IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, October 2024

Performance Evaluation of AMBA-3 AHB-Lite Protocol Verification: Techniques and Insights

Hiranmaye Sarpana Chandu
Independent Researcher
chanduhiranmaye9@gmail.com

Abstract: *Verification is a crucial stage in SoC manufacturing so that the DUT has met specifications required by its user. It offers the DUT the specific implementation and functionality for the same purpose. The SoC design has the manufacturing capability, but it does not meet time to market requirements. The verification process, which looks into the right or wrong connection of the AHB LITE in the SoC design, has come out as one of the most strategic areas of concern in the design approach especially with increasing SoC designs. For instance, AMBA 3 AHB LITE is useful in SoC design that requires only one master, one or more slaves or several slaves. This paper discusses the AMBA 3 AHB-Lite protocol with more emphasis on design simulation and also the architecture of the testbench. AMBA in its first implementation presents two main buses, the "Advanced High-performance Bus (AHB)" for central control IPs and the "Advanced Peripheral Bus (APB)" for peripheral IPs. This paper focuses on the AHB-Lite subset as it is implemented to have high-bandwidth operations with a single bus master. It describes the features of AHB-Lite in terms of data transfer types and phases. The paper also expands on the verification environment of the AHB-Lite interconnects evaluating the testbench architecture, which encompasses the generators and master and slave agents and, drivers and scoreboards. The efficient creation of reusable testbenches and management of randomization is underscored by the Universal Verification Methodology (UVM). This paper shows how UVM helps to improve verification by finding corner cases and reporting coverage data. The paper concludes with an evaluation of simulation results, illustrating the effectiveness of the AHB-Lite protocol and its verification framework.*

Keywords: AHB-Lite, UVM, AMBA 3, Verification, SoC.APB

I. INTRODUCTION

The process from the architectural or RTL level all the way to the chip layout can be mapped out by a good SOC design flow. The designer will use this as a starting point for their work [1]. Electronic systems that are heavily embedded and function in contexts with limited resources are known as system-on-chips, or SoCs. These systems are well-known for their ability to compete with high-performance CPUs from the past, but with much reduced energy usage and costs [2][3]. The CPU bus protocol is a crucial component of co-verification as it controls data transfer between the memory, CPU, and any specialised hardware. ARM processors employ many time-dependent bus protocols due to the flexibility of a SoC architecture; thus, to identify the best design for achieving the goals, a multi-dimensional design space analysis is necessary[4][5].

Modern "Micro Controller Bus Architecture (AMBA)" bus protocols are an amalgamation of various ARM standards. The AMBA protocol is based on the idea of connecting these different components while enabling their reuse in various configurations. An understanding of the AMBA Protocol, its uses, and how they all contribute to system-on-chip (SoC) architectures should be your primary objective[1]. The most popular on-chip bus architecture in the majority of SoCs is the AMBA (Advanced Microcontroller Bus architecture) bus. It was ARM that initially introduced AMBA. For high-performance synthesizable designs, AMBA has you covered with their AHB-Lite synthesizable design. Data transfers among masters and slaves are accelerated by AHB because it allows for system modules with high clock frequencies[6][7]. This article focuses on describing AHB-lite in detail and briefly presents different AMBA procedures. Furthermore, discuss the concept of design verification, its importance in the industry, and UVM.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, October 2024

The rationale for assessing the performance of an AMBA 3 AHB-Lite protocol derives from its functionality in advanced SoC architectures where reliable and high-bandwidth data transfer are paramount. With further development of SoC architectures, it is vital to pay special attention to reliability and lack of instability in bus interfaces of a new level such as AHB-Lite. This paper seeks to propose a solution to this challenge of verifying such complex protocols by analysing both basic and sophisticated verification methods such as the UVM. In this way, it aims at improving the effectiveness of the verification, reveal possible challenges and, finally, check if the AHB-Lite protocol works as expected in practice.

- Analysis of both basic and improved approaches towards verifying AMBA 3 AHB-Lite.
- This paper presents an assessment of the conventional and superior approaches to verifying AMBA 3 AHB-Lite using UVM methodologies.
- It describes the AHB-Lite transfer types and burst operations and provides valuable tips about the UVM testbench structure and elements.
- The study also covers the comparison between the current approach and the use of universal verification methodology in terms of randomization, assertion and overall coverage.

