Efficient AHB-Lite to SPI Bridge

Vidyalankar Institute of Technology

Shitalkumar Pal - 22104A0066, Dhanashree Nar - 22104A0037, Atharva Dhagaonkar - 22104A0048, Sujal Chavan - 22104A0051

shitalkumar.pal@vit.edu.in, dhanashree.nar@vit.edu.in, atharva.dhagaonkar@vit.edu.in, sujal.chavan@vit.edu.in

Abstract

This work presents an efficient AHB-Lite to SPI bridge designed and implemented on the PYNQ-Z2 FPGA platform. The goal is to enable high-speed and reliable communication between an ARM-based processor and SPI peripherals such as TFT displays. The system features a custom SPI master, AHB-Lite protocol handling, FIFO-based buffering, and real-time software control through Jupyter Notebook. The design was validated using a Digital Storage Oscilloscope (DSO), confirming correct timing, functionality, and bus behavior.

Introduction

Modern embedded systems frequently require communication between a high-performance processor bus and low-speed peripheral devices. ARM-based SoCs commonly use the AHB-Lite bus architecture, whereas external peripherals—such as OLEDs, sensors, and memory devices—operate on SPI.

Commercial solutions do not always provide flexibility or insight into low-level timing. Therefore, a custom bridge offers advantages such as

- customized timing controls,
- simplified debugging,
- adaptable data flow, and
- hardware-level optimization.

This project implements a fully functional AHB-Lite to SPI bridge with FIFO buffering, DC/RESET control lines, and protocol compliance. The bridge uses AXI4-Lite for processor communication and converts it into AHB-Lite transactions internally.

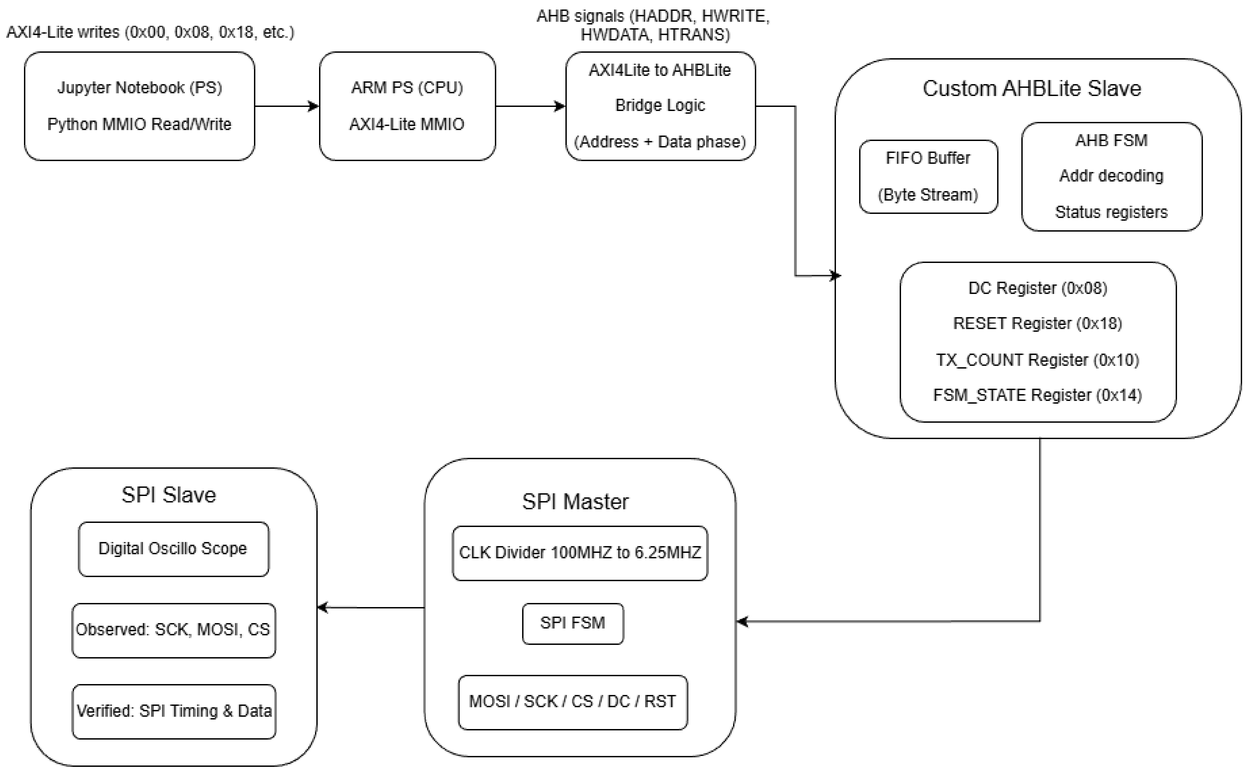
Literature Review

Several existing works and IP cores demonstrate AHB and SPI interfacing, but they often come with limitations:

