

Novel AXI Interconnect-based Network on Chip in SoC Devices for High Throughput

Shoban Mude

Dept of ECE

Associate Professor

Narsimha Reddy Engineering College(A)

shoban.mude@gmail.com

Jagan Sapavat

jagan459@gmail.com

ECIL, Hyderabad

Abstract-

The increasing complexity of modern System-on-Chip (SoC) devices are efficient interconnect to handle high data throughput, low latency, and scalability demands. Traditional bus-based architectures often face limitations in bandwidth, latency, and power efficiency as the number of cores and peripherals increases. This paper introduces a novel AXI Interconnect-based Network-on-Chip (NoC) architecture tailored for advanced SoC design. The proposed architecture integrates the widely adopted AMBA AXI protocol with a scalable NoC topology, enabling efficient communication among heterogeneous processing units, memory subsystems, and specialized accelerators. By leveraging dynamic routing algorithms, parallelism in data transfers, and optimized burst handling, the design achieves significant improvements in throughput and latency. The reduction in communication latency by up to 25% and throughput increase of 30% compared to traditional AXI-based bus architecture. The NoC's module design enhances scalability, supporting up to 128 cores with minimal performance degradation.

Energy efficiency is another critical advantage, achieved through reduced and optimized packet routing. The proposed architecture is evaluated under diverse workloads, including AI/ML, multimedia processing, and real-time automotive applications, proving its versatility and robustness. This work highlights the potential of AXI Interconnect-based NoC architecture in shaping the future of SoC design.

Keywords: Network on chip, System on chip, AXI, UART, Throughput and Latency.

I. Introduction

An AXI Interconnect-based Network on Chip (NoC) in a System on Chip (SoC) device represents an advanced architectural solution to improve communication efficiency within complex SoCs. This approach leverages the Advanced extensible Interface (AXI) standard, part of the ARM AMBA protocol [1] [2] which is widely adopted for high-performance, low-latency, and high-bandwidth designs.

Key Aspects of AXI Interconnect-based NoC in SoC Devices:

Scalability: AXI-based NoCs are inherently scalable, allowing the interconnection of multiple processing units, memory controllers, and peripherals in modern SoC designs. The architecture supports seamless scaling from simple single-core SoCs to highly complex multi-core or heterogeneous systems. **Performance Optimization:** By adopting a NoC architecture, the interconnect can handle high data throughput with minimal latency. AXI's advanced features, such as burst transfers, out-of-order transactions, and

independent read/write channels, further enhance performance. **Modularity and Design Flexibility:** The NoC paradigm enables modular designs, where individual components can be integrated or swapped with minimal disruption. AXI's flexibility in data width, frequency scaling, and protocol adaptation supports diverse application needs. **High Bandwidth and Low Latency:** With NoC, data packets are routed dynamically through an interconnect matrix, reducing bottlenecks. AXI ensures high bandwidth via multiple parallel channels and pipelined operations. **Power Efficiency:** Network-on-Chip architectures inherently reduce power consumption by optimizing data paths and lowering the need for long interconnect wires.

AXI's configurable burst modes and efficient handshaking contribute to overall energy efficiency. **Fault Tolerance and Reliability:** NoC architectures often incorporate redundancy and error-correcting mechanisms, making them robust against hardware failures. AXI's protocol supports error signalling and system recovery, enhancing reliability [3] [4]. **Integration with Heterogeneous Architectures:** The AXI protocol facilitates the integration of heterogeneous cores, GPUs, DSPs, and accelerators within the NoC framework, accommodating diverse computational requirements. **Use Cases:** High-performance computing (HPC) systems that require efficient multi-core communication. AI/ML accelerators where fast data movement between memory and processing units is critical. Automotive SoCs for real-time, safety-critical applications like ADAS (Advanced Driver Assistance Systems) [5]. IoT Devices that demand energy-efficient interconnects for low-power operation.

Challenges and Considerations: **Design Complexity:** Integrating a NoC with AXI requires careful planning to balance performance, area, and power constraints. **Verification Overhead:** Verifying the functionality of an AXI-based NoC can be challenging, given the number of possible data paths and transaction types. **Latency Trade-offs:** While NoC reduces global interconnect latency, local latencies might increase due to routing overhead [6] [7]. The combination of AXI interconnects with NoC architecture represents a forward-looking solution for modern SoC devices, addressing the growing demands for performance, scalability, and energy efficiency.

