DESIGN A MASTER-SLAVE SPI CORE COMMUNICATION WITH PHERIPHERAL CHIPS

K. SANTHOSHI, K. CHARAN TEJA, K. ANISH YADAV, K. VENKATA AKIL, K. ROHITH REDDY

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY AUTONOMOUS
DEPARTMENT OF ELECTORNICS AND COMMUNICATION ENGINEERING
(Affiliated to JNTUH, Accredited By NBA, Accredited by NAAC with "A+" grade)
2025-2026

ABSTRACT

Serial Peripheral Interface (SPI) is a widely used synchronous communication protocol for high-speed data transfer between a master device and one or more slave peripheral chips. This paper presents the design and implementation of a Master-Slave SPI core aimed at efficient communication with multiple peripheral devices such as sensors, EEPROMs, and ADCs. The proposed SPI core supports configurable clock polarity (CPOL), clock phase (CPHA), and flexible data widths, making it adaptable for diverse embedded applications.

The master SPI core generates the clock signal and controls communication using chip select lines, while the slave core responds to commands and exchanges data accordingly. The design is implemented using hardware description languages (HDL) such as Verilog/VHDL and can be synthesized on FPGA platforms. Emphasis is placed on optimizing timing, reducing latency, and ensuring reliable data transfer.

The system architecture includes shift registers, control logic, clock generation units, and buffering mechanisms. Peripheral chips are interfaced through dedicated chip select signals, allowing scalable expansion. The communication protocol ensures full-duplex transmission, enabling simultaneous sending and receiving of data.

Simulation results demonstrate correct functionality under different SPI modes and clock frequencies. The design achieves high throughput with minimal resource utilization. This SPI core can be used in applications like IoT systems, industrial automation, and embedded control systems where reliable and fast communication is required.

INTRODUCTION

In modern embedded systems, efficient communication between microcontrollers and peripheral devices is crucial. SPI (Serial Peripheral Interface) is one of the most commonly used protocols due to its simplicity, speed, and flexibility. Unlike asynchronous communication protocols, SPI operates synchronously, meaning data transfer is coordinated by a clock signal generated by the master device.

SPI follows a master-slave architecture where a single master communicates with multiple slave devices. The communication occurs through four main signals: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCLK (Serial Clock), and SS (Slave Select). The master controls the clock and selects the slave device using the SS line.

This paper focuses on designing a robust SPI core that supports master-slave communication with multiple peripheral chips. The need for such a design arises in applications like sensor networks, memory interfacing, and real-time data acquisition systems. A flexible SPI core allows designers to integrate multiple peripherals efficiently without complex wiring or protocols.

The proposed design includes configurable parameters such as clock frequency, data width, and SPI modes (Mode 0 to Mode 3). These features enhance compatibility with various peripheral devices. Additionally, the design ensures synchronization and minimizes errors during transmission.

The implementation is targeted for FPGA platforms, enabling rapid prototyping and testing. Simulation tools are used to validate functionality and performance. This work aims to provide a scalable and efficient SPI communication solution suitable for modern embedded systems.

SYSTEM ARCHITECTURE

The SPI communication system consists of a master core, one or more slave cores, and peripheral chips connected through a shared communication bus. The architecture is designed to support full-duplex communication, enabling simultaneous transmission and reception of data.

Main Components

- SPI Master Core
- SPI Slave Core
- Clock Generator

- Shift Registers
- Control Unit
- Peripheral Interface

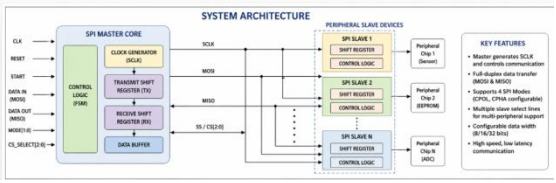
The master core generates the clock (SCLK) and controls data flow through MOSI and MISO lines. It also manages slave selection using multiple SS lines. Each slave device is connected via a dedicated SS signal, ensuring that only one device communicates at a time.

