

DESIGN SPEC DOCUMENT

ECE-593: Fundamentals of Pre-Silicon Validation

Maseeh College of Engineering and
Computer Science
Winter, 2025



Project Name: Design and Verification
of AHB to APB Bridge using UVM

Members:

Raghavendra Davarapalli	-	955118446
Maithreyi Venkatesan	-	900119348
Neil Austin Tauro	-	956372423
Suraj Vijay Shetty	-	967611331

Date: 01/31/2025

Project Name	Design and Verification of AHB to APB Bridge using UVM
Location	WCC-310
Start Date	01/27/2025
Estimated Finish Date	03/01/2025
Completed Date	03/04/2025

Prepared by: Team 14	
Prepared for: Prof. Venkatesh Patil	
Team Member Name	Email
Raghavendra Davarapalli	ragha@pdx.edu
Maithreyi Venkatesan	maithven@pdx.edu
Neil Austin Tauro	ntauro@pdx.edu
Suraj Vijay Shetty	surajvs@pdx.edu

Design Features:

The AHB to APB bridge implements the following key features:

- Provides a seamless interface between the high-performance AHB bus and the low-power APB peripheral bus, handling the necessary protocol conversions and signal timing requirements between the two interfaces.
- 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.
- Implements separate read and write data paths with configurable data widths (32-bit standard), enabling efficient data transfer between the AHB and APB domains.
- Features an integrated address decoder that generates peripheral select signals (psel) based on the AHB address, simplifying peripheral interfacing and improving design modularity.
- Includes built-in wait state generation and handling capabilities to manage timing differences between the AHB and APB domains when required for specific peripherals.
- Supports burst operation on the AHB side while managing the translation to individual APB transfers, maintaining efficient data throughput for burst transactions.
- Supports burst transfers, including incrementing burst operations (INCR4), allowing efficient data movement and improved throughput for sequential transactions.

Project Description:

The AHB to APB bridge project encompasses both robust design implementation and comprehensive verification methodologies. The key features include:

- 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.
- 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.
- Incorporates dedicated Verification IP (VIP) components that facilitate comprehensive testing of all bridge operations, including address decoding, data transfers, and protocol compliance verification.
- The 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 a sophisticated address decoder for generating peripheral select signals (psel), along with separate read and write data paths supporting 32-bit width transfers.
- The verification environment leverages QuestaSim for simulation and 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:

Design Input Signals:

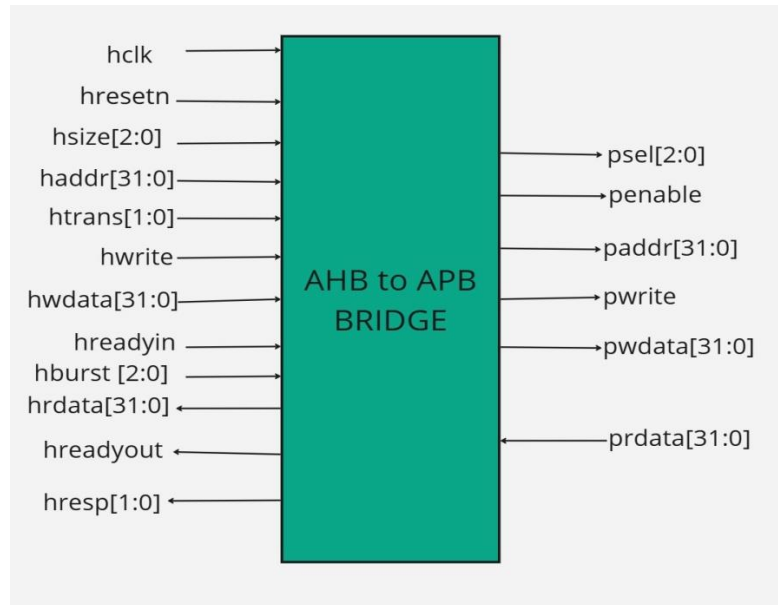
1. hclk- Clock signal for the AHB interface.
2. hresetn-Active low reset signal for the AHB interface.
3. hsize[2:0] -Size of the transfer on the AHB interface.
4. hburst[2:0] - Specifies the type of burst operation (INCR4).
5. haddr[31:0] -32-bit address for the AHB transfer.
6. htrans[1:0] -Transfer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

Design Output Signals:

