



## AUTOMATED RTL CODE GENERATION FOR PARAMETERIZED AMBA AXI4 INTERCONNECT

Shaik Sumiya Begum<sup>1</sup>, Gannera Mamatha<sup>2</sup>

<sup>1</sup>Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University College of Engineering, Anantapur, Andhra Pradesh, India.

[sumiyabegum686@gmail.com](mailto:sumiyabegum686@gmail.com)

<sup>2</sup>Associate Professor, Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University College of Engineering, Anantapur, Andhra Pradesh, India.

[gmamatha.ece@jntua.ac.in](mailto:gmamatha.ece@jntua.ac.in)

**Abstract—** Flexible on-chip interconnects are necessary due to increasing complexity of SoC as they accommodate different SoC topologies. The AMBA AXI4 protocol is prevalent because of its features including separate read and write channels, burst transactions, simultaneous transactions, and provision for multiple master-slaves. Unfortunately, there are significant changes in RTL required with changes in configuration parameters including number of master/slaves, address size, and data size among others. In this paper, an automation method and scalable framework is developed for generating RTL for configurable AMBA AXI4 interconnects. The architecture parameters can be configured through a GUI created using python. Thereafter, the Verilog RTL will be automatically generated based on these parameters including AXI channels, arbitration, address decoding, routing, and VALID/READY handshaking. The tool is evaluated using configurations of 1M-2S, 2M-4S, 2M-8S, and 4M-8S in EDA playground and EP Wave.

**Keywords—** AMBA AXI4, automated RTL generation, AXI interconnect, Verilog, configurable RTL, System-on-Chip (SoC), EDA Playground, UVM, DMA.

### I. INTRODUCTION

Modern SoCs contain processors, DMA controllers, memory blocks, peripherals, and accelerators that need an efficient interconnect for communication. This is why AMBA AXI4 protocol has been very popular as an interconnect for on-chip

communication due to the availability of independent read/write channels, burst transactions, transaction IDs, and VALID/READY flow control [1], [2].

The complexity of AXI4 interconnect increases significantly if there are many masters and slaves involved in the design. Proper arbitration, address decoding, multiplexing, demultiplexing, transaction and response handling are required. Any change in system parameters may need significant RTL changes and re-verification. The authors have previously shown in their conference paper an automatic tool using Python/Tkinter framework for AXI RTL generation and directed/UVM-based verification. In this journal work, the authors extend the above work by investigating asymmetric master-slave topologies, i.e., the number of masters and slaves are not the same.

### II. RELATED WORK

Earlier works have looked into high performance SoC interconnects, AXI arbitration, synthesizable AXI, DMA architectures, and UVM-based verification. Arbitration, buffering, and topology related issues for high performance interconnects were examined by Lyubavin and Telpukhov [1]. AXI4 design and verification problems were considered in [2] by Saini, whereas synthesizable AXI implementations and the need for address decoding were studied by Reddy et al. [3].

In addition to this, multi-master and multi-slave AXI architectures [6] as well as real-time AXI interconnects for embedded

systems [7] have been analyzed. UVM-based approach allows creating reusable drivers, monitors, sequencers, and scoreboards for AXI verification [5], [9], [10]. DMA-based AXI architectures have been looked into [8] for fast data transfers.

Even though the above-mentioned papers prove the importance of AXI interconnect design and verification, topology change of the manually designed RTL interconnect still requires significant modification. This problem was handled before using automation of RTL generation.

The current work continues the above research by adding experimental results for non-square master-slave configurations and another UVM verification waveform.

### III. AMBA AXI4 AND PROPOSED ARCHITECTURE

#### A. AXI4 Architecture

AXI4 consists of five channels that work independently, which are Write Address (AW), Write Data (W), Write Response (B), Read Address (AR), and Read Data (R). In all channels, there is the VALID/READY protocol to transfer data whenever both are asserted.

In AXI4, there is burst transfer support, transaction IDs, and multiple outstanding transactions. The suggested generator can implement FIXED, INCR, and WRAP burst modes.

In a multi-master/multi-slave setup, the general communication channel is:

Masters → Arbitration → Address Decoding → Routing → Slaves

Read data and write responses move in the opposite direction.

### IV. PROPOSED AUTOMATED RTL GENERATION FRAMEWORK

The proposed design framework includes the following blocks: Python GUI, RTL generation core, generated Verilog interconnect, and HDL simulation setup.

