Module: SV For Verification

Section: Final Project **Task:** Verification of AHB-3 Lite

Verification of AHB3 Lite Protocol Using SystemVerilog Layered Testbench

Project Links :

1. EDA Link

2. Google Drive Link

Abstract:
This report presents the verification of the AHB3 Lite protocol using a SystemVerilog layered testbench. It describes the verification plan, test cases, and methodology applied to ensure compliance with the AHB3 Lite protocol standards. The verification environment, including testbench architecture, stimulus generation, and scoreboard implementation, is discussed, followed by an evaluation of the results.

1. Introduction

- **1.1Background:**
  The Advanced High-performance Bus (AHB) protocol family, developed as part of ARM's Advanced Microcontroller Bus Architecture (AMBA), is designed for high-speed, high-bandwidth interconnects in System-on-Chip (SoC) designs. AHB facilitates efficient communication among various components, including processors, memory, and peripherals. Among its variants, AHB3 Lite stands out as a simplified version that meets the performance demands of modern applications while minimizing design complexity. By omitting features such as multiple master support and split transactions, AHB3 Lite enables lower pin counts and easier integration, making it particularly suitable for a range of embedded and high-performance systems.

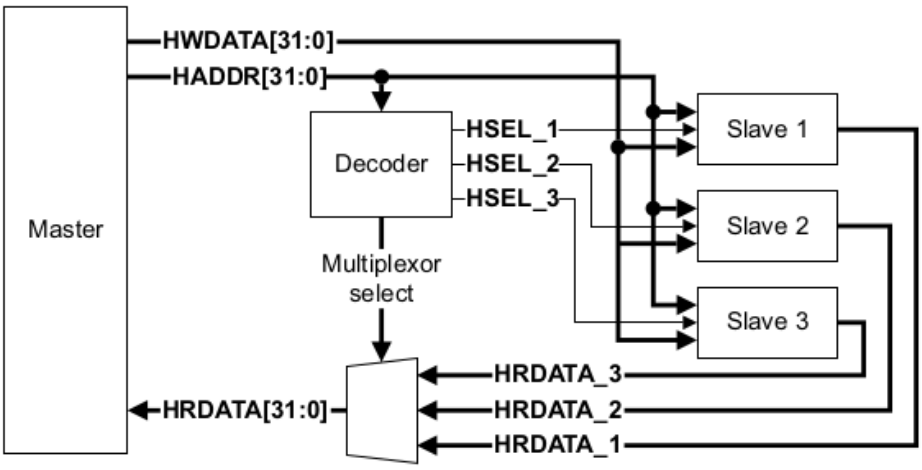


Figure 1-1 AHB-Lite block diagram

AHB3 Lite operates in a single-master configuration, allowing straightforward connections between the master device—typically a processor—and slave devices like memory and peripherals. Despite its simplifications, AHB3 Lite maintains the capability for high-speed transfers, including burst transfers, ensuring it can handle the demands of various applications effectively. Its efficient design makes it ideal for memory interfaces and peripheral communication, delivering a robust solution for resource-constrained embedded systems while providing the necessary performance for high-speed operations.

- **1.2_AHB3_Lite_Overview:**
  Here is the diagram given

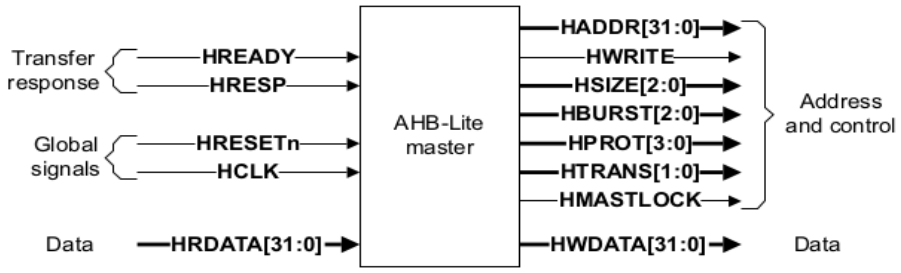


Figure 1-2 Master interface

Design Input Signals :

- **HCLK:** Clock signal for the AHB interface.
- **HRESETn:** Active-low reset signal for the AHB interface.
- **HSIZE[2:0]:** Size of the transfer on the AHB interface.
- **HADDR[31:0]:** Address for the AHB transfer.
- **HTRANS[1:0]:** Transfer type on the AHB interface.
- **HWRITE:** Write enable signal for the AHB interface.
- **HWDATA[31:0]:** Write data on the AHB interface.
- **HREADYin:** Ready signal indicating the availability of the AHB interface.

Design Output Signals :

- **HRDATA[31:0]:** Read data from the AHB interface.
- **HREADYout:** Ready signal indicating the readiness of the AHB interface.
- **HRESP[1:0]:** Response indicating the status of the AHB transfer.