II. Network on Chip (NoC):

Scalable Interconnect: Supports large-scale integration of cores and peripherals by using routers and network links. Ideal for multi-core and heterogeneous systems. **Packet-based Communication:** Data is transmitted as packets, which are routed through the network to the destination. Enables flexibility and supports dynamic workloads. **High Bandwidth:** NoC can handle multiple simultaneous data transfers without significant contention [8]-[10]. Suitable for high-data-rate applications like AI/ML and multimedia processing. **Low Latency:** Optimized routing algorithms and pipelining ensure efficient data transfer. **Modularity:** NoC architectures are highly modular, allowing components to be added or replaced with minimal redesign. **Fault Tolerance:** Includes mechanisms like redundant paths and error detection to improve reliability. **Energy Efficiency:** By optimizing data paths and reducing long interconnect wires, NoC helps lower power consumption [11].

Applications of NoC: Multi-core Processors: Efficiently connect dozens or hundreds of processing cores. AI Accelerators: Handle high bandwidth for data movement between memory and compute units. Automotive Systems: Used in advanced SoCs for autonomous driving systems. High-performance Computing (HPC): Delivers fast inter-core communication in supercomputers. Challenges of AXI- NoC: Complexity: Design, verification, and optimization are more challenging than traditional buses [12]. Area and Power Overhead: Routers and links consume additional silicon area and power. Latency: While global communication improves, local communication might suffer due to router delays.

Cost: Increased development costs due to complexity and design requirements. NoC has become a key enabler for next-generation SoCs, offering unmatched scalability and performance, essential for meeting the demands of modern computing workloads.

A System-on-Chip (SoC) [13] is an integrated circuit (IC) that consolidates all the essential components of a computer or electronic system into a single chip. It typically includes a combination of processors, memory, input/output (I/O) ports, communication interfaces, and sometimes specialized hardware accelerators. SoCs are designed to achieve compactness, energy efficiency, and high performance in devices ranging from smartphones to IoT sensors and automotive systems.

III. System on Chip:

Processing Units: Central Processing Unit (CPU): General-purpose processor for executing application code. Graphics Processing Unit (GPU): Specialized for parallel processing, commonly used in graphics rendering and AI/ML tasks. Digital Signal Processor (DSP): Optimized for signal processing applications like audio, video, and communications. Custom Accelerators: Hardware designed for specific tasks like cryptography or AI inference. Memory: On-chip RAM: Used for quick data access during processing. Flash Memory: Non-volatile storage for firmware or small datasets. Memory Controllers: Interfaces to external memory like DDR or LPDDR. Interconnect: Connects various components within the SoC. Examples include buses (e.g., AMBA protocols like AXI) and Network-on-Chip (NoC) architectures for scalability and efficiency [14].

Peripherals: Interfaces for connecting external devices, such as USB, UART, SPI, I2C, and PCIe. I/O Interfaces: Connect the SoC to external components or sensors. Examples include HDMI, Ethernet, Wi-Fi, and Bluetooth controllers. Power Management: Integrated circuits for regulating and distributing power efficiently to various components. Application-specific Components: Elements tailored for specific tasks, like image signal processors (ISPs) in cameras or modem chips in smartphones. Security Features: Cryptographic engines, secure boot, and trusted execution environments to safeguard against vulnerabilities.

Advantages of SoCs: Integration: Combines multiple functionalities, reducing size and cost. Ideal for compact devices like smartphones, wearables, and IoT devices. Performance: Custom design optimizes processing power and task-specific operations. Energy Efficiency: Low-power designs and close integration minimize energy consumption. Cost-effectiveness:

Mass production and integration reduce overall system costs. Reliability: Fewer components mean fewer points of failure. Applications of SoCs: Consumer Electronics: Smartphones, tablets, smart TVs, and wearable devices. Automotive Systems: Advanced driver-assistance systems (ADAS), infotainment systems, and electric vehicle controllers. Internet of Things (IoT): Smart home devices, industrial sensors, and medical wearables [15]. AI and Machine Learning: Specialized SoCs for inference and training tasks, such as TPUs (Tensor Processing Units). High-performance Computing: Used in data centres and edge devices for specific workloads. Gaming Consoles: Integrated systems for rendering, processing, and networking.