Design flow is the following:

User parameters → Python GUI → RTL generator → Verilog RTL → EDA Playground → EPWave/ UVM verification GUI supports selection of the following parameter values:

| Parameter Values |                   |
|------------------|-------------------|
| Data width       | 8, 16, 32 bits    |
| Address width    | 8, 16 bits        |
| Bus type         | AXI FULL          |
| Burst type       | FIXED, INCR, WRAP |
| Masters          | 1, 2, 4, 8        |
| Slaves           | 1, 2, 4, 8        |

Generator uses selected values for substitution in parameterized Verilog template. Generated RTL code includes all AXI AW, W, B, AR and R channels as well as arbitration, address decoding and routing functionality. The idea of parameterized design was used in the previous implementation.

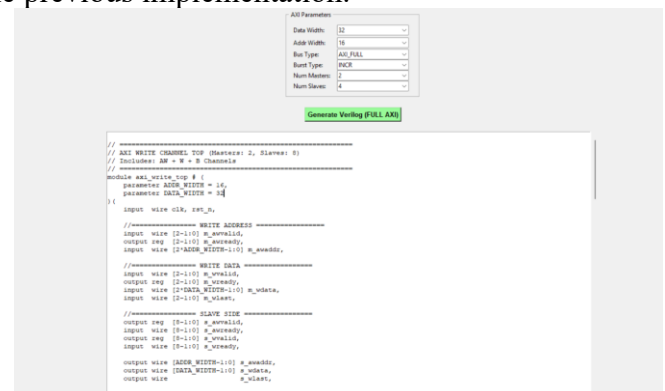


Fig.1. Python based GUI for AXI4 RTL Generation

### V. VERIFICATION METHODOLOGY

Verification of the generated configuration was done via directed RTL simulation and UVM-based verification.

In the first approach, generated configurations were executed using EDA Playground and analyzed using EPWave. The configurations under verification are listed in Table I.

TABLE I

### EVALUATED AXI4 CONFIGURATIONS

| Configuration | Masters | Slaves | Verification |
|---------------|---------|--------|--------------|
| 1M-2S         | 1       | 2      | Directed     |
| 2M-4S         | 2       | 4      | Directed     |
| 2M-8S         | 2       | 8      | Directed     |
| 4M-8S         | 4       | 8      | Directed     |
| DMA/FIFO/RAM  | 1       | 2      | UVM          |

Waveform was analyzed in terms of clock/reset behavior, address transfer, transaction ID, write/read data, VALID/READY handshake, slave selection and response activity.

The result confirms the capability of the generator to generate topology with different number of masters and slaves.

#### B. 2M-4S Configuration

### VI. RESULTS AND DISCUSSION

#### A. 1M-2S Configuration

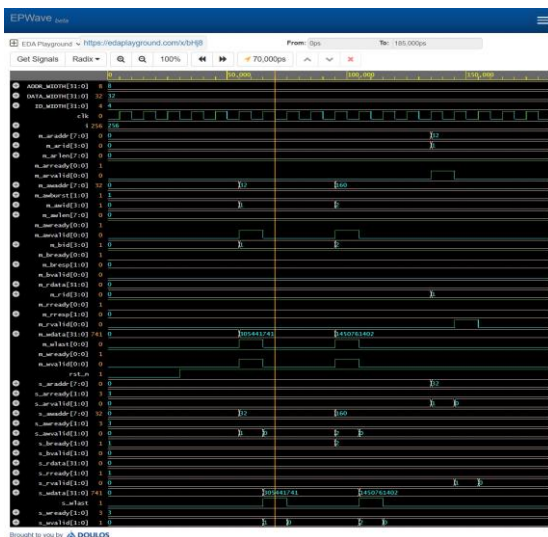


Fig. 2. Output waveform for 1-master/2-slave AXI4 interconnect.

The waveform of 1M-2S configuration represents the asymmetric interconnect with 1 master interface and 2 slave interfaces. All address, write data, transaction control and slave side signals are active during the simulation. The VALID/READY signals represent the transfer process between master, interconnect and selected slave.

