International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

# Design and Simulation of Single Master Single Slave Serial Peripheral Interface (SPI) with Adaptive Baud Rate Using Verilog

*Mrs. P. Shalini[1], G. Shalini Priiya[2], Ch. Sudeepthi[3], M. Mounika[4], SK. Abdul Waseem Abbas[5], SK. Safin[6]*
[1]*Assistant Professor, Dept. Of ECE, PBR VITS, Kavali, Nellore District, Andhra Pradesh, India*
[2-6]*UG Students, Dept. Of ECE, PBR VITS, Kavali, Nellore District, Andhra Pradesh, India*

---

**Abstract -** This paper presents the design and simulation of the Serial Peripheral Interface (SPI) communication protocol using Verilog Hardware Description Language (HDL). The implementation focuses on establishing reliable communication between a master and a single slave device. The design is simulated using Vivado software targeting an Artix-7 FPGA platform. The SPI protocol is widely used in embedded systems due to its simplicity, high speed, and full-duplex communication capability. In this work, key SPI signals such as Serial Clock (SCLK), Master Out Slave In (MOSI), Master In Slave Out (MISO), and Chip Select (CS) are implemented and analysed. The results demonstrate successful data transmission between master and slave through simulation waveforms. The proposed design ensures efficient synchronization between master and slave devices with minimal latency. The simulation results validate the reliability and robustness of the SPI protocol for high-speed embedded system applications.

***Key Words***: SPI, Verilog HDL, FPGA, Vivado, Serial Communication, Master-Slave

## 1. INTRODUCTION

In modern digital systems, efficient communication between hardware components is essential for achieving high performance and reliability. Serial communication protocols play a crucial role in enabling data exchange between processors, microcontrollers, and peripheral devices. Among these, the Serial Peripheral Interface (SPI) protocol is widely used due to its simplicity, high speed, and ability to support full-duplex communication. It is commonly applied in embedded systems for interfacing sensors, memory devices, and display modules.

SPI operates in a master-slave configuration where a single master device controls one or more slave devices using four main signals: Serial Clock (SCLK), Master Out Slave In (MOSI), Master in Slave Out (MISO), and Chip Select (CS). The master generates the clock signal and initiates communication, while the slave responds accordingly. Compared to other communication protocols such as I2C and UART, SPI provides faster data transfer rates and a straightforward hardware implementation, making it suitable for high-speed applications.

With the growing demand for efficient hardware-based communication systems, designing protocols using Hardware Description Languages like Verilog has become increasingly important. In this work, the SPI protocol is designed and simulated using Verilog HDL in Vivado, targeting an FPGA-based environment. The implementation focuses on a single master and single slave configuration, and the functionality is verified through simulation waveforms, demonstrating accurate data transmission and proper synchronization between devices.

## 2. LITERATURE SURVEY

The Serial Peripheral Interface (SPI) protocol has been widely studied and implemented in embedded systems due to its high-speed communication and simple architecture. Many researchers have focused on designing SPI master and slave modules using Hardware Description Languages such as Verilog and VHDL. These studies emphasize the importance of proper clock synchronization and signal control to ensure accurate data transmission between devices. FPGA-based implementations of SPI have gained popularity because they provide flexibility, reconfigurability, and efficient testing of digital designs before hardware realization.

Several works in the literature have explored optimization techniques to enhance the performance of SPI communication systems. Researchers have proposed improved designs using finite state machines (FSMs) to manage data transfer efficiently and reduce latency. Some studies also focus on reducing power consumption through clock gating and efficient resource utilization. In addition, multi-slave SPI configurations have been analysed, where a single master communicates with multiple slave devices using separate chip select lines, although this increases system complexity.

## 3. EXISTING SYSTEM

In existing systems, the Serial Peripheral Interface (SPI) protocol is typically implemented using microcontroller-based designs or pre-built hardware modules, where communication between master and slave devices is managed through fixed configurations. These implementations rely on standard SPI controllers that handle clock generation, data shifting, and chip select operations internally, reducing design complexity but limiting flexibility.

