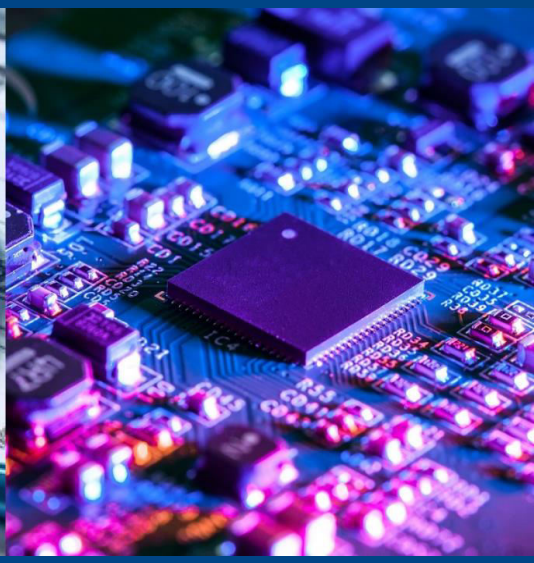
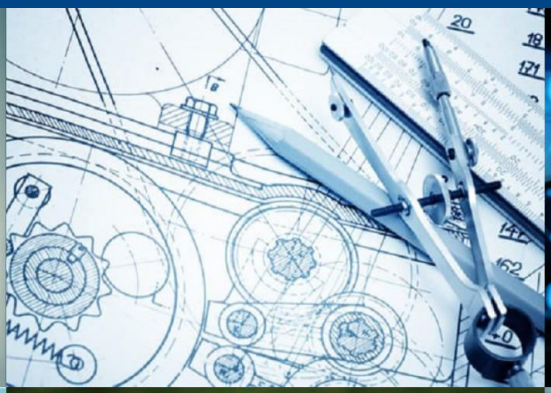
ISSN: 2582-7219

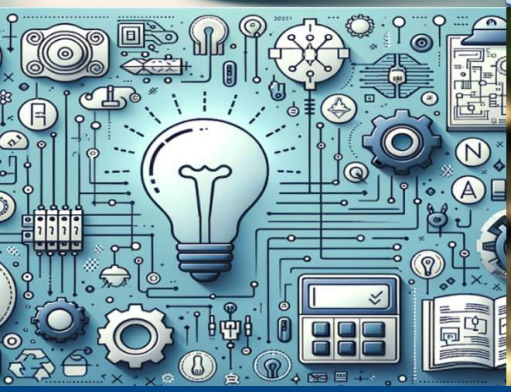




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 9, Issue 3, March 2026**

© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI: 10.15680/IJMRSET.2026.0903233

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Design and Implementation of AMBA AXI 4.0 Master for High Speed Performance SoC

D. N V Swathi[1], Y. Niharikasri[2], K.V Padmavathi[3], G. D N Sahithi[4], B. Sujatha[5]

U.G. Student, Department of ECE, SVIET Engineering College, Nandamuru, Pedana, Andhra Pradesh, India[1,2,3,4]

Assistant Professor, Department of ECE, SVIET Engineering College, Nandamuru, Pedana, Andhra Pradesh, India[5]

ABSTRACT: Modern System-on-Chip (SoC) designs require efficient and high-speed communication between different components such as processors, memory, and peripherals. Traditional bus architectures like AHB and APB suffer from limitations such as high latency and lack of parallel data transfer.

This project presents the design and implementation of an AXI-based master–slave communication system using Verilog HDL. The AXI protocol, developed under AMBA architecture, provides high-performance communication by using independent channels for address and data transfer. The system consists of an AXI master, AXI slave, and a top module connecting both.

The master initiates write and read operations, where data is written to a specific memory location and later retrieved successfully. A Finite State Machine (FSM) is used to control the sequence of operations, and a VALID–READY handshake mechanism ensures reliable data transfer. The system is simulated using EDA Playground, and results confirm correct functionality. This project demonstrates efficient and scalable communication suitable for modern SoC applications.

KEYWORDS: AXI Protocol, AMBA, FSM, Verilog HDL, SoC, Handshake Mechanism, Master–Slave

I. INTRODUCTION

The advancement of semiconductor technology has led to the development of System-on-Chip (SoC) designs, where multiple components such as processors, memory, and input/output peripherals are integrated onto a single chip. Efficient communication between these components plays a crucial role in determining overall system performance.

Traditional bus architectures such as Advanced High-performance Bus (AHB) and Advanced Peripheral Bus (APB) have been widely used in earlier systems. However, these architectures suffer from limitations such as higher latency, limited scalability, and lack of support for parallel data transfer. As system complexity increases, these limitations become more significant.

To address these challenges, the Advanced Microcontroller Bus Architecture (AMBA) introduced the Advanced eXtensible Interface (AXI) protocol. AXI is designed to provide high-speed, low-latency communication by using multiple independent channels and a flexible handshake mechanism.

This project focuses on designing and implementing an AXI-based master–slave communication system using Verilog HDL. The system demonstrates efficient data transfer, proper synchronization, and improved performance compared to traditional bus architectures.