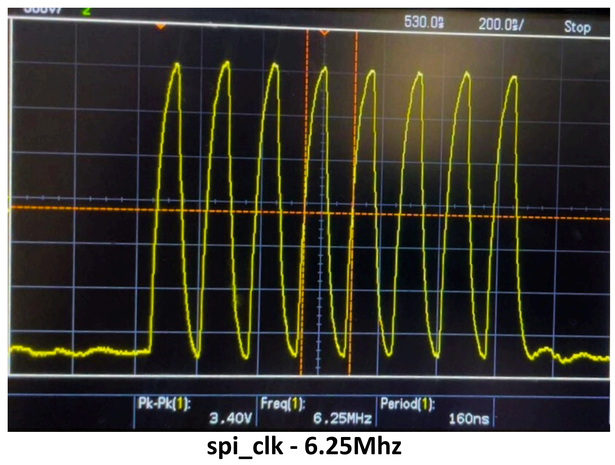
1. ARM AMBA Specifications (AHB-Lite & AXI4-Lite):
2. Defines high-performance pipelined bus communication but lacks direct interfacing to SPI peripherals.
3. Xilinx AXI Quad SPI IP:
4. A rich implementation but heavy for resource-constrained systems and not customizable at the protocol level.
5. OpenCores SPI Master:
6. Offers generic SPI support but lacks built-in AHB-Lite integration and efficient buffering.
7. Research Papers on Bus Bridges:
8. Most emphasize throughput optimization but not ease of debugging or FPGA-based educational implementations.

This project fills the gap by providing a lightweight, easily extensible, fully transparent AHB-Lite → SPI bridge with full access to low-level timing.