A. Structure of the paper

The following is the format of the paper: Section I introduces the AMBA 3 AHB-Lite protocol. Section II covers the AMBA protocols overview. Section III details AHB Lite interconnect verification. Section IV explores the AMBA 3 AHB-Lite protocol. Section V presents a case study on its verification system. Relevant literature is reviewed in Section VI, and important conclusion are presented in Section VII.

II. OVERVIEW OF AMBA PROTOCOLS

An open standard that has been around since 1996, the AMBA Bus enables any SoC makers to utilise processors based on the Arm architecture. In the AMBA design, there are two types of buses: one that connects to the central IPS and is known as the advanced high-performance bus, and another that is known as the flexible advanced peripheral bus that is connected to a myriad of other components and peripherals. The "AMBA Advanced High-Performance Bus (AHB)" is often used for linking DSPs, processors, and memory controllers to various peripheral IPS. A bridge between the APB and AHB buses is another feature of AMBA that makes it adaptable for updating SOCS by swapping out appropriate IPs with a newer design (like FM receiver) or version (like Bluetooth module). Bus-to-bus connectors called bridges enable IPS to link several communication [8]. Architecture of AMBA bus displayed in Figure 1:

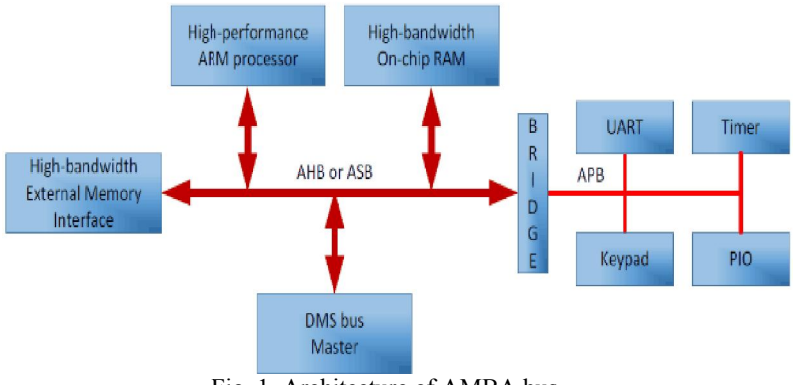

Fig. 1. Architecture of AMBA bus.

A subset of the full AHB standard, AHB-Lite provides high-bandwidth operation and is best used when there is just one bus master [4].

A. Operations Of AHB-LITE

The transfer begins when the master drives the control and address signals. The address, direction, and breadth of the transmission, as well as whether it is a burst, are transmitted by these signals. Transfers could be:[11]

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, October 2024

- Single
- Increasing the bursts that don't wrap around address borders
- Bursts of wrapping that stop at certain address boundaries.

TABLE I: Transfer Types[9].

Transfer Type	Description	HTRANS [1:0]
IDLE	No data transmission is necessary in this instance.	00
BUSY	Here, let masters access the ability to insert idle state throughout a cycle.	01
NON-SEQUENTIAL	Here, the control signal and address are randomly generated and have nothing to do with the prior transfer.	10
SEQUENTIAL	Here, the control signal and the address are associated with a prior transfer.	11

Read data buses allow data to be delivered from slaves to masters, while write data buses allow data to be updated. Included in every step are: Topics covered include the first phase and the control cycle. At least one data phase round[1].

III. AHB LITE INTERCONNECT VERIFICATION

The testbench architecture must be understood from a broader viewpoint in order to comprehend the AHB LITE interconnect verification environment. The testbench architecture consists of numerous components that are properly integrated to form a verification intellectual property (VIP). Three components make up the full verification process: 1. Test 2. Verification environment 3. Functional examination.