## 4. PROPOSED SYSTEM

The proposed method presents the design and simulation of the Serial Peripheral Interface (SPI) protocol using Verilog Hardware Description Language. The system is developed to achieve reliable and high-speed communication between digital components in an FPGA-based environment. The design emphasizes simplicity, accuracy, and flexibility by implementing a single master and single slave configuration.

The overall architecture of the system consists of two main modules: the SPI Master and the SPI Slave. The interaction between these modules is achieved through four primary signals: Serial Clock (SCLK), Master Out Slave In (MOSI), Master In Slave Out (MISO), and Chip Select (CS).

The design supports full-duplex operation, allowing simultaneous transmission and reception of data. Proper timing control is maintained to ensure that data is sampled and shifted correctly on clock edges. This structured communication mechanism enhances the reliability and performance of the SPI system.

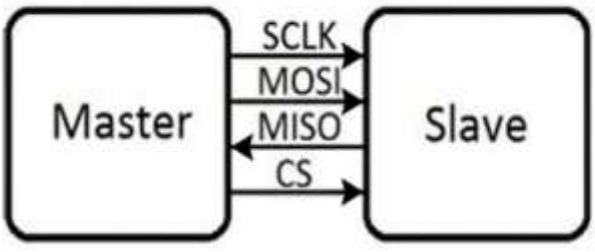


**Fig.4.1**: Block Diagram of SPI Communication System

The SPI master module is responsible for generating the clock signal and initiating data transfer. It sends data serially to the slave through the MOSI line and simultaneously receives data through the MISO line. The master also controls the Chip Select signal to enable or disable the slave device during communication.

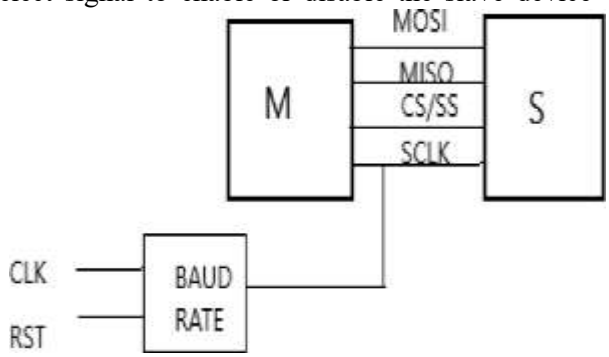


**Fig.4.2:** Block diagram of SPI with Adaptive baud rate

The architectural layout of the proposed system is illustrated in Fig. 4.2. The system is fundamentally divided into three primary functional blocks: the Baud Rate Generator, the SPI Master (M), and the SPI Slave (S).

### ADVANTAGES

- Enhanced Flexibility and Device Compatibility
- Dynamic Power Efficiency
- Optimized Data Throughput Renewable
- Logic Resource Optimization on FPGA
- Mitigation of Signal Integrity Issues
- Sustained Full-Duplex Efficiency

### APPLICATIONS

- Internet of Things (IoT) Sensor Networks
- Smart Wearables and Battery-Operated Devices
- Multi-Speed Peripheral Bridging
- Industrial Automation and Noisy Environments
- FPGA-Based Embedded Co-Processors

## 5. RESULTS AND DISCUSSIONS

To evaluate the performance and efficiency of the proposed Single Master Single Slave SPI with an adaptive baud rate, the design was synthesized and simulated using the Xilinx Vivado tool. The performance was benchmarked against a conventional, static-rate SPI implementation in terms of area utilization, power consumption, and propagation delay.