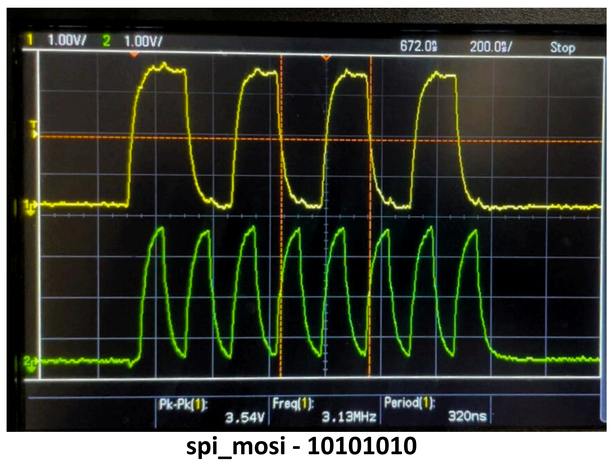
Methodology



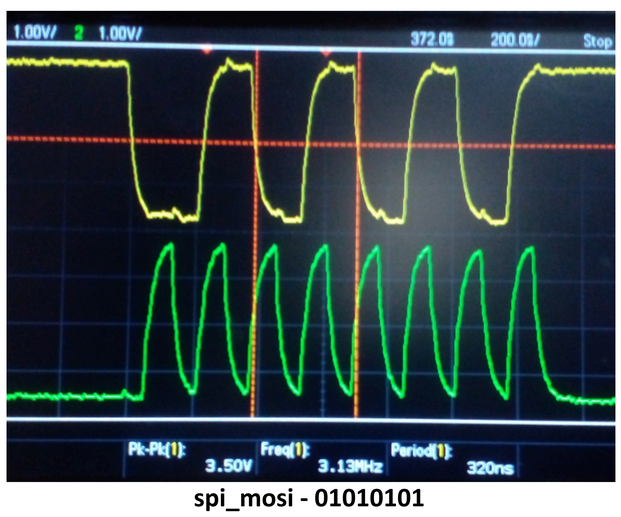
Results



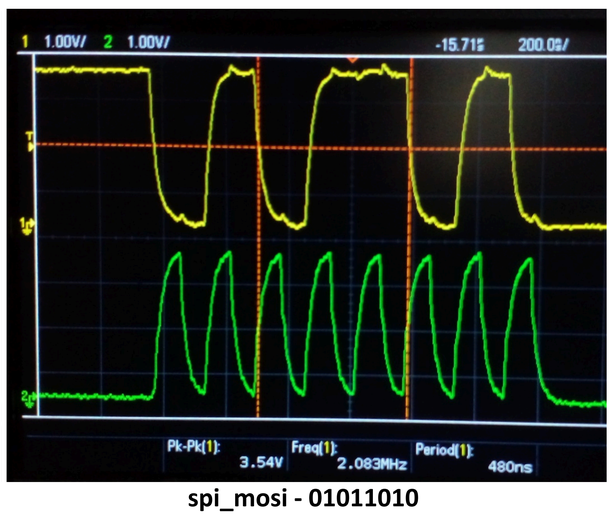
spi_clk - 6.25Mhz



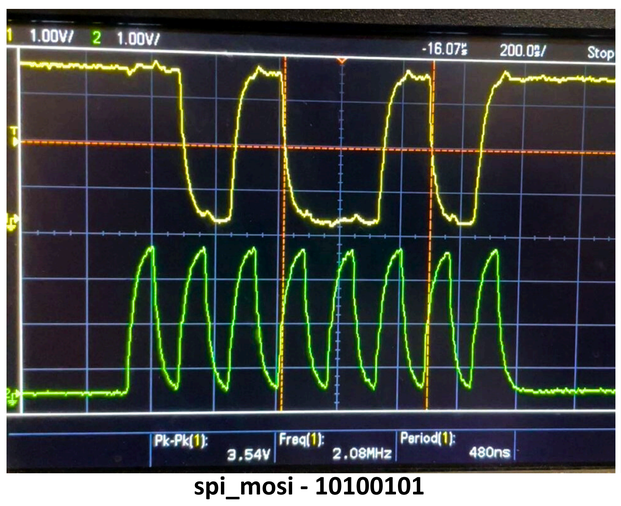
spi_mosi - 10101010



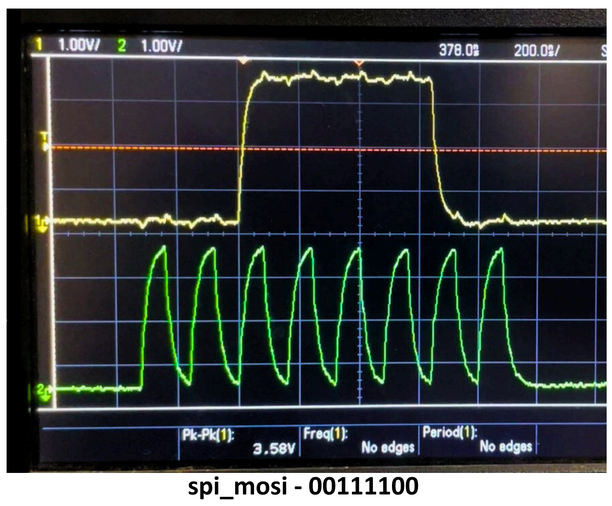
spi_mosi - 01010101



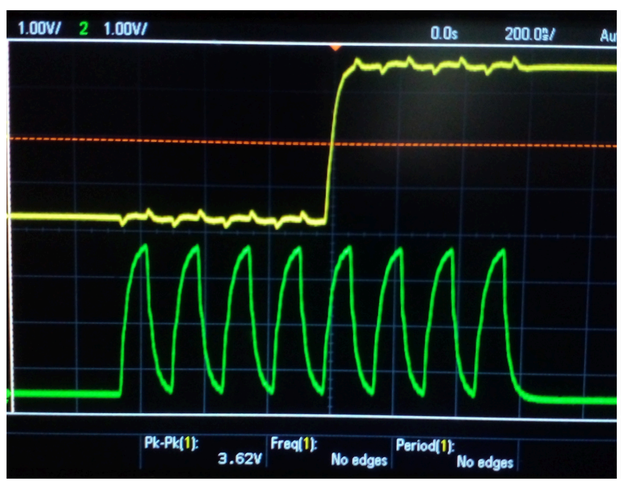
spi_mosi - 01011010



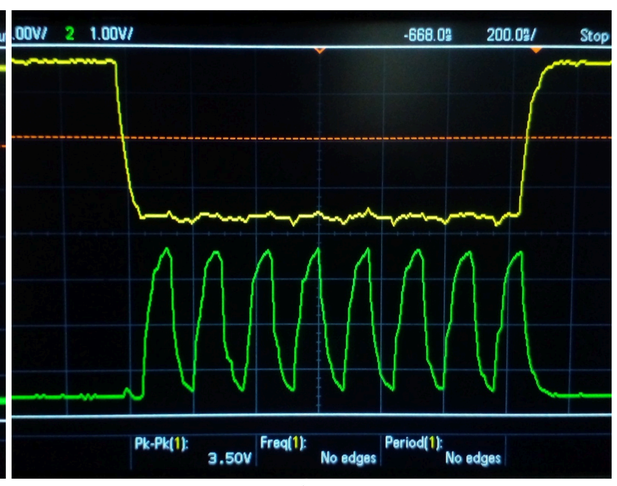
spi_mosi - 10100101



spi_mosi - 00111100



spi_mosi - 00001111



spi_cs

Conclusion

An efficient AHB-Lite to SPI bridge was successfully designed, implemented, and verified on the PYNQ-Z2. The bridge provides deterministic behavior, configurable timing, and easy integration for ARM-based FPGA systems. Real-time hardware verification confirmed correct SPI clock generation, data shifting, chip select behavior, and control signal logic. This work demonstrates the feasibility of custom bus bridges for educational, industrial, and research applications.

References

1. ARM AMBA 3 AHB-Lite Protocol Specification.
2. Xilinx AXI Reference Guide – UG761.
3. Xilinx SPI Interface Specifications – PG153.
4. OpenCores SPI Master Project Documentation.
5. PYNQ-Z2 Hardware Reference Manual.