.

II. RELATED WORK

In recent years, significant research has been conducted in the field of SoC communication systems. Early systems primarily used AHB and APB protocols, which were suitable for basic communication but lacked advanced features required for high-performance applications.

© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI: 10.15680/IJMRSET.2026.0903233

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

With the introduction of AXI protocol, researchers have focused on implementing high-speed communication systems using FPGA and embedded platforms. These systems utilize independent channels for address and data transfer, enabling parallel operations and improving throughput.

Several studies have explored multi-master AXI systems with complex arbitration mechanisms to support multiple devices communicating simultaneously. Other research works have focused on optimizing AXI performance by reducing latency and improving bandwidth.

Despite these advancements, simplified AXI implementations are essential for understanding the fundamental working of the protocol. This project provides a clear and structured implementation of AXI master–slave communication, making it suitable for educational and practical applications.

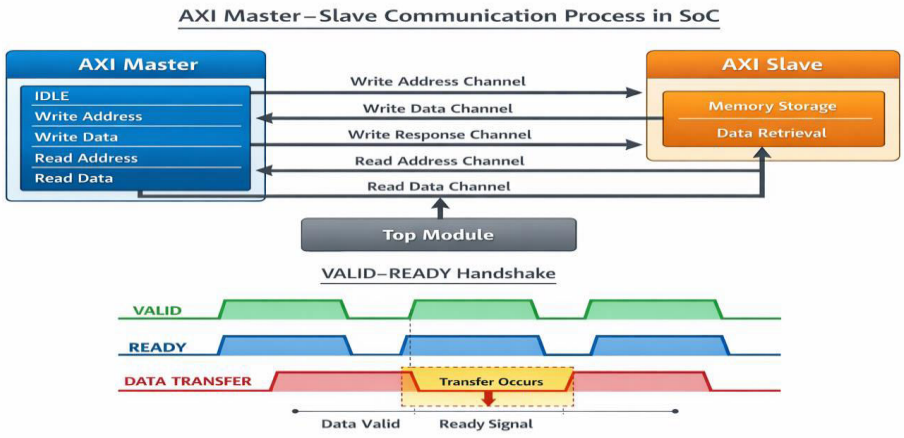
III. METHODOLOGY

The proposed system is designed to implement a high-speed communication interface based on the Advanced eXtensible Interface (AXI) protocol using Verilog HDL. The architecture follows a master–slave model in which the AXI master initiates all communication operations, while the AXI slave performs memory-related functions such as storing and retrieving data. A top module is used to integrate both components and ensure proper signal routing between them.

The AXI master is implemented using a Finite State Machine (FSM) to control the sequence of operations in a structured and synchronized manner. The FSM governs the transition between different operational states, beginning from an idle condition and proceeding through write and read processes. During the write phase, the master first transmits the address information through the write address channel, followed by the data through the write data channel. The slave receives this information and stores the data in a predefined memory location. After the completion of the write operation, the system generates a response signal to indicate successful data transfer.

Once the write process is completed, the master initiates the read operation by sending the required address through the read address channel. The slave accesses the stored data from memory and transmits it back to the master through the read data channel. This process verifies the correctness of the data transfer by ensuring that the data written to a specific address is accurately retrieved during the read operation.

The communication between the master and slave modules is carried out using five independent channels defined by the AXI protocol, namely write address, write data, write response, read address, and read data channels. These channels operate independently, allowing parallel execution of read and write operations, thereby improving system performance and reducing latency. Unlike traditional bus architectures, this separation of channels enables efficient utilization of system resources and enhances throughput.



© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI: 10.15680/IJMRSET.2026.0903233

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

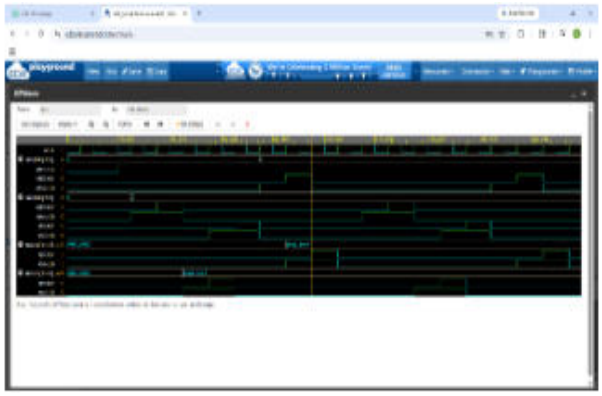
To ensure reliable data transfer, the system employs a VALID–READY handshake mechanism. The VALID signal is asserted by the transmitting module to indicate that valid data is available, while the READY signal is asserted by the receiving module to indicate its readiness to accept data. Data transfer occurs only when both signals are simultaneously asserted, ensuring proper synchronization between the master and slave. This mechanism eliminates the need for strict timing dependencies and prevents data loss, thereby improving communication reliability.