The slave core includes shift registers that store incoming and outgoing data. It operates based on the clock signal received from the master. The control unit manages data transfer, synchronization, and error handling.

Architecture Table

Component	Function
Master Core	Controls communication and clock generation
Slave Core	Responds to master and processes data
Shift Register	Serial-to-parallel data conversion
Clock Generator	Generates timing signals
Control Unit	Manages data flow and synchronization
Peripheral Chips	External devices (ADC, EEPROM, sensors)

The modular design allows easy integration of additional slave devices. The architecture ensures minimal delay and high data throughput. It is scalable and suitable for complex embedded applications.



DESIGN METHODOLOGY

The design of the SPI master-slave core is carried out using a structured methodology involving specification, modeling, simulation, and verification. The first step is defining system requirements, including data rate, clock frequency, and number of slave devices.

The SPI master is designed to generate clock signals and control communication. It includes a finite state machine (FSM) that manages data transfer operations such as idle, start, transmit, receive, and stop states. The slave module is designed to respond to the master's commands and handle data accordingly.

Shift registers are used for serial data transmission. Data is loaded into the register and shifted out bit by bit

synchronized with the clock. Similarly, incoming data is captured and stored.

The design supports all four SPI modes by configuring CPOL and CPHA values. This ensures compatibility with various peripheral devices.

Design Parameters Table

Parameter	Description	Value (Example)
Clock Frequency	SPI clock speed	1 MHz – 50 MHz
Data Width	Number of bits per transfer	8/16/32 bits
SPI Modes	CPOL & CPHA combinations	0,1,2,3
Slave Devices	Number of peripherals supported	Up to 8

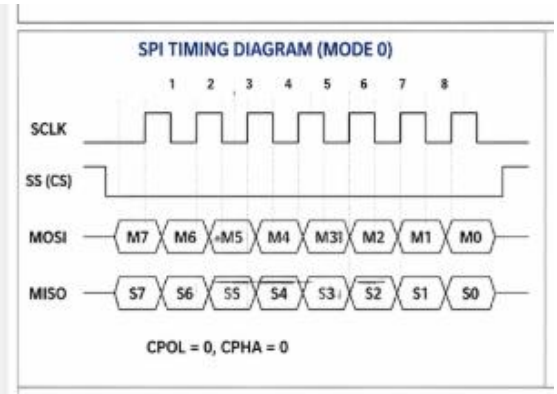
Simulation is performed using tools like ModelSim or Vivado. Test benches are developed to verify functionality under different scenarios. Timing analysis ensures that setup and hold requirements are met.

RESULTS AND DISCUSSION

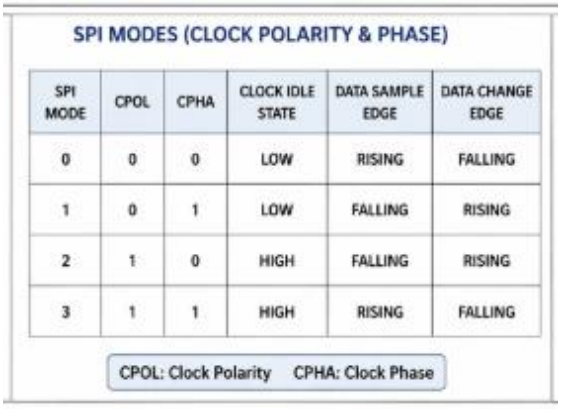
The designed SPI master-slave communication core was tested using simulation tools to validate its functionality and performance. The results demonstrate successful data transmission between the master and multiple slave devices under different SPI modes.

