Design and Implementation of AXI4 Master UVC and Slave UVC Using UVM

Nisha S[1], Jalaja S[2]
[1]*MTech Student, Dept of ECE, Bangalore Institute of Technology, India*
[2]*Associate Professor, Associate Dean, Department of Electronics Engineering (VDT), Bangalore Institute of Technology, India*
doi.org/10.64643/IJIRTV12I4-184334-452

***Abstract*—This paper presents design of reusable AXI4 Master Slave Universal Verification Components (UVCs) using the Universal Verification Methodology (UVM) for efficient verification of AXI4 protocol-based System-on-Chip (SoC) design. The UVM structure was selected due to its modularity, scalability, and reusability, making it well-suited for complex protocol verification. The developed UVCs support all AXI4 transaction types, including FIXED, INCR, and WRAP bursts, aligned/unaligned accesses, narrow transfers, and out-of-order transactions. The verification environment integrates sequencers, drivers, monitors, functional coverage, and a scoreboard to ensure protocol compliance and data integrity. Both directed and constrained-random test cases were applied to achieve thorough coverage across diverse scenarios. Simulation results demonstrate higher functional coverage (83%), showing that the proposed verification environment is both robust and efficient. By providing a reusable and scalable AXI4 verification framework, this work contributes to faster verification closure and supports integration into large SoC-level environments. The QuestaSim tool was used for all simulations.**

***Index Terms*—AMBA, AXI, Functional Coverage, Handshake Mechanism, Master UVC, QuestaSim, Slave UVC, SoC, UVM (Universal Verification Methodology), VIP (Verification IP)**

I. INTRODUCTION

The integration of multiple processing units, memory blocks, and peripherals into a single chip has made the System-on-Chip (SoC) a dominant design paradigm in modern electronics. To enable efficient data exchange within these complex systems, robust and scalable on-chip communication protocols are required. Among the protocols in the AMBA family, the Advanced eXtensible Interface (AXI4), is now the most popular due to its ability to support high bandwidth, low latency, and flexible interconnect architectures.

The Universal Verification Methodology (UVM) has become the industry standard for functional verification, offering modularity, reusability, and scalability. By adopting constrained-random stimulus generation, functional coverage metrics, and reusable verification components, UVM provides an efficient way to verify complex protocols like AXI4. Fig.1 shows the standard UVM testbench architecture adopted for this verification environment.

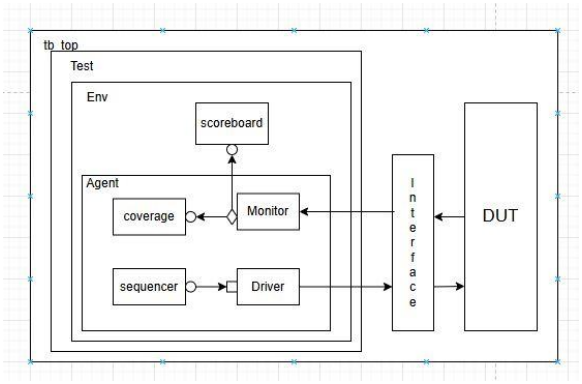

Fig.1. UVM Testbench Architecture

The UVM testbench consists of essential components such as driver, sequencer, and monitor, organized into AXI Master and Slave agents. These agents interact with the AXI interface signals, while functional coverage and a scoreboard are used to check protocol compliance and ensure verification completeness.

AXI4 provides several advanced features such as independent read/write channels, burst-based transfers, unaligned data handling, out-of-order transactions, and multiple outstanding transfers. These features improve throughput and design flexibility, but they also make functional verification of AXI4-based designs highly challenging. Conventional directed

testing is insufficient to cover the wide range of possible transaction scenarios, and hence, a more structured and reusable methodology is required. The five channels and handshake mechanism are depicted in Fig. 2.