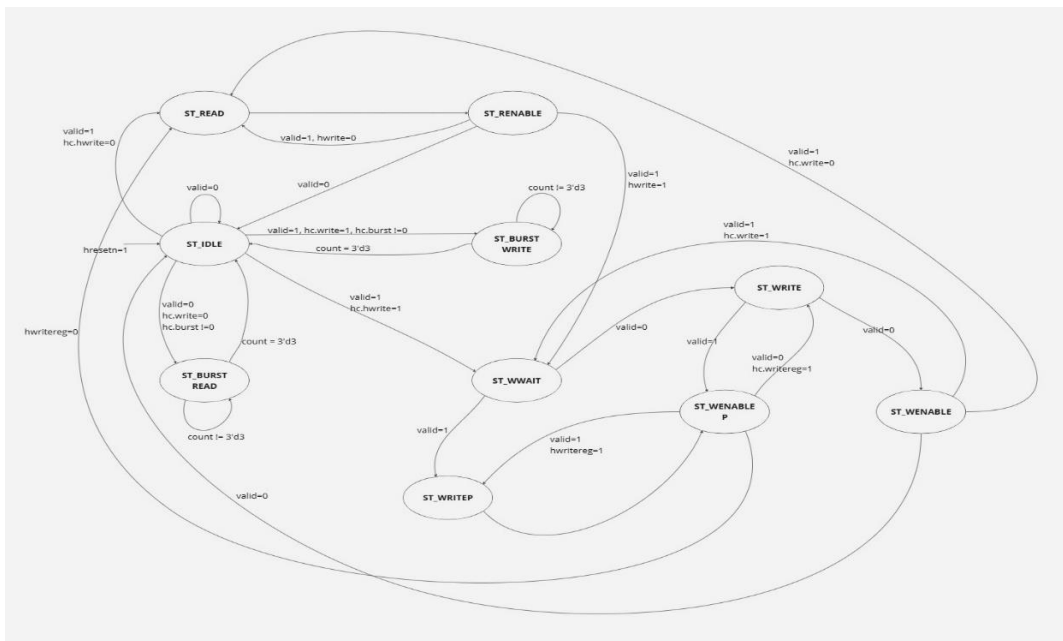
1. hrdata[31:0]-32-bit read data from the AHB interface.
2. hreadyout - Ready signal indicating the readiness of the AHB interface.
3. hresp[1:0] - Response indicating the status of the AHB transfer.
4. psel[2:0] -The signal indicates that the slave device is selected, and that a data transfer is required. 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.
5. penable - Enable signal for the APB interface.
6. paddr[31:0]-32-bit address for the APB transfer.
7. 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.
8. pwdata[31:0]-32-bit write data on the APB interface.

Block Diagram

AHB to APB Bridge:



State Machine for AHB2APB Interface:



To verify the fundamental accuracy of the selected code, QuestaSim is used to compile the design files, and the process completed successfully without any errors or issues. Additionally, a basic testbench simulation was created to analyze the waveform produced by the design, and a screenshot of the results is attached below.