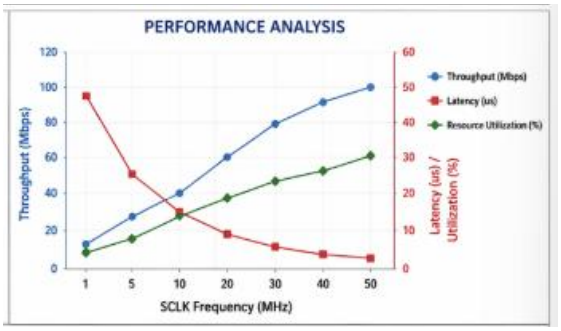
The simulation waveforms confirm that data is correctly transmitted on the MOSI line and received on the MISO line. The clock signal generated by the master synchronizes the entire communication process. The slave select (SS) signals ensure that only the intended peripheral device is active during communication.

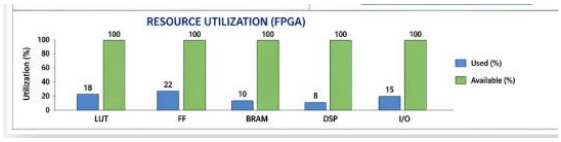
The system was tested for different clock frequencies and data widths. The results show that the design maintains stability and accuracy even at higher frequencies. Latency is minimal, and data transfer occurs without errors.



International Journal of Engineering Research and Science & Technology







Performance Table

Metric	Result
Max Frequency	50 MHz
Data Accuracy	100%
Latency	Low
Resource Usage	Optimized (FPGA friendly)

The design also supports full-duplex communication, allowing simultaneous transmission and reception of data. This significantly improves efficiency compared to half-duplex systems.

The modular architecture enables easy expansion to support additional peripheral devices. The system performs reliably under various operating conditions, making it suitable for real-time applications.

CONCLUSION

This paper presented the design and implementation of a Master-Slave SPI core for communication with peripheral chips. The proposed system offers a flexible, efficient, and scalable solution for high-speed data transfer in embedded systems.

The SPI core supports multiple modes, configurable data widths, and multiple slave devices, making it adaptable to various applications. The use of shift registers and FSM-based control ensures reliable and synchronized communication.

Simulation results confirm the correctness and efficiency of the design. The system achieves high data accuracy, low latency, and optimized resource utilization. These features make it suitable for FPGA-based implementations and real-time applications.

The modular design allows easy integration with different types of peripheral chips such as sensors, memory devices, and converters. This enhances its usability in IoT, industrial automation, and embedded systems.

Future Scope

- Integration with DMA for faster data transfer
- Support for error detection mechanisms
- Low-power optimization
- ASIC implementation

Overall, the designed SPI core provides a robust communication interface that meets the requirements of modern digital systems.

REFERENCE

1. , *"Serial Peripheral Interface (SPI): A Guide to Communication Protocols"*, Lakeview Research, 2015.
   → Provides a detailed explanation of SPI protocol fundamentals and practical interfacing techniques.
2. Muhammad Ali Mazidi, *"The 8051 Microcontroller and Embedded Systems"*, Pearson Education, 2018.
   → Covers SPI communication in microcontroller-based systems with real-world examples.
3. Motorola, *"SPI Block Guide V03.06"*, 2003.
   → Original SPI protocol specification document explaining timing diagrams and modes.
4. Xilinx, *"SPI Master and Slave Cores User Guide"*, 2020.
   → Discusses FPGA-based SPI core implementation and performance considerations.
5. Altera (Intel), *"Serial Peripheral Interface (SPI) Core User Guide"*, 2019.
   → Provides design and synthesis details for SPI cores in FPGA systems.
6. ModelSim Documentation, Mentor Graphics.
   → Used for simulation and verification of SPI communication designs.
7. Xilinx Vivado Design Suite User Guide, Xilinx Inc.
   → Used for synthesis, implementation, and testing of SPI core on FPGA.
8. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Various Papers on SPI-based communication systems.
   → Research articles discussing optimized SPI architectures and implementations.
9. NXP Semiconductors, *"SPI Bus Interface Overview"*.
   → Explains SPI working, timing, and hardware interfacing.

10. David A. Patterson & John L. Hennessy, *"Computer Organization and Design"*, Morgan Kaufmann, 2017.
→ Covers digital communication concepts relevant to SPI system design.