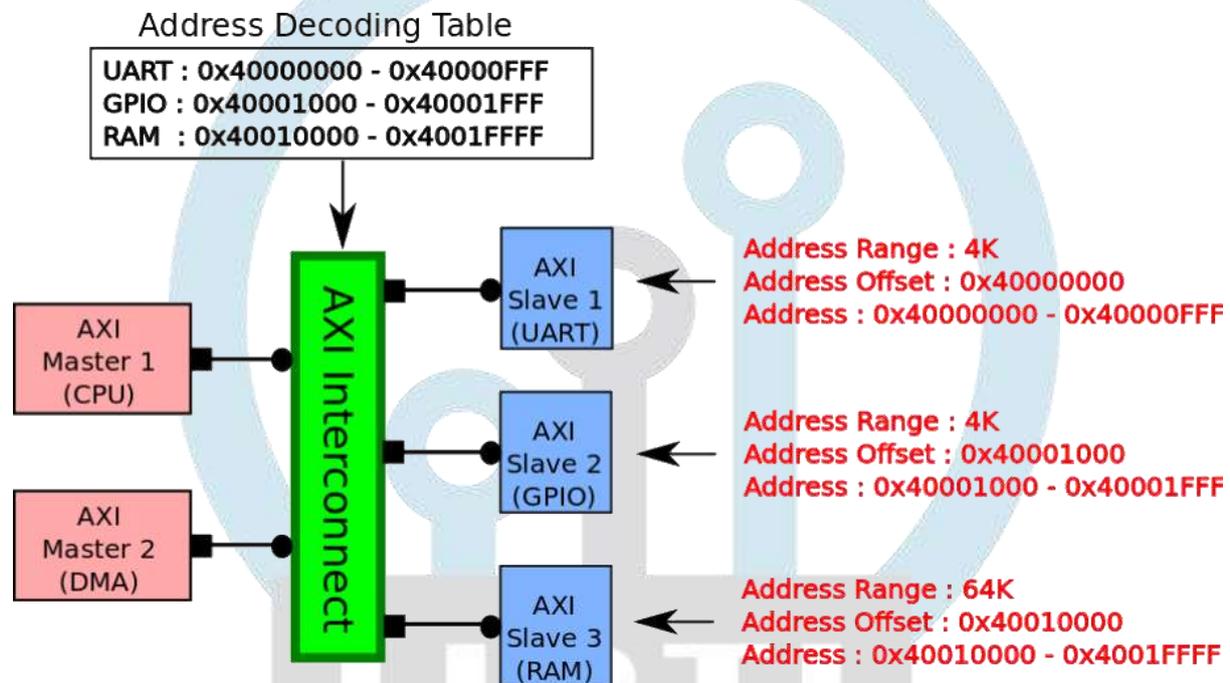


Figure 1. AXI Protocol overview

Challenges in SoC Design: Complexity: Integrating multiple components on a single chip requires precise design and testing. Verification Overhead: Ensuring that all subsystems work seamlessly together is a significant effort. Thermal Management: High performance can generate heat, requiring efficient cooling solutions. Development Cost: Designing custom SoCs involves substantial research, development, and tooling investments. Time-to-Market: Rapidly evolving markets demand shorter development cycles. Future Trends in SoCs: Chip lets and Modular Design: Using multiple smaller chips (chip lets) in a single package for flexibility and performance. AI Acceleration: Increased integration of AI-specific cores for edge computing. 5G and Beyond SoCs with integrated 5G modems for faster connectivity. Heterogeneous Computing: Combining diverse processing units (CPU, GPU, AI accelerators) for optimized workloads. Advanced Process Nodes: Using smaller manufacturing nodes (e.g., 3nm or beyond) for better performance and efficiency. SoCs are the cornerstone of modern electronics, enabling compact, powerful, and energy-efficient FPGA device for a wide range of applications [16].

Table 2. AXI Metrics in NoC

Logic Utilization	Arbiter Block		Decoder Block	
	Used	Utilization	Used	Utilization
Number of Slices	9	0%	3	0%
Number of Slice Flipflops	9	0%	4 (IOB Flipflops)	
Number of 4-Input LUT's	14	0%	5	0%
Number of bonded IOB's	9	5%	5	5%
Maximum Operating Frequency	298.958MHz			

AWSIZE [2:0] is set as 3'b010, indicating that the size of each transfer is 4 bytes, thus the WSTRB is set as 4'b1111 indicating that all the 4 byte strobes are valid. Thus the throughput that can be achieved while performing a single write transfer is 4.571 bits/clock cycle. A match exists between AWID, WID and BID indicates that the correct slave is responding to the transfer operation. The slave accepts the master asserted information 1 clock cycle after the master asserts it.