Slave Diagram

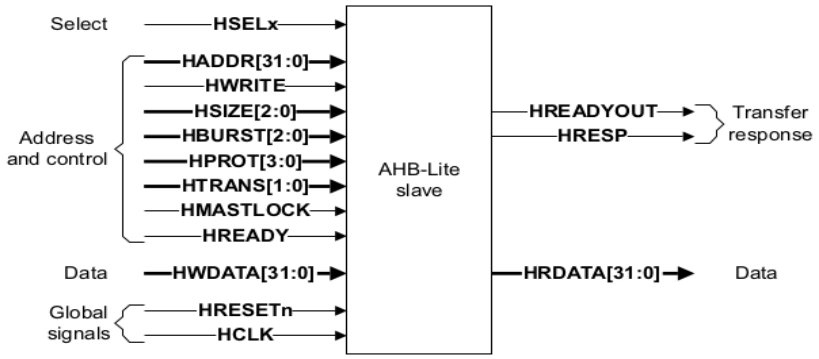


Figure 1-3 Slave interface

- **1.3 Importance of Verification:**.

  Robust verification of AHB3 Lite-based designs is paramount due to their critical role in System-on-Chip (SoC) architectures, where reliability and performance directly impact overall system functionality. As AHB3 Lite interconnects various components within an SoC, ensuring that each module adheres to the protocol specifications is essential for maintaining data integrity and system stability. Inadequate verification can lead to protocol compliance issues, resulting in data corruption, incorrect transfer sequences, or even system failures during operation. Given that these designs often serve in safety-critical applications, thorough verification processes are crucial to mitigate risks and ensure that all components function seamlessly together.

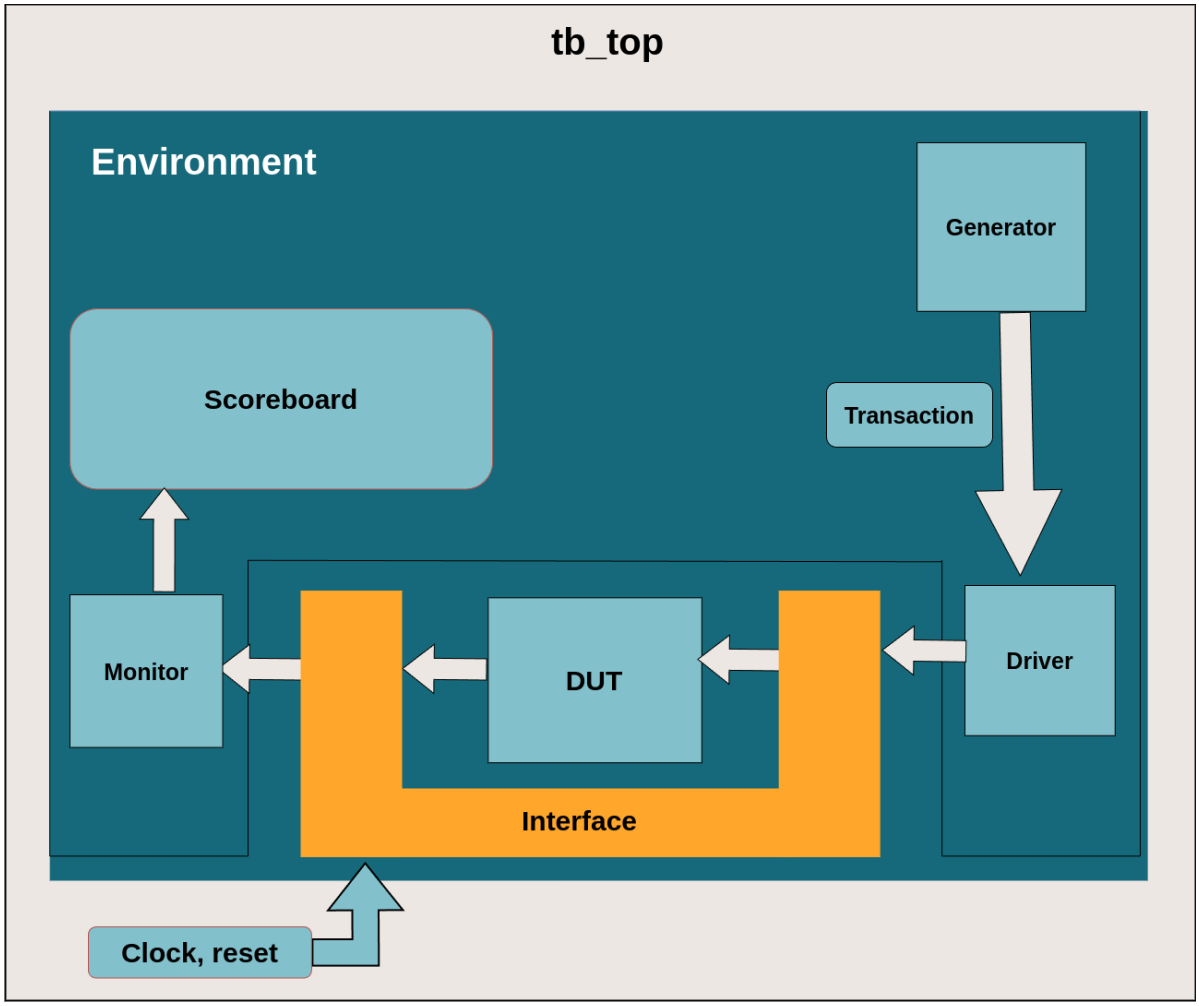
  Additionally, interoperability between different AHB3 Lite components can pose significant challenges, particularly in multi-vendor environments where different design implementations may vary. Common issues include mismatches in signal timing, data widths, and response behaviors, which can lead to functional discrepancies and degraded system performance. Effective verification strategies, including simulation and formal methods, are necessary to uncover these issues early in the design process, ensuring that the components not only comply with AHB3 Lite specifications but also work harmoniously in a larger SoC context. By prioritizing rigorous verification, designers can enhance the reliability and efficiency of their AHB3 Lite-based systems, paving the way for successful deployment in a variety of applications.