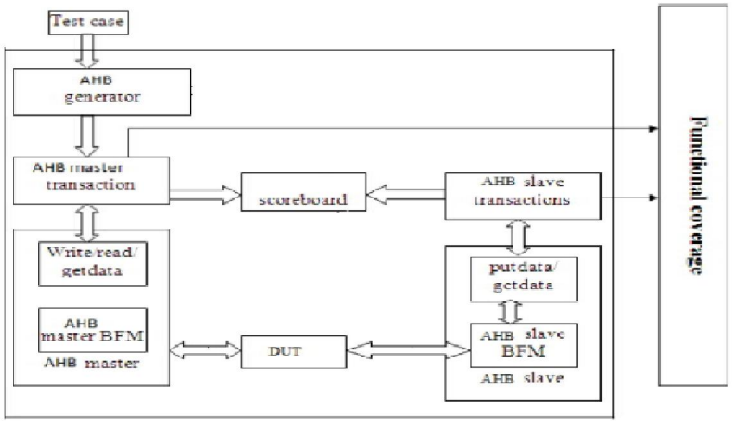

Fig. 2. Architectural block diagram for AHB lite interconnect

Figure 2 depicts the AHB LITE interconnect architecture block diagram, which comprises the test case, environment, and functional verification.

A. Test

To evaluate if a feature is functioning as intended, a test case is essentially a file that describes an action, event, input, and matching (anticipated) response. As the verification environment is being designed, it is imperative that modelling the design input and output and creating a memory to hold these tests be started. The generator in the testbench architecture is then tasked with processing these test cases.

B. Verification Environment

The functional correctness of the DUT must be confirmed by the verification environment, which generates and drives the input sequence and records the design's output for comparison with the golden (anticipated) result. To carry out the particular work in the verification environment, several components are used; these components are shown in Figure 2.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, October 2024

- **Generator:** System Verilog provides a mechanism to control the order and distribution of random generators, and generators produce test cases at random. The master agent receives these test instances.
- **Master agent:**The link between the generator, scoreboard, LITE driver, and functional coverage is occasionally referred to as a connector or transaction. Additionally, it manages the DUT's setup operations and translates high-level data obtained from the generator (burst transmission) into a specific instruction.
- **Master driver:** It receives information from the master agent and uses it to drive the DUT by transforming it into the DUT's real input. It gets a response from the tested design and then transforms it back into object format before sending it on to the master agent. Just as a master agent and driver, a slave driver and agent do the same job.
- **Scoreboard:** It stores the predicted output in a reference model and goes by another name, tracker. When the generator sends out random stimuli to the DUT, the scoreboard receives the identical signals, and the process continues until the DUT produces an output. The last step in the verification process is for the scoreboard to compare the real and golden outputs.

C. Functional coverage

It is the one that shows the extent to which the design's functionality is addressed. Functional verification includes aspects such as:

- Specifies which verification feature was successfully verified.
- The specific feature that was verified, together with the degree of completion of the verification.
- Offers assurance that errors repaired in the past won't recur in the design.

IV. AMBA 3 AHB-LITE PROTOCOL

To meet the criteria for synthesizable designs with good performance, one uses the AMBA-3 AHB-Lite protocol. With this bus interface, you may run at greater bandwidths while still supporting a single bus master. High clock frequency, high performance systems, such the ones listed below, may have their characteristics extracted via the AHB-Lite protocol.

- Burst transfers
- Non tristate operation
- Wide data bus configuration such as 64, 128, 256, 512 and 1024 bits
- Single clock edge operations

Popular AHB-Lite slaves include peripherals with higher bandwidth, memory interfaces external to the system, and internal memory devices. While AHB-Lite slaves normally remain on the AMBA-APB with limited bandwidth peripherals, a specific type of APB slave called an APB bridge was employed to cross the upper level of the bus and increase system performance[8].

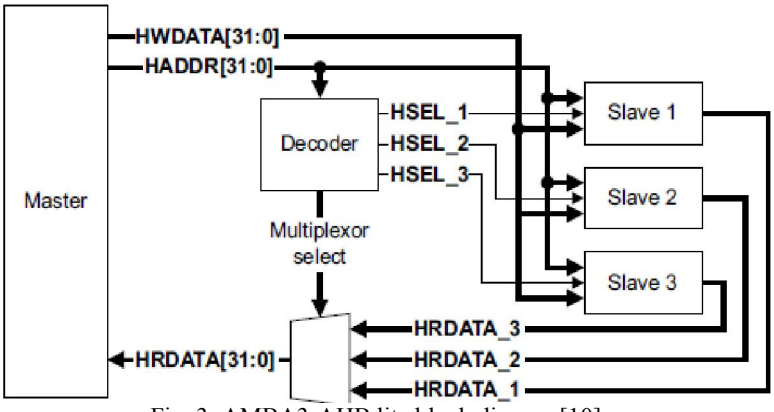

Fig. 3. AMBA3-AHB lite block diagram[10]