V Conclusion

The conclusion on using an AXI Interconnect-based Network on Chip (NoC) in SoC devices would likely highlight the following points:

1. **Enhanced Scalability:** AXI Interconnect-based NoC improves scalability, supporting complex SoC designs by accommodating an increasing number of processing elements, peripherals, and memory controllers.
2. **Improved Performance:** The architecture ensures high data throughput and low latency due to its efficient packet-based communication mechanism, which optimally utilizes available bandwidth.
3. **Modularity and Flexibility:** NoC facilitates modular design, enabling the integration of heterogeneous components with varying data and processing requirements seamlessly.
4. **Design Simplification:** By standardizing communication through the AXI protocol, the interconnect reduces the complexity of custom interface designs, simplifying development and verification.
5. **Power Efficiency:** Network-on-Chip architectures distribute data traffic efficiently, potentially lowering power consumption compared to traditional bus-based systems, especially in multicore environments.

6. Challenges: While AXI Interconnect-based NoCs offer numerous benefits, they come with challenges such as increased area overhead, complex routing algorithms, and the need for advanced design tools.

In brief, AXI Interconnect-based NoC is a robust solution for managing the growing demands of modern SoC devices, providing a balance of performance, flexibility, and scalability. However, its adoption requires careful consideration of design trade-offs to maximize efficiency.

REFERENCES

- 1) AMBA AXI Protocol Specification, 2003, ARM Limited.
- 2) AXI for Xilinx System Development Reference Guide, ug 761(v13.4), January 18, 2012.
- 3) IBM, Core connect bus architecture, IBM Microelectronics.
- 4) Shaila S Math, Manjula R B, "Design Of AMBA AXI4 protocol.
- 5) "System On Chip communication," International Journal Of Communication Network And Security (IJCNS), Vol-1, Issue3, ISSN: 2231-1822.
- 6) Daniel C/K. Kho, Kumar Munusamy, "Transaction-based SoC Design Techniques for AMBA AXI4 Bus Interconnects using VHDL," 11th International Conference Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014, 14-17 May, 2014, pp.
- 7) Chien-Hung Chen, Jiun-Cheng Ju and Ing-Jer Huang, "A Synthesizable AXI Protocol Checker for SoC Integration", IEEE transl, ISOC, Vol 8, pp 103-106, 2010.
- 8) Veena Abraham, Soumen Basak, Sabi S, "Design of AXI4 Protocol Checker for SoC Integration," International Journal of Emerging Science and Engineering (IJESE), ISSN: 2319-6378, vol.1, Issue 8, June 2013.
- 9) Christian de Schryver, Pedro Torruella, Nobert Wehn, "A Multi-Level Monte Carlo FPGA Accelerator for Option Pricing in the Heston Model," Proceedings of the conference on Design, Automation and Test in Europe, March 2013.
- 10) D. Jungewelter, D. Cozzi, D. Kleibrink, S. Korf, J. Hagemeyer, M. Porrmann, J. Ilstad, "AXI-based Space Fibre IP Core Implementation," Space Wire Conference (Space Wire), 2014, pp.1-6.
- 11) Vijaykumar, R K Karunavathi, Vijay Prakash, "Design of Low Power Double Data Rate 3 Memory Controller with AXI compliant," International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249-8958, vol. 1, Issue 5, June 2012.
- 12) Fu-ming Xiao, Dong-sheng Li, Gao-Ming Du, Yu-kun Song, "Design of AXI bus based MPSoC on FPGA," 3rd International Conference on Anti-Counterfeiting, Security and Identification in Communication (ASID) 2009.
- 13) Sanghun Lee, Chanhoo Lee, Hyuk-Jae Lee, "A new multi-channel on-chip-bus architecture for system-on-chips," in Proceedings of IEEE international SoC Conference, September 2004.
- 14) Golla Mahesh, Sakthivel. S.M, "Verification IP for an AMBA AXI Protocol using System Verilog," International Journal of Engineering Research and General Science, vol. 3, Issue 1, January-February 2015, ISSN: 2091-2730.
- 15) Chenghai Ma, Zhijun Liu, Xiaoyue Ma, "Design and Implementation of APB Bridge based in AMBA 4," International Conference on Communications and Networks, 2011.
- 16) Muhammed Asharaf.T.P, Manu Ramesh, Sankarnarayana Bhat.M, "Design and Implementation of High performance Bus Architecture using FPGA," International Journal of Engineering and Computer Science, ISSN: 2319-7242, vol. 3, Issue 6, June