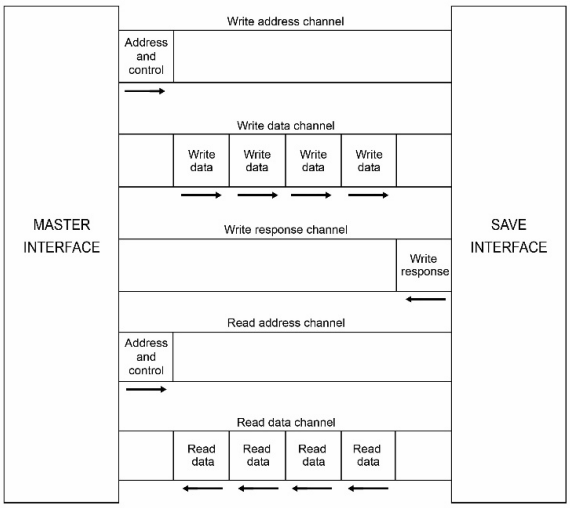

Fig.2. AXI Handshaking Channels

The AXI protocol defines five independent channels: write address (AW), write data (W), write response (B), read address (AR), and read data (R). Each channel uses a VALID/READY handshake mechanism to synchronize data transfer between master and slave. This paper presents design & implementation of reusable AXI Master and Slave Universal Verification Components (UVCs) using UVM. The proposed environment supports a comprehensive set of AXI4 features, including aligned and unaligned transfers, burst types (FIXED, INCR, WRAP), narrow transfers, and out-of-order transactions. The verification environment was validated using directed and constrained-random tests, and simulation results demonstrate improved functional coverage and reusability. The work contributes towards creating a scalable and modular AXI verification framework suitable for complex SoC designs.

II. LITERATURE SURVEY

Several researchers explored the verification of the AMBA-AXI4 protocol using UVM and related methodologies. Most existing works focus on coverage-driven and assertion-based environments, demonstrating UVM's capability in handling complex bus protocols. These studies highlight the efficiency of constrained-random stimulus generation and functional coverage metrics in detecting protocol violations. Karuna et al.[1] presented a coverage-driven and assertion-based AXI4 verification environment using UVM, focusing on protocol compliance and systematic detection of errors. Their work demonstrated the effectiveness of combining assertions with constrained-random stimulus, but it lacked modular Master and Slave components for broader reusability. Harish et al.[3] and Arun and Suganya[4] investigated constrained random verification of AXI4 burst operations, validating aligned and unaligned transfers as well as corner cases such as write-first and read-first transactions. However, their focus remained limited to specific scenarios, and reported functional coverage(58–65%) was insufficient for comprehensive protocol compliance. Another work [5] explored assertion-based verification with emphasis on handshaking and transaction dependencies but lacked independent Master and Slave components and did not cover all AXI4 burst types. Vinay H. and Balaji B. S.[8] designed a reusable AXI Master and Slave Verification IP (VIP) using UVM, emphasizing reconfigurability, multi-master/multi-slave support, and burst operations. While their approach improved reusability, it retained a monolithic VIP structure rather than fully modular UVCs. According to the AMBA AXI4 specification [10], the protocol supports advanced features such as VALID/READY handshaking, independent channels, out-of-order completion, narrow transfers, and burst types (FIXED, INCR, WRAP). Nevertheless, most prior research did not comprehensively implement or verify the full feature set. These limitations motivate the development of a robust, modular UVM-based AXI4 verification environment. The proposed work addresses these gaps by implementing independent Master and Slave UVCs, supporting all AXI4 transaction types, and achieving higher functional coverage, thereby improving reusability, scalability, and completeness in AXI4 verification.

III. PROPOSED SYSTEM

To address the limitations in existing AXI4 verification studies, this work proposes a modular UVM-based verification environment featuring independent Master and Slave UVCs. The Master

UVC consists of a driver, monitor, and scoreboard with coverage, while the Slave UVC contains a responder and monitor. This modular structure ensures reusability, scalability, and separation of responsibilities, allowing easy extension to more complex SoC verification scenarios. The Master UVC drives AXI4 transactions using a sequence-driven approach, while the monitor observes the bus signals to update the scoreboard and functional coverage metrics. The Slave UVC responds to incoming transactions, ensuring protocol compliance, and its monitor checks for correct handshaking, data integrity, and burst behavior. Assertions are integrated in both UVCs to validate critical protocol features, such as VALID/READY handshaking, burst boundaries, and transaction ordering. The proposed environment supports all AXI4 transaction types, including FIXED, INCR, and WRAP bursts, as well as narrow transfers. The framework aims to achieve higher functional coverage compared to previous studies, addressing gaps in prior verification environments. Fig.3. illustrates the overall architecture, showing the interaction between Master & Slave UVCs, monitors, responder, and scoreboard.