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, October 2024

Figure 3 shows a one-master, three-slave AHB-Lite system. Bus circuit logic contains an address decoder and slave-to-master multiplexor. Decoders help choose slaves by monitoring the master's address. Based on each slave's and master's respective outputs, the multiplexor transfers data between them. The schematic of AHB-Lite is shown in Figure 3[11].

- **AHB LITE Master:** Initiating write and read operations requires control and address information, which is provided by the AHBLITE master, a device attached to the high throughput bus. The constant 32-bit transfer size of the AHB LITE connection is maintained.
- **AHB LITE Slave:**An AHB LITE master writes data and addresses to the memory component of the system-on-chip (SoC) called the AHB LITE slave, which reads data from its memory location. Without the AHB LITE master, the AHB LITE slave cannot start the operation. The AHB LITE slave replies to the master address decoder's signal by sending HRDATA and a transfer response signal back to the master host.
- **Address Decoder:** This component identifies the AHB LITE slave that was engaged in a given transfer and sends a unique signal to that slave based on the decoded address. It is used when there are more than two slaves in the design. An address decoder receives the AHB LITE master's addresses and selects a slave based on them.
- **Multiplexer:** If there are more than two AHB LITE slaves, the information may be multiplexed using the multiplexer, and the slaves can then send a response signal to the AHB LITE master.

V. CASE STUDY ON VERIFICATION SYSTEM OF AMBA 3 AHB LITE PROTOCOL

As crucial as designing is, verification plays a crucial role in chip testing and production. Verification offers the actual functioning and execution of a "Design under Test (DUT)" and determines whether or not it complies with the requirements. This study presents the verification of an AMBA 3 AHB LITE protocol in accordance with the design requirements. Verification techniques have changed throughout time, with the advisor being the first in 2000 and the UVM being the most recent in 2013. The UVM methodology is a standard methodology that was developed in 2013 by Synopsys, Cadence, and Mentor Graphics and is cited in Accellera. Code reuse is one of the main characteristics of UVM, whereby common code is abstracted away in modules and added to the library. Once the library name is included, these modules may be used with ease using a common syntax. Consequently, UVM shortens the time needed to validate a chip by lessening the verification engineer's workload[12].

A. UVM Testbench Architecture

We must first have a more comprehensive understanding of the architecture before we can comprehend the verification environments. The architecture is made up of several parts that are correctly combined to generate the necessary VIP. These parts consist of a DUT (Design Under Test), a sequence item, a sequencer, a monitor, a driver, an agent, a collector, and a scoreboard. As seen in Figure 4 below, these parts may be joined to one another.

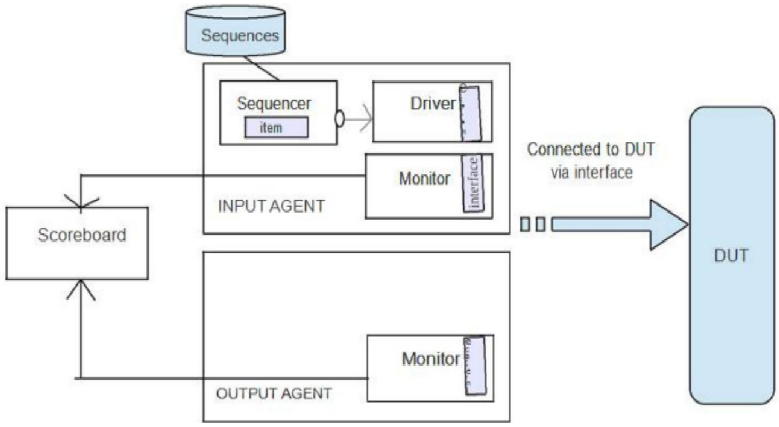

Fig. 4. UVM Testbench Architecture[13]