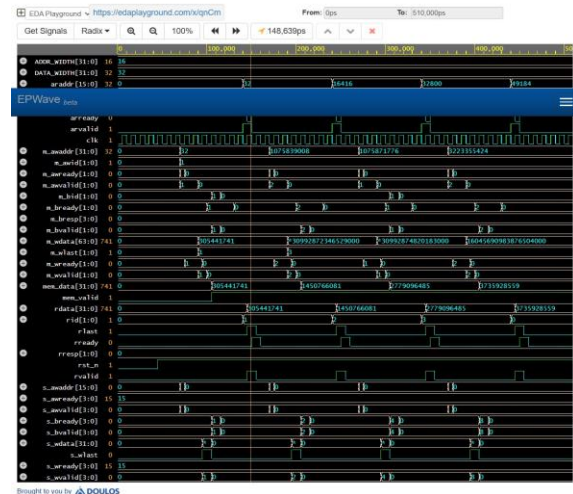


Fig. 3. Output waveform for 2-master/4-slave AXI4 interconnect.

The 2M-4S waveform represents multiple master interfaces communicating with four slave interfaces. The transaction addresses and IDs changes are accompanied by the activity on master and slave side. The VALID/READY signals represent the transfer phases across the generated interconnect.

The result shows that the generator can accept additional master and slave interfaces

with no modification in the generation methodology.

### C. 2M-8S Configuration

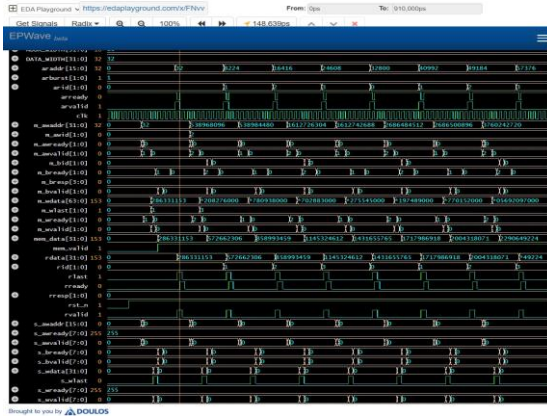


FIG. 4. OUTPUT WAVEFORM FOR 2-MASTER/8-SLAVE AXI4 INTERCONNECT.

The waveform of 2M-8S represents the extension of slave dimension to 8 interfaces but keeps the same number of master interfaces. Multiple transaction addresses, IDs, data values and handshake signals are observed during the simulation.

The result shows that the number of slave interfaces can be increased regardless of the number of masters.

### D. 4M-8S Configuration

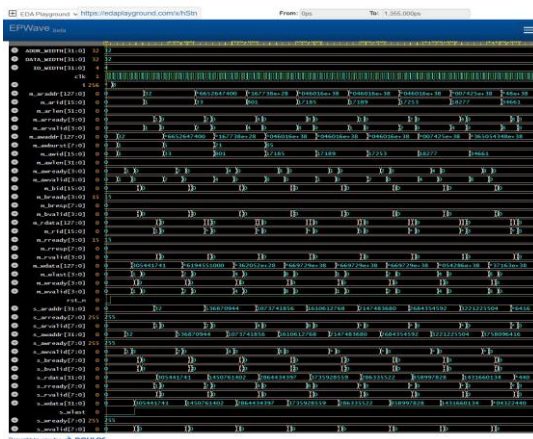


Fig. 5. Output waveform for 4-master/8-slave AXI4 interconnect.

The 4M-8S configuration represents the largest asymmetric topology considered in this work. It consists of 4 master interfaces and interconnect with 8 slave interfaces.

The waveform represents changes in transaction IDs, addresses, write data and VALID/READY activities. The result confirms that the generated architecture is able to support the large number of initiators and target interfaces using the same parameter-driven generation methodology.

### E. UVM Verification

The UVM verification environment consists of a DMA master and FIFO/RAM slave interface. The verification block contains the test, environment, agent, sequencer, driver, monitor and scoreboard.

In this methodology, the driver drives transactions into the DUT, the monitor monitors the activity on the bus and the scoreboard compares the behavior with the expected transaction results.

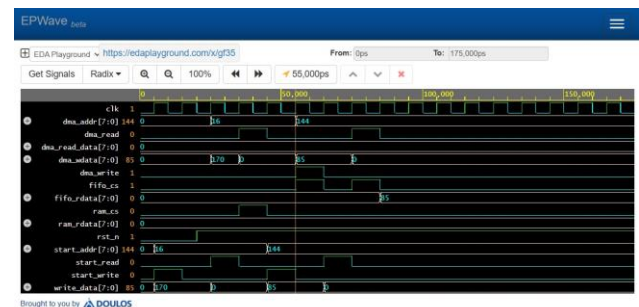


Fig. 6. UVM verification waveform for DMA master with FIFO and RAM slave interfaces.