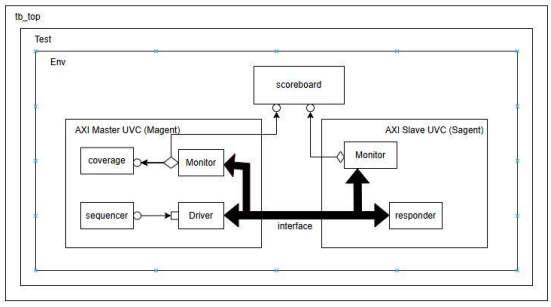

Fig.3. Proposed Architecture

This approach provides a flexible, robust, and extendable verification system, ensuring comprehensive AXI4 protocol verification suitable for both academic and industrial applications.

IV. IMPLEMENTATION

A. Master UVC

The Master agent consists of a sequencer, driver, monitor, and coverage collector. The sequencer generates AXI4 transactions, which the driver drives onto the interface connected to the Slave agent. The Master monitor observes these transactions and forwards the expected outputs to the environment scoreboard. Functional coverage within the Master agent tracks transaction types, burst lengths, addresses, and handshaking events, providing insight into which scenarios have been exercised by the test sequences.

B. Slave UVC

The Slave agent includes a responder and a monitor. The responder receives transactions from the Master interface and generates responses according to the AXI4 protocol. The Slave monitor observes the actual outputs from the responder and sends this information to the environment scoreboard. This setup ensures accurate tracking of transactions and synchronized verification between Master and Slave agents.

C. Scoreboard and Functional Coverage

The environment scoreboard collects expected outputs from Master monitor and actual outputs from Slave monitor, comparing them to report matches or mismatches. Functional coverage tracks transaction types, burst completions, address ranges, and WRAP boundary events across all testcases, providing a clear view of which AXI4 scenarios have been exercised and ensuring completeness of verification.

D. Wrap Burst Handling

Wrap bursts are handled by calculating the upper and lower address boundaries, ensuring correct alignment for transactions that wrap around the burst boundary. The formula for calculating wrap addresses is provided below.

$$\text{Total\_num\_bytes} = (\text{AxLEN} + 1) \times 2^{\text{AxSIZE}}$$
$$\text{Offset} = \text{AwADDR} \% \text{Total\_num\_bytes}$$
$$\text{Lower Wrap Boundary (LB)} = \text{AwADDR} - \text{Offset}$$
$$\text{Upper Warp Boundary (UB)} = \text{LB} + (\text{Total\_num\_bytes} - 1)$$

Here, LB represents the lower wrap boundary, which is the first address of the burst, and UB represents the upper wrap boundary, which is the last address of the burst. All addresses generated during a wrap burst must fall within this LB–UB range.

E. Verification Flow

During verification, the sequencer in the Master agent generates a transaction sequence, which is driven by Master driver through the interface to Slave responder. The Master monitor captures the expected behavior,

while the Slave monitor observes the actual behavior of the responder. Both monitors forward their transaction data to the scoreboard, which performs a comparison to determine correctness. Matches and mismatches are logged, and functional coverage is updated in real time, tracking exercised transaction types, burst behaviors, and wrap boundary occurrences. Directed and constrained-random tests cover aligned and unaligned transactions, FIXED, INCR, and WRAP bursts, narrow transfers, and multiple outstanding operations. Simulation results demonstrated functional coverage in the range of 83%, thereby improving completeness and reusability of the verification environment compared to earlier works.

V. RESULTS AND DISCUSSIONS

The AXI4 verification environment was implemented in SystemVerilog using UVM and simulated on QuestaSim. Different transaction scenarios were exercised to validate protocol compliance and measure verification completeness.

Fig.4 shows the VALID/READY handshake on write channel, confirming correct synchronization between the Master & Slave agents.