The following are examples of some of the components utilised in the UVM:

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, October 2024

- **Driver:**One of the VIP creation parts. Sequencers handle driver signal and data transaction demands. Following processing during the run phase, these transactions are sent over an interface to the DUT for verification. Every basic driver we develop extends from the uvm_driver base class, which is a parameterised class with two parameters: response (RES) and request (REQ). These arguments belong to the sequence item type, which is really a transaction. Where many parallel phases are taken into consideration, pipelining may also be included in this case. Prior to adding further phases like the run phase, we first establish a virtual interface that is utilized for communication with a design that is being verified.
- **Sequencers and Sequences:** It is a part that creates random sequences that are subsequently required for DUT verification. Sequence components found in this component are utilised together to build sequences. The port and export mechanisms enable the sequencer to send the sequences it generates to the driver. In order to produce the necessary transactions for DUT verification, several restrictions are imposed on the sequencer in order to make the link with these sequence items.
- **Monitor:** It is a passive component that serves two purposes: it checks and provides coverage by doing verification and gathering transactions from the interface. In this instance, transactions are sampled and then written to the analysis port. It is nearly the same as the driver portion of the code.
- **Agents:** These elements serve as a container for the driver, sequencer, and monitor and may be either active or passive. Active mode allows the agent to produce the driver, sequencer, and monitor; passive mode restricts it to constructing the monitor. These elements are integrated in the correct and necessary way to create the whole environment.

A. Compilation In UVM

We need to first get familiar with the different UVM stages in order to comprehend the architecture. These are:

- **The build phase:** this is the highest stage, where all the parts needed to construct the UVM environments are built.
- **The connect phase:** the construction step, which involves connecting the assembled components, comes next. Export and analysis ports might be applied here.
- **The end_of_elaboration_phase:** Activities that follow elaboration are classified below. We may publish topological information for example.
- **The start_of_simulation_phase:** the constructed components are created at this point.
- **The runphase:** this stage can be compared to a task that is in progress over some time.
- **The extract, check and report phase:** these are the last stages that are carried out.

B. Design Simulation Process

Randomization is very important in design simulations because it is the primary means of finding obscure corner cases that a directed test might not come across Users may provide constraints in a succinct and declarative manner using Universal Verification Methodology (UVM). This makes it possible for a solver to generate arbitrary values that satisfy the constraints and thus make the testing process more robust. This output waveform depicts the signals from internal data processing through AMBA 3 AHB LITE signals and the signals generated by the DUT. Coverage metrics are very important when evaluating the outcome of the simulation. Percentage of code coverage, percentage of assertion coverage, and total percentage inform the analyst on how comprehensive testing has been done on the design. Assertions are used in HDL languages such as VHDL and Verilog; they are significant for confirming certain behaviours and observing that a design departs from expectations. Assertion coverage, which emphasizes the fact that randomization helps to meet constraints. A cumulative figure of total coverage percentage is a sum of assertion and functional coverage, where functional aspects are checked using cover points and bins within the cover group.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, October 2024

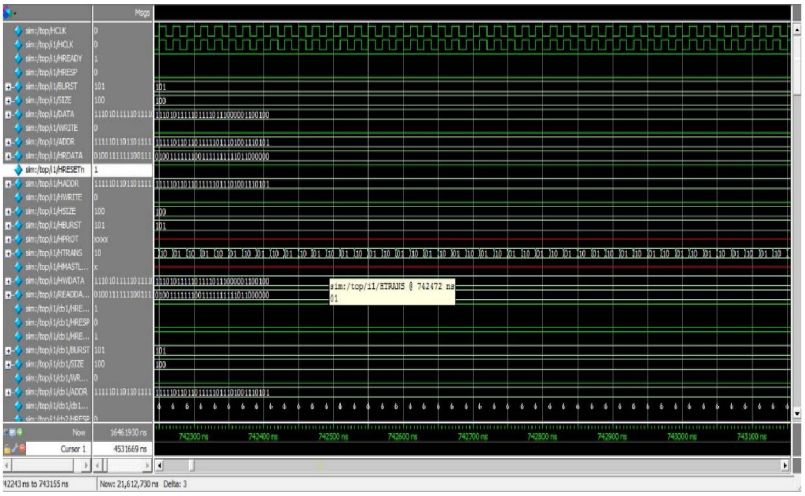

Fig. 5. Output Waveform of AMBA 3 AHB LITE Protocol

All of the signals being created by the DUT are represented in the output waveform, as seen in Figure 5. It is also possible to observe the internal registers processing data and showing the results using AMBA 3 AHB LITE signals.