**Fig. 5.1**: Simulation waveform

The functional behaviour of the SPI protocol was verified by observing the generated simulation waveforms. The control logic successfully initiated communication by pulling the Chip Select (CS) line low. The adaptive baud rate generator successfully established a dynamically scaled SCLK. The full-duplex transmission was validated as the Master module shifted data out on the MOSI line while simultaneously shifting in data from the Slave on the MISO line. The system maintained perfect bit-synchronization without any data loss or frame overlaps during clock transitions.

| Name | Slice LUTs (134600) | Slice Registers (269200) | Bonded IOB (400) | BUFGCTRL (32) |
|---|---|---|---|---|
| SPI_TOP | 47 | 63 | 20 | 2 |
| MISO_UUT (SPI_MISO) | 19 | 23 | 0 | 0 |
| MOSI_UUT (SPI_MOSI) | 20 | 23 | 0 | 0 |

**Fig. 5.2**: Area utilized

This reflects a notable reduction in area. The compact nature of the proposed design is attributed to the elimination of redundant static registers and the implementation of an optimized, shared-counter clock division algorithm in Verilog.


**Fig. 5.3**: Power report

Power analysis was performed to gauge the system's thermal and energy efficiency.

- The existing method exhibited a total power dissipation of **1.588 W**.
- The proposed method reduced this consumption to **1.391W**.

The reduction in power is primarily due to the adaptive clocking mechanism. By scaling down the clock frequency during lower throughput demands, the active switching capacitance is minimized, leading to a direct drop in dynamic power consumption.

Fig. 5.4: Delay analysis

- The critical path delay for the existing method stood at **4.200 ns**.
- The critical path delay for the proposed method was measured at **3.994 ns**.

The slight improvement in propagation delay implies that removing complex, multi-stage static dividers streamlined the logic path, allowing electrical signals to route faster across the synthesized FPGA cloth.

## 6. CONCLUSION AND FUTURESCOPE

### CONCLUSION

This paper presented the design and simulation of a Single Master Single Slave Serial Peripheral Interface (SPI) with an adaptive baud rate utilizing Verilog HDL on an FPGA platform. The system successfully maintained the core advantages of the traditional SPI protocol—such as simple four-wire full-duplex communication and synchronous data transfer—while introducing dynamic flexibility to the bus speed.

Future extensions of this research will explore the integration of built-in self-test (BIST) capabilities directly on the FPGA to further verify chip performance in real-time, as well as extending the controller to handle multi-master and multi-slave topologies over the same adaptive interface.

## FUTURE SCOPE

The future scope of this research lies in expanding the SPI architecture to support multi-master and multi-slave configurations, enabling more complex communication networks within high-performance System-on-Chip (SoC) environments. Integrating Built-In Self-Test (BIST) capabilities and error-correction algorithms, such as Cyclic Redundancy Check (CRC), would further enhance the system's reliability for critical applications in aerospace and medical electronics.

## REFERENCES

[1] Rahul Jandyam, 2Sanjay Reddy Kandi, (2017) Design and Implementation of SPI Module in Verilog HDL using FPGA Design Flo.

[2] Dr Baswaraj Gadgay (2015) FPGA Implementation of Serial Protocol Bridge using SPI and I2C. Control & Automation, (ISSN: 2348-4748, Volume 2, Issue 6, June 2015)

[3] Shumit Saha, Md. Ashikur Rahman, Amit Thakur (2014) Design and Implementation of SPI Bus Protocol with Built-In-Self-Test Capability over FPGA. International Conference on Electrical Engineering and Information & Communication Technology

[4] W.L.Li, D.P. Y. Implementation of Asynchronous Serial Port and Synchronous Serial Port Conversion Using FPGA. Electronic Engineer, pp.52-53.

[5] S.Zhang, W.Li. Modular design method of FPGA. Journal of Electronic Measurement and Instrument, pp. 560-565.

[6] W.C.Zhu, S. ,Zhang, H.L.Jiang. (2017) Design of high-speed data communication interface based on ARM and FPGA. Journal of Guilin University of Electronic Technology, pp. 293-297

[7] F.J.Sun, C.X.Yu. Verilog Implementation of SPI Serial Bus Interface. Modern Electronic Technique, pp. 105-106,109.

[8] W.Mai, W.Liu. Design and Implementation of SPI Interface Based on FPGA and MSP430. Instrumentation Users, pp. 100-102.

[9] Design and implementation of a high-speed Serial Peripheral Interface