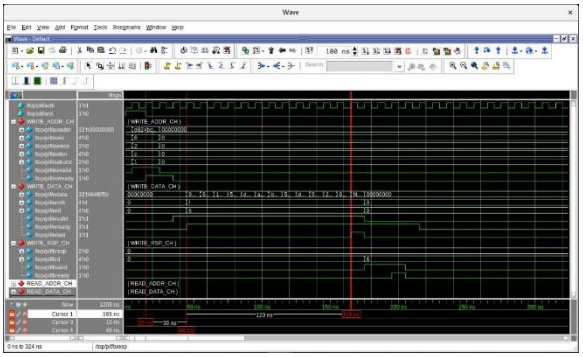

Fig.4. VALID/READY handshake on the write channel

Fig.5 illustrates a 5-write and 5-read INCR burst transaction, demonstrating proper sequencing of address, data, and response phases across AW, W, B, AR, and R channels. This confirms the correct handling of aligned incremental bursts. Fig.6 presents a WRAP burst transaction, where the address wraps around at the calculated boundary. This validates the correctness of the wrap address formula and boundary alignment.

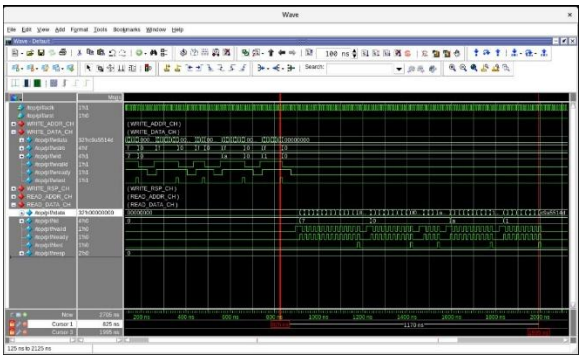

Fig.5. 5-write and 5-read INCR burst transaction

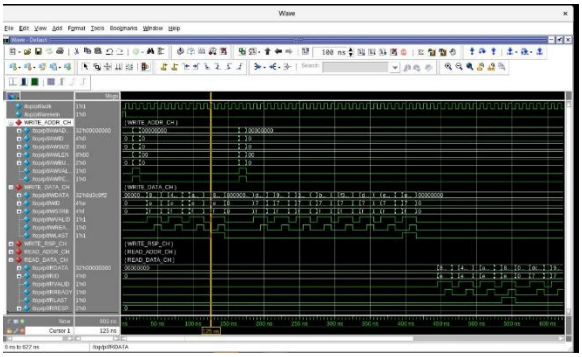

Fig.6. WRAP burst transaction showing boundary rollover

Fig.7 corresponds to the FIXED burst transaction, where repeated transfers occur to the same address location. The results confirm protocol compliance for constant-address bursts. Fig.8 highlights an unaligned transfer, showing that the environment correctly handles beginning addresses that are out of alignment to the bus width. Fig.9 demonstrates a narrow transfer, verifying support for transfer sizes smaller than the data bus width.

