHB interface.
8. hresp[1:0] - Response indicating the status of the AHB transfer.

Block Diagram

AHB to APB Bridge-




State machine-



## References/Citations


1. https://chatgpt.com/
2. https://developer.arm.com/documentation/ihi0033/latest
3. https://github.com/VyshnaviChilukamukku/AHB-to-APB-Bridge-Verification
4. Design and Implementation of a Synchronous AHB to APB Bridge in System-on-Chip Applications - Hongjin Chen
5. https://dev.to/sahanagonalnagaraja/designing-and-verifying-an-ahb-to-apb-bridge-my-learning-experience-24ao