VI. LITERATURE REVIEW

The following section provide a literature review on AMBA- AHB-Lite Protocol Verification.

In,Patel, Soni and Mehta, (2022) provide the conventional AMBA a modern upgrade with the General AHB Lite. The AHB Protocol is the foundation of this high-performance system-on-chip design[1].

In,Madl et al., (2006) analyses and evaluates SoC designs based on AMBA and makes two key contributions to this field. The first contribution provides a method that can be used to analyze and evaluate the effectiveness of AMBA-based SoC architectures using formal models. This way, the end-to-end execution boundaries for SoC architectures using AMBA can be specified more accurately when using this technique. This dissertation compares AMBA-based SoC architectures and delivers two key findings. We provide formal model-based analysis and assessment recommendations for AMBA-based SoC designs. However, when the SoC designs are based on AMBA, this method proves effective in making accurate judgments regarding the end-to-end execution limits[2].

In, Hegde and Singh, (2012) details the architectural modelling of the AMBA-AHB-Lite subset of AHB. The focus here is on the AHB-Lite slave model, which, via a number of test cases, demonstrates how fast and accurate the simulations are when utilising the right simulator Riviera and the comprehensive semantic support of the standard language System Verilog[4].

In,Singh, Shrivastava and Tomar, (2011) details the verification and design of the AMBA 3 AHB Lite Protocol. ARM introduced AMBA first. This article provides an overview of the AMBA subset known as AHB-Lite at a system level. The AHB-Lite protocol for sequential and non-sequence transfers (increment and wrap of varying burst sizes) is also being developed and tested. The Bus protocols should be considered. For example, AMBA's AHB stands out from its siblings APB and AXI thanks to its innovative burst transfers and other characteristics that boost bandwidth[6].

In,S, (2021) have studied the specifications, features, and design of the AMBA AHB-lite protocol, which outlines its features and method of operation, in order to get an understanding of how these AMBA protocols are used in microcontrollers. The need of verification, its development, and the benefits of the UVM over traditional verification techniques have also been discussed. As it allows the modules to communicate with one other, the AMBA protocol is the most popular choice for SOC designs and microcontrollers[7].

In,Kante, Kakarla and Yadlapati, (2016) explains how the AMBA-AHB-Lite subset of AHB is system level modelled. Particularly for the often-used communication libraries, there is pressure to develop models that execute more quickly. Designing for high-performance synthesizable designs is what AMBA AHB-Lite does. Elite data transmission capability and support for a single transport ace are provided by this transport interface[9].

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, October 2024

In,Shrivastava, Tomar and Singh, (2011) cite a research that compared the performance of many AMBA SOC bus protocols. The ARM protocol known as AMBA. A quick overview of the AMBA 2.0, 3.0, and 4.0 protocols is given at the outset. Advanced features including split transfers, burst transfers, and pipelining are included into these protocols[14].

In,Kommineni et al., (2023) presented a memory controller design based on the idea of a single master and several slaves, compliant with the AMBA 3 AHB_Lite standard. Functional coverage and a System Verilog verification environment allowed us to make sure the design complied with ARM's requirements. The testbed environment's various components, including a transaction and generator for creating input stimuli, a driver for sending data to the design under test (DUT), a monitor for tracking signals coming from the DUT, and a scoreboard for reporting on the design's operational status, are used to test the AHB Lite protocol using 4, 8, and 16 beat increment bursts, wrapping, and single bursts with waited transfer responses[15].

In,Shanavas et al., (2024) focus on APB and the AHB Lite protocols. The AMBA consists of the APB and the AHB. Unlike the AHB, APB follows the non-pipelined convention. Inside the IC communication takes place between the master and the slave using both the models. Both APB and the AHB deal with the communication itself. The basic understanding of the APB and the AHB protocol is dealt with here and how efficiently the data is transmitted from the master to the slave is seen[16].