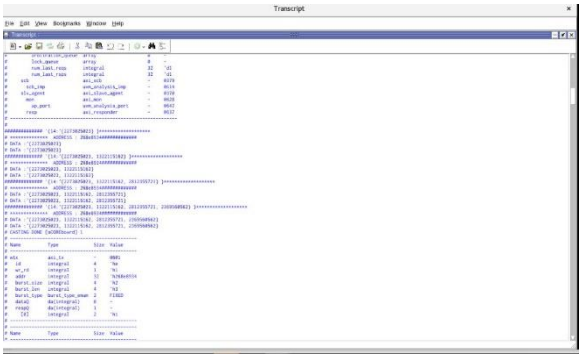

Fig.7. FIXED burst transaction transcript window

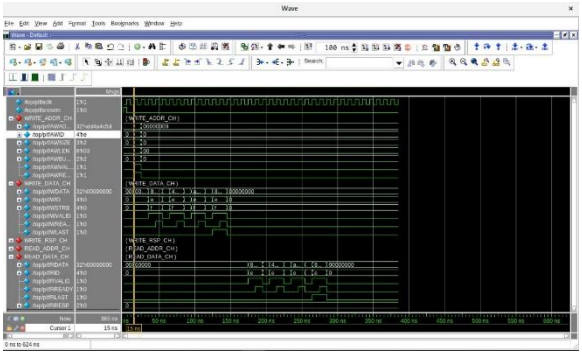

Fig.8. Unaligned transfer

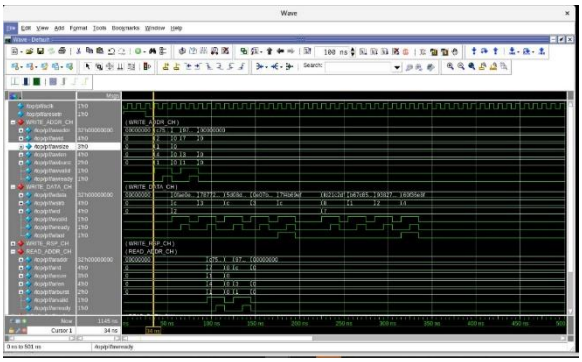

Fig.9. Narrow transfer with reduced transfer size

Fig.10 shows an out-of-order completion case, confirming that the environment is capable of handling responses that return in a different order from the requests, in accordance with AXI protocol rules.

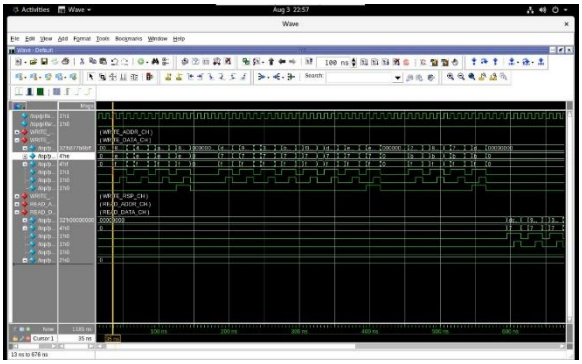

Fig.10. Out-of-order Transfer

Finally, the functional coverage report is shown in Fig.11, with an achieved coverage of approximately 83%. Coverage bins included transaction types (FIXED, INCR, WRAP), burst lengths, address alignments, narrow transfers, and out-of-order completions. This demonstrates that the verification environment exercised most critical features of the AXI4 protocol. Overall, the waveform and coverage results confirm that the proposed UVM-based verification environment is both reusable and scalable, providing comprehensive validation of AXI4 protocol features compared to earlier works.

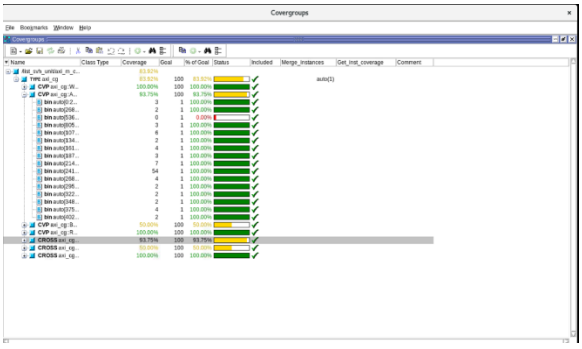

Fig.11. Functional coverage report

VI. CONCLUSION

This work presented a modular UVM-based verification environment for the AXI4 protocol, implemented in SystemVerilog and simulated using QuestaSim. Independent Master and Slave UVCs were integrated with a common scoreboard and coverage model to support all major AXI4 transaction types, including FIXED, INCR, WRAP, unaligned, narrow, and out-of-order cases. Simulation results confirmed correct VALID/READY handshaking and accurate comparison of expected and actual outputs, achieving a functional coverage of 83%. The proposed environment improves upon earlier works by providing better reusability, scalability, and completeness in protocol verification. Its modular design makes it simple to expand to more intricate SoC designs, while coverage metrics ensure systematic validation of protocol scenarios. Future work can focus on enhancing coverage by adding additional corner-case tests and extending the environment to multi-master configurations.