[illegible]

```
# Top level modules:
# top
# End time: 18:35:39 on Jan 31, 2025, Elapsed time: 0:00:01
# Errors: 0, Warnings: 0
# vlsim -v -m -w -t "testbench" work top
# Start time: 18:35:39 on Jan 31, 2025
# ** Note: (vlsim-1812) Design is being optimized...
# ** Note: (vopt-143) Recognized 1 FSM in module "APB_FSM_Controller(fast)".
# Loading sv_std.vcd
# Loading work_top(fast)
# Loading work_apb2ahb_interface(fast_1)
# Loading work_bridge(fast)
# Loading work_ahb_master(fast)
# Loading work_ahb_slave_interface(fast)
# Loading work_apb_fsm_controller(fast)
# Loading work_apb_interface(fast)
# Time: 0 | state = 0000 valid = 0 Size=0 Burst=0 Address=00000000 HAddress=xxxxxxxx Write_Data=00000000 Trans=0 Write/Read=0 penable=0 Ready=1 hrdta=00000000 prdata=xxxxxxxx
# -----Performing single write-----
# Time: 25 | state = 0000 valid = 1 Size=0 Burst=0 Address=00000000 HAddress=00000001 Write_Data=00000000 Trans=0 Write/Read=0 penable=0 Ready=1 hrdta=00000000 prdata=xxxxxxxx
# Time: 35 | state = 0001 valid = 0 Size=0 Burst=0 Address=00000000 HAddress=00000001 Write_Data=00000000 Trans=0 Write/Read=0 penable=0 Ready=1 hrdta=00000000 prdata=xxxxxxxx
# Time: 45 | state = 0011 valid = 0 Size=0 Burst=0 Address=00000001 HAddress=00000001 Write_Data=00011111 Trans=0 Write/Read=0 penable=0 Ready=1 hrdta=00000000 prdata=xxxxxxxx
# Time: 55 | state = 0111 valid = 0 Size=0 Burst=0 Address=00000001 HAddress=00000001 Write_Data=00011111 Trans=0 Write/Read=1 penable=1 Ready=1 hrdta=00000000 prdata=xxxxxxxx
# -----Performing single read-----
# Time: 65 | state = 0000 valid = 1 Size=0 Burst=0 Address=00000001 HAddress=00000001 Write_Data=00011111 Trans=0 Write/Read=1 penable=0 Ready=1 hrdta=00000000 prdata=xxxxxxxx
# Time: 75 | state = 0000 valid = 1 Size=0 Burst=0 Address=00000001 HAddress=00000002 Write_Data=00011111 Trans=0 Write/Read=1 penable=0 Ready=1 hrdta=00000000 prdata=xxxxxxxx
# Time: 85 | state = 0010 valid = 0 Size=0 Burst=0 Address=00000002 HAddress=00000002 Write_Data=00000000 Trans=0 Write/Read=0 penable=0 Ready=1 hrdta=00000000 prdata=xxxxxxxx
# Time: 95 | state = 0101 valid = 0 Size=0 Burst=0 Address=00000002 HAddress=00000002 Write_Data=00000000 Trans=0 Write/Read=0 penable=0 Ready=1 hrdta=00000000 prdata=00000024
# Time: 105 | state = 0000 valid = 1 Size=0 Burst=0 Address=00000002 HAddress=00000002 Write_Data=00000000 Trans=0 Write/Read=0 penable=0 Ready=1 hrdta=00000024 prdata=00000024
# -----Performing burst write-----
# Time: 125 | state = 0000 valid = 1 Size=2 Burst=3 Address=00000002 HAddress=00000002 Write_Data=00000000 Trans=2 Write/Read=0 penable=0 Ready=1 hrdta=00000024 prdata=00000024
# Time: 135 | state = 1001 valid = 1 Size=2 Burst=3 Address=0000000C HAddress=0000000C Write_Data=00000000 Trans=3 Write/Read=0 penable=0 Ready=1 hrdta=00000024 prdata=00000024
# Time: 145 | state = 1001 valid = 1 Size=2 Burst=3 Address=0000000C HAddress=0000000C Write_Data=00000000 Trans=3 Write/Read=0 penable=1 Ready=1 hrdta=00000024 prdata=ffffff01
# Time: 155 | state = 1001 valid = 1 Size=2 Burst=3 Address=0000000C HAddress=0000000C Write_Data=00000000 Trans=3 Write/Read=0 penable=1 Ready=1 hrdta=ffffff02 prdata=ffffff01
# Time: 165 | state = 1001 valid = 1 Size=2 Burst=3 Address=0000000C HAddress=0000000C Write_Data=00000000 Trans=3 Write/Read=0 penable=1 Ready=1 hrdta=ffffff03 prdata=ffffff01
# Time: 175 | state = 0000 valid = 0 Size=2 Burst=3 Address=0000000C HAddress=0000000C Write_Data=00000000 Trans=3 Write/Read=0 penable=1 Ready=1 hrdta=ffffff04 prdata=ffffff01
# -----Performing burst write-----
# Time: 215 | state = 0000 valid = 1 Size=2 Burst=3 Address=0000000C HAddress=0000000f Write_Data=00000000 Trans=2 Write/Read=0 penable=1 Ready=1 hrdta=ffffff04 prdata=ffffff01
# Time: 225 | state = 1010 valid = 1 Size=2 Burst=3 Address=0000000f HAddress=0000000f Write_Data=00000000 Trans=3 Write/Read=0 penable=1 Ready=1 hrdta=ffffff04 prdata=ffffff01
# Time: 235 | state = 1010 valid = 1 Size=2 Burst=3 Address=0000000f HAddress=0000000f Write_Data=00011111 Trans=3 Write/Read=0 penable=1 Ready=1 hrdta=ffffff04 prdata=ffffff01
# Time: 245 | state = 1010 valid = 1 Size=2 Burst=3 Address=00000003 HAddress=00000003 Write_Data=00011112 Trans=3 Write/Read=0 penable=1 Ready=1 hrdta=ffffff04 prdata=ffffff01
# Time: 255 | state = 1010 valid = 1 Size=2 Burst=3 Address=00000007 HAddress=00000007 Write_Data=00011111 Trans=3 Write/Read=1 penable=1 Ready=1 hrdta=ffffff04 prdata=ffffff01
# Time: 265 | state = 0000 valid = 1 Size=2 Burst=3 Address=00000010 HAddress=000000ff Write_Data=00011114 Trans=0 Write/Read=1 penable=1 Ready=1 hrdta=ffffff04 prdata=ffffff01
# -----Notes Setup / testbench.vcd-----
# Time: 365 ns Iteration=0 Instance/ top
# Break in Module top at testbench.v line 60
```

References/Citations
<ol style="list-style-type: none">1. https://github.com/prajwalgekkouga/AHB-to-APB-Bridge2. https://chatgpt.com/3. https://developer.arm.com/documentation/ih0033/latest