Table II summarises key aspects of various studies on the performance evaluation and verification of the AMBA-3 AHB-Lite protocol.

TABLE II: Summarising related work on verification of the AMBA-3 AHB-Lite protocol

Reference	Methodology	Key Findings	Drawbacks	Future Work
[1]	Examining AHB Lite for SoC Design	AHB Lite supports high performance with single bus master and multiple slaves.	Limited focus on advanced verification techniques.	Incorporate advanced verification strategies to enhance robustness.
[2]	Performance Analysis with Formal Models	Provides end-to-end execution bounds and proves functional correctness, uncovering potential deadlocks.	May not cover all practical scenarios and edge cases.	Extend models to address more complex scenarios and ensure wider applicability.
[4]	System Verilog Simulation	Evaluates AHB-Lite slave model simulation speed and accuracy using System Verilog.	Focuses primarily on simulation aspects without extensive verification environment details.	Develop comprehensive verification environments for enhanced coverage.
[6]	Design and Verification of AHB-Lite Protocol	Describes the development, testing, and verification of AHB-Lite for various transfer types.	Mainly focuses on design aspects, less on advanced verification methods.	Explore advanced verification techniques and case studies.
[7]	Review of AMBA Protocols and UVM	Reviews AMBA protocols and the benefits of UVM over traditional methods.	Limited case study depth and practical application details.	Apply UVM in real-world scenarios for validation and performance enhancement.
[9]	System Level Modeling and Verification	Covers modeling and verification of AHB-Lite for high-performance designs.	Speed and efficiency issues in simulation models.	Improve model efficiency and expand verification scenarios.
[14]	Comparative Study of AMBA	Compares AMBA 2.0, 3.0, and 4.0, highlighting	Less focus on AHB-Lite specific	Detailed analysis of AHB-Lite verification

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, October 2024

	Protocols	advanced features like pipelining and burst transfers.	verification.	and performance metrics.
[15]	Memory Controller Design and Verification	Implements AHB-Lite standard and verifies using System Verilog with functional coverage.	May not address all practical edge cases.	Expand testing to cover a broader range of operational conditions.
[16]	APB and AHB Lite Protocol Analysis	Analyzes APB and AHB protocols, focusing on data transfer efficiency.	Basic overview with limited depth in AHB-Lite verification.	Detailed study of AHB-Lite data transfer and integration with APB.

VII. CONCLUSION

The AMBA 3 AHB-Lite protocol should be validated to ensure optimization in the SoC design and avert any failure in the system. Thus, the given paper, using UVM to analyze randomization and the completeness of coverage, shows how these parameters can contribute to improving the verification process. Finally, the AMBA 3 AHB-Lite protocol with high-frequency bus operation and more accessible structure well fits single-master systems. These elements are generators, agents, and scoreboards, which are an integrated part of the testbench architecture that allows for evaluating the protocol's performance effectively. This aspect calls attention to precise simulation and coverage analysis pertinent to any possible problems and adherence to design benchmarks. Altogether, the utilization of UVM and other methods can help achieve higher efficiency and accuracy in the tests, which in turn positively affect the ability to implement highly effective SoCs.

However, there are some challenges encountered when dealing with the AMBA 3 AHB-Lite protocol verification as it is described below. A core restriction is the ability to define a large number of test cases that would contain all possible conditions and edge cases. This challenge can result into inadequate coverage and more so reduced chances of identifying edge cases. Also, for as powerful as UVM is, its usage can be demanding on the resources needed and it usually takes time and skill to set up properly. Future work should encompass improvement of methods for generating more effective tests and the improvement of algorithms that will make the testing process more efficient. Studies into the application of machine learning techniques for adaptive test case generation and coverage prediction could extend the existing knowledge and create a foundation for more profound and successful high-performance SoC design verification.