REFERENCES

[1] S. Karuna, P. Nishmika, Md. R. Jilani, M. M. Kowsik, M. C. Gowd, and P. Koteswararao, “Verification of Coverage-Driven and Assertion Based AXI4 Protocol VIP Using UVM,” Journal of Engineering Sciences, vol. 16, no. 4, 2025.

[2] B. Thakkar and V. Jayashree, "Verification of Driver Logic Using AMBA-AXI UVM," International Journal of VLSI Design \& Communication Systems (VLSICS), vol. 9, no. 3, Jun. 2018.

[3] H. T. L. Harish, M. C. Chandrashekhar, and M. N. Eshwarappa, "System Verilog Verification Techniques for AXI4 Slave Devices: A Practical Approach," Research Article, Dec. 2023.

[4] C. G. Arun and S. Suganya, "Constrained Random Verification of Burst Operation in AXI-4," 2022 IEEE International Interdisciplinary Humanitarian Conference for Sustainability (IIHC-2022), Nov. 18–19, 2022.

[5] V. Melikyan, A. Kirakosyan, S. Harutyunyan, and T. Kaplanyan, "UVM Verification IP for AXI," 2021 IEEE East-West Design & Test Symposium (EWDTS), 2021.

[6] N. Gaikwad and V. N. Patil, "Verification of AMBA AXI On-Chip Communication Protocol," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018.

[7] K. A. Sangamnerkar, Y. D. Kapse, and N. R. Kolhare, "Scalable Design and Implementation IP for Advanced Extensible Interface (AXI) Protocol," 2023 IEEE Fifth International Conference on Advances in Electronics, Computers and Communications, 2023, doi:10.1109/ICAECC59324.2023.10560234.

[8] V. H and B. B. S, "Design of Reusable VIP(Verification IP) of AXI Master and Slave in UVM," 2024 International Conference on Computing, Semiconductor, Mechatronics, Intelligent Systems and Communications (COSMIC), Mangalore, India, 2024, pp. 277-280,doi:10.1109/COSMIC63293.2024.1087145.

[9] R. Bhaktavatchalu, B. S. Rekha, G. Ananta Divya, and V. Usha Sai Jyothi, "Design of AXI bus interface modules on FPGA," in 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2016, pp. 141-146.

[10] ARM Limited, "AMBA® AXI Protocol Specification," ARM IHI 0022, Issue K, Sep. 2023.

[11] G. M., R. Sebastian, S. R. Mary, and A. Thomas, "A SV-UVM framework for Verification of SGMII IP Core with reusable AXI to WB Bridge UVC," in 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, Jan. 2016.

[12] M. Chen, Z. Zhang, and H. Ren, "Design and Verification of High Performance Memory Interface Based on AXI Bus," in 2021 IEEE 21st International Conference on Communication Technology (ICCT), China, 2021.

[13] V. K. Krishna Mangu, V. Vineeth, and S. Rajagopalan, "Robust Modular UVM Framework for Peripheral Verification," in 2024 India Council International Conference, India, 2024.

[14] M. Salman and R. Bhaktavatchalu, "HIGH PERFORMANCE AND LOW POWER AXI-LITE4 MASTER CONTROLLER USING SYSTEMVERILOG," in 2024 2nd International Conference on Artificial Intelligence and Machine Learning (AIMAL), 2024.

[15] P. Guoteng, L. Li, O. Guodong, F. Qingchao and B. Han, "Design and Verification of a MAC Controller Based on AXI Bus," 2013 Third International Conference on Intelligent System Design and Engineering Applications, Hong Kong, China, 2013, pp. 558-562, doi: 10.1109/ISDEA.2012.136

[16] M. P. Deepu and R. Dhanabal, "Validation of transactions in AXI protocol using system verilog," 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), Vellore, India, 2017, pp. 1-4, doi: 10.1109/ICMDCS.2017.8211605.

[17] N. Pothana and V. H. P. Reddy, "Verification of Coverage-Driven and Assertion based AXI4 Protocol VIP using UVM," 2023 First International Conference on Cyber Physical Systems, Power Electronics and Electric Vehicles (ICPEEV), Hyderabad, India, 2023, pp. 1-7,doi: 10.1109/ICPEEV58650.2023.10391880.