The waveform includes signals dma\_addr, dma\_read, dma\_write, dma\_wdata, fifo\_cs, fifo\_rdata, ram\_cs and ram\_rdata. Observed chip-select activities of FIFO and RAM

slave interfaces represent the routing of the DMA transaction.

#### VII. SCALABILITY DISCUSSION

From the experimental results, it can be noticed three types of scalability. First, the number of masters can be increased from 1 to 4 with the same basic arbitration and routing methodology. Second, the number of slaves can be increased independently as seen in 2M-4S and 2M-8S configurations. Third, the results confirm the support for asymmetric topologies, instead of limiting the architecture to square configurations.

The progression of the experiments

1M - 2S → 2M - 4S → 2M - 8S → 4M - 8S, shows that the same generation framework can be used for different master-slave ratio configurations.

The UVM experiment provides an additional verification of the generated interconnect through DMA transactions and FIFO/RAM interfaces. In this way, the directed and UVM simulation provides functional proof of the generated AXI communication paths.

#### VIII. CONCLUSION

Automated and scalable framework for generating configurable AMBA AXI4 interconnect RTL has been presented. The Python based GUI allows the designer to select master number, slave number, address width, data width and burst type and automatically generates the Verilog RTL.

The journal extension evaluates four asymmetric configurations: 1M-2S, 2M-4S, 2M-8S and 4M-8S. EDA Playground and EPWave simulation provide an output with address, data, transaction ID and VALID/READY activities across the generated interfaces. The additional UVM based verification of DMA/FIFO/RAM configuration confirms transaction routing through the generated interconnect.

The results confirm that the proposed parameter-driven method can be reused for

different interconnect topologies without manual redesign of the whole RTL architecture. Future work will include latency and throughput measurements, QoS-based arbitration, coverage analysis and synthesis evaluation.

#### ACKNOWLEDGMENT

The authors sincerely thank the Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University College of Engineering, Anantapur, for providing the technical and academic support for this work.

#### REFERENCES

- [1] K. D. Lyubavin and D. V. Telpukhov, "Methods for Constructing High-Performance Interconnects in System-on-Chip Designs," 2024 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), 2024, pp. 14–18.
- [2] L. M. Saini, "Design, Analysis and Verification of AMBA AXI4.0," 2025 International Conference on Visual Analytics and Data Visualization, 2025, pp. 131–136.
- [3] M. S. Reddy, B. B. Rajesh, and T. G. Prasad, "A Synthesizable Design of AMBA-AXI Protocol for SoC Integration," International Journal of Engineering Inventions, vol. 1, no. 3, pp. 19–26.
- [4] P. Angadi, S. V. Shenvi, and S. Lokesh, "Design and Simulation of AXI4 Stream Interconnect Using Verilog," 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems, 2024, pp. 1–4.
- [5] N. Pothana and V. 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, 2023, pp. 1–7.



[6] M. K. Yadav, R. K. Paliwal, and K. C. Mahajan, "AXI Interface with a Multiple Master and Slave through Interconnect," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 7, no. 10, 2018.

[7] Z. Jiang, N. Audsley, D. Shill, K. Yang, N. Fisher, and Z. Dong, "AXI-Interconnect RT: Towards a Real-Time AXI-Interconnect for System-on-Chips," *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium*, 2021, pp. 437–440.

[8] M. Shruthi, M. S. Thejashwini, S. Vathsalya, and G. Amulya, "Design of AXI4-Based DMA Controller for High-Speed Data Transfers," *International Journal of Engineering Development and Research*, vol. 13, no. 4, pp. 495–501, 2025.

[9] T. L. Harish, M. C. Chandrashekhar, and M. N. Eshwarappa, "SystemVerilog Verification Techniques for AXI4 Slave

Devices: A Practical Approach," *International Journal of Human Computations and Intelligence*, vol. 2, no. 6, pp. 268–274, 2023.

[10] R. Samanth and S. G. Nayak, "Design and SV Based Verification of AMBA AXI Protocol for SoC Integration," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 1465–1469, 2019.

[11] V. N. Brahmanandam and C. Monohar, "Design of Burst Based Transactions in AMBA-AXI Protocol for SoC Integration," *International Journal of Scientific & Engineering Research*, vol. 3, no. 7, 2012.

[12] S. S. Begum and G. Mamatha, "Configurable RTL Code Generation for AMBA AXI Bus Architecture with UVM-Based Verification," *2026 1st International Conference on VLSI Circuits and Systems (ICVCS)*, 2026, doi: 10.1109/ICVCS69586.2026.11682962.