REFERENCES

[1] D. Patel, B. Soni, and R. Mehta, "Verification of AHB Lite Bus Protocol : A High Performance AMBA Bus in Verilog," *IJCRT*, 2022, [Online]. Available: https://ijcrt.org/papers/IJCRT_193030.pdf

[2] G. Madl, S. Pasricha, L. A. D. Bathen, N. Dutt, and Q. Zhu, "Formal performance evaluation of AMBA-based system-on-chip designs," in *IEEE International Conference on Embedded Software, EMSOFT 2006*, 2006. doi: 10.1145/1176887.1176932.

[3] K. Ullah *et al.*, "Ancillary services from wind and solar energy in modern power grids: A comprehensive review and simulation study," *J. Renew. Sustain. Energy*, vol. 16, no. 3, 2024, doi: 10.1063/5.0206835.

[4] M. S. Hegde and S. Singh, "M Odelling and V Erification of E Xtensible a Uthentication P Rotocol Using Spin," vol. 4, no. 6, pp. 81–98, 2012.

[5] V. R. Sandeep Gupta, Puneet Matapurkar, Priyanka Gupta, "INTEGRATION OF SOLAR AND WIND ENERGY: A REVIEW OF CHALLENGES AND BENEFITS," *J. Emerg. Technol. Innov. Res.*, vol. 10, no. 3, pp. e604–e609, 2023.

[6] A. K. Singh, A. Shrivastava, and G. S. Tomar, "Design and implementation of high performance AHB reconfigurable arbiter for onchip bus architecture," *Proc. - 2011 Int. Conf. Commun. Syst. Netw. Technol. CSNT 2011*, no. 1, pp. 455–459, 2011, doi: 10.1109/CSNT.2011.99.

[7] S. S, "A Review on AMBA AHB Lite Protocol and Verification using UVM Methodology," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 2, pp. 473–481, 2021, doi: 10.22214/ijraset.2021.33120.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, October 2024

[8] A. Pattedar and S. V. Siddamal, "Design and Verification of AMBA 3 AHB Lite Protocols by using GO2UVM Package," *Int. J. Eng. Res. Electron. Commun. Eng.*, vol. 3, no. 10, pp. 6–13, 2016, [Online]. Available: https://technoarete.org/common_abstract/pdf/IJERECE/v3/i10/2.pdf

[9] S. Kante, H. K. Kakarla, and A. Yadlapati, "Design and verification of AMBA AHB-lite protocol using verilog HDL," *Int. J. Eng. Technol.*, 2016.

[10] S. Kante, H. K. Kakarla, and A. Yadlapati, "Design and verification of AMBA AHB-lite protocol using verilog HDL," *Int. J. Eng. Technol.*, vol. 8, no. 2, pp. 734–741, 2016.

[11] 1-Pallavi T Lambe and 2-Meghana Kulkarni, "Functional Verification of AMBA AHB LITE Interconnect using Systemverilog," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 6, no. 7, pp. 5420–5426, 2017, doi: 10.15662/IJAREEIE.2017.0607032.

[12] S. Sutherland, S. Davidmann, and P. Flake, *System verilog for design: Second edition: A guide to using system verilog for hardware design and modeling*. 2006. doi: 10.1007/0-387-36495-1.

[13] P. Agarwal, "AMBA 3 AHB LITE PROTOCOL Verification through an Efficient and Reusable Environment with an Optimum Assertion and Functional and Code Coverage in UVM," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 4, no. 1, 2016, doi: 10.15680/IJIRCCE.2016. 0401111.

[14] A. Shrivastava, G. S. Tomar, and A. K. Singh, "Performance comparison of AMBA bus-based system-on-chip communication protocol," in *Proceedings - 2011 International Conference on Communication Systems and Network Technologies, CSNT 2011*, 2011. doi: 10.1109/CSNT.2011.98.

[15] A. Kommineni, M. K. Gundu, Y. Kim, and S. Jadhav, "Design & Verification of AMBA AHB-Lite Memory Controller," in *2023 IEEE 13th Annual Computing and Communication Workshop and Conference, CCWC 2023*, 2023. doi: 10.1109/CCWC57344.2023.10099257.

[16] I. H. Shanavas, P. A. Reddy, M. M. Baburaj, N. C. Krishna, and N. S, "Design and Analysis of APB and AHB Lite Protocol," pp. 1–6, 2024, doi: 10.1109/icdi3c61568.2023.00009.