The entire design is implemented using Verilog HDL, which provides a flexible platform for modeling digital systems. A testbench is developed to generate the necessary clock and reset signals and to simulate the behavior of the system under different conditions. The simulation is performed using EDA Playground, where waveform outputs are analyzed to verify the correctness of the design. The results confirm that the system successfully performs both write and read operations, demonstrating the effectiveness of the AXI protocol in achieving high-speed and reliable communication.

IV. EXPERIMENTAL RESULTS

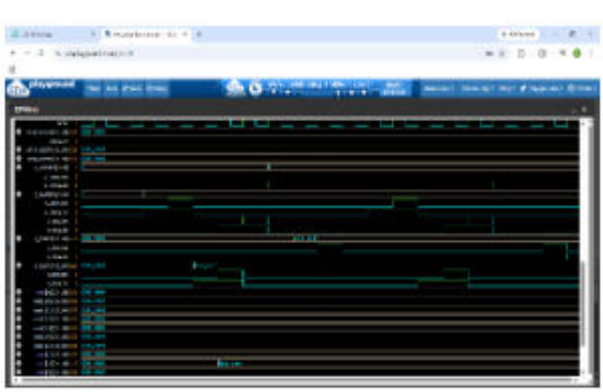
The proposed AXI-based system successfully performs write and read operations between the master and slave modules. The master writes data (0xDEADBEEF) to a specific address, which is stored in the slave memory and correctly retrieved during the read operation. The simulation waveforms show proper synchronization using the VALID–READY handshake mechanism. The results confirm that the system achieves reliable and efficient data transfer.

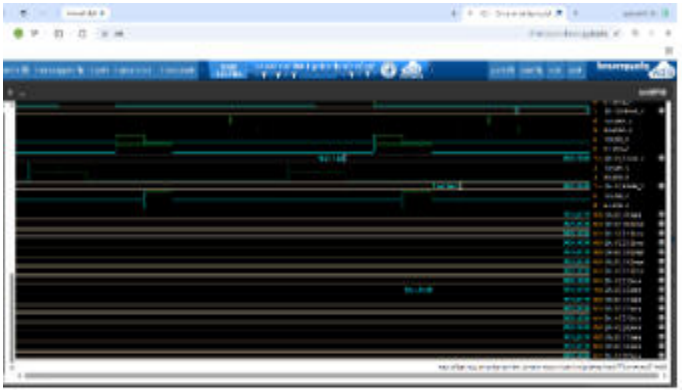
Fig 2:



(a) dut output

(b) master output





(c) slave output

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

(d) write output (e) read output

Fig 2 shows the results of design and implementation of AMBA AXI 4.0 master high speed soc (a) Dut output,(b) master Output,(c) Output of slave,(d) Output of write,(e) Output of Read.

V. CONCLUSION

The AXI-based master–slave communication system is successfully designed and implemented using Verilog HDL. The system demonstrates high-speed, reliable, and efficient data transfer using independent communication channels and a handshake mechanism.

The use of FSM ensures proper sequencing of operations, while simulation results confirm the correctness of the design. The project highlights the advantages of AXI protocol over traditional bus architectures, including improved performance and scalability.

Future work can focus on extending the system to support multiple masters and slaves, implementing burst transfer capabilities, and deploying the design on FPGA hardware for real-time applications.

REFERENCES

[1] XTECH_B_AXI4_Technical_Seminar, © Copyright 2012 Xilinx Confidential-Internal.
[2] IHI0022C_amba_axi_v2_0_protocol_spec, Copyright © 2003 2010 ARM, All rights reserved. ARM IHI 0022C (ID030510).
[3]Design of AMBA AXI4 protocol for System-on-Chip communication, International Journal of Communication Network and Security (IJCNS), Vol-1, Issue-3 ISSN: 2231-1882.
[4] Design and Implementation of AMBA based AXI 4 Slave Interface, IJIRSET, Vol. 4, Issue 6, June 2015..
[5]"AMBA AXI 4 and ACE Protocol Specification" 28 October 2011 D Non-Confidential First release of AMBA AXI 4 and ACE Protocol Specification.
[6] Xu Yang ; Harbin Inst. of Technol., Harbin ; Zhang Qing-li ; Fu Fang-fa ; Yu Ming-yan, "NISAR: An AXI compliant on-chip NI architecture offering transaction reordering processing" ASIC, 2007. ASICON '07. 7th International Conference, 22-25 Oct. 2007.
[7] Manjula, R.B. ; Manvi, S.S. ; Kaunds, P. "Data transactions on system-on-chip bus using AXI4 protocol" Recent Advancements in Electrical, Electronics and Control Engineering (ICONRAEeCE), 2011 International Conference, 15-17 Dec. 2011
[8] "Verilog-A Language Reference Manual Analog Extensions to Verilog HDL", Version 1.0, Open Verilog International August 1, 1996
[9] "Verilog-AMS Language Reference Manual", Release 2.3.1, Accellera Systems Initiative , 06-2009
[10] "Verilog-AMS Language Reference Manual". Release 2.4, Accellera Systems Initiative, 06-2014.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com