2. Verification Methodology

- **2.1 Verification Plan:**

  Verification plan can be found here:

- **2.2 Layered Testbench Architecture:**
  Here are the components of the layered testbench



including:

- **2.2.1 Interface Module:** The AHB interface connecting to the Design Under Test (DUT), including signal details.
- **2.2.2 Driver:** Responsible for driving stimulus to the DUT based on generated transactions.
- **2.2.3 Monitor:** Observes the interface for protocol adherence and captures responses.
- **2.2.4 Generator:** Generates randomized or constrained inputs.
- **2.2.5 Scoreboard:** Compares expected and actual outputs for verification.

3. Verification Plan and Test Scenarios

3.1 Functional Test Cases

The following functional test cases were developed to ensure the integrity and reliability of basic read, write, and burst transfer operations:

- **3.1.1 Single Write and Read Transfers**:
    - Test cases such as **test_001** (Basic Read Data) and **test_002** (Basic Write Byte Sized) verify the accuracy of data transactions at single byte and half-word levels. The tests ensure that data written to a specified address can be correctly read back, confirming data integrity. Specifically, **test_001** checks for correct `HRDATA` retrieval when a read operation is performed, while **test_002** validates the successful writing of `HWDATA` to the memory.
- **3.1.2 Burst Transfers**:
    - Burst transfer functionality is assessed through tests like **test_006** (WRAP4 Burst Transfer Test for Word), **test_007** (Increment 4 Burst Test for Byte Transfers), and **test_008** (Increment 4 Burst Test for Half-Word Transfers). These tests examine the DUT's ability to handle sequential and non-sequential burst operations, verifying that data consistency is maintained across multiple addresses during burst transactions.
- **3.1.3 Error Scenarios**:
    - Test cases such as **test_004** (Reset Single Transfer Test) simulate potential error conditions, such as read/write operations during reset state. They ensure that the system responds appropriately when the slave device is inactive or when there is an attempt to access memory under invalid conditions.

3.2 Corner Case Test Cases

To address boundary conditions and edge cases, the following tests were included:

- **Minimum and Maximum Burst Lengths**: Tests such as **test_006**, **test_007**, and **test_008** investigate burst transfers of varying lengths, including the maximum number of beats allowed.
- **Idle States**: **test_005** (HSELx Test) examines the system's behavior when the slave is not active and verifies that no unintended operations occur.
- **Varying Response Delays**: Tests ensure that the DUT can handle variations in response time during read/write operations without data corruption.

Conclusion

6.1 Summary

The verification effort successfully validated the operational integrity of the DUT through a comprehensive suite of functional, corner case, and performance tests. The results confirm that the memory interface meets the design specifications and is capable of reliable operation in various scenarios.

6.2_Future_Work:

To enhance the verification environment for the AHB3 Lite protocol, several potential improvements can be considered. Firstly, adding coverage metrics would provide insights into the effectiveness of the test cases by identifying untested scenarios and ensuring all aspects of the protocol are exercised adequately. This could include functional coverage for various transaction types and state transitions, which would help in assessing the completeness of the verification effort.

Additionally, the verification of 8-beat wrapping burst transfers has not been conducted and presents an opportunity for further testing. Implementing specific test cases to evaluate the performance and correctness of 8-beat wrapping burst operations will ensure that this important functionality is validated. These enhancements will contribute to a more robust and comprehensive verification process for the AHB3 Lite protocol.