

# Design of a SPI Master Controller with 4-Mode and 4-slave support with a FIFO buffer

1<sup>st</sup> Carlos Torres Valle

Department of Computer Engineering

University of Texas at San Antonio

San Antonio, United States

carlos.torresvalle@my.utsa.edu

**Abstract**—External communication is a critical part on modern System-on-Chip (SOC) design, requiring efficient and flexible communication interfaces. This project focuses on the VLSI implementation of a high performance SPI (Serial Peripheral Interface) masters controller using a 180 nm technology from TSMC, this design support 4-Modes of SPI for CPOL (Clock Polarity) and CPHA (Clock Phase), additionally, this SPI controller support up-to 4 independent Chip Select ( $\bar{CS}$ ) Lines. This design also contain an 4-Entries FIFO Buffer (First-In First-Out) to support efficient streaming data transfer, this will eliminate the need of the CPU to wait for the serial transaction to complete. The timing analysis demonstrated achieved a maximum frequency of 285.5 MHz, exceeding the initial 200 MHz target, additionally, a secondary low power synthesis using automated clock gating reduced the dynamic power consumption by 52.5%.

**Index Terms**—SPI, FIFO, VLSI, SOC, Communication Protocol

## I. INTRODUCTION

The integration of complex peripherals devices into embedded systems has required the need for reliable, high speed communications protocols. One of the most important one is the Serial Peripheral Interface (SPI) which is considered the standard for short distance synchronous communication due to his full duplex capability and simple hardware requirements. SPI is widely used in the industry to communicate micro controllers with sensors, flash memory, display controller and other external devices essential on System-on-Chip architectures.

The standard SPI operates on a Master-Slave topology using 4 wires; The first wire is for a Serial Clock (SCLK), the second wire is for Master Out Slave In (MOSI), the third wire is Master In Slave Out (MISO) and the last one is Chip Select (CS). The Master controller is responsible for initializing the communication, generating the serial clock signal and managing the selection of slaves devices. While the protocol is simple and efficient to implement. In traditional designs, the CPU often stalls for serial transaction to complete, creating a bottleneck.

This paper presents the design, behavioral simulation, and synthesis of an SPI Master controller implemented in a TSMC 180 nm technology. To avoid the CPU bottleneck, this design implements an 4-Entries FIFO buffer which decouple the CPU with the serial transmission.

Another features of this design, is the configurable clock serial generator that supports all the 4 SPI modes (CPOL and CPHA) and dedicated hardware to control up to four slaves devices. The primary objective of this project was to create a full synthetizable Verilog code for high speed performance and low area utilization.

The base line design has a maximum frequency of 285.5 MHz with a power consumption of 0.77 mW, however, we optimized this design for low power consumption using an automated integrated clock gating (IGC); This optimization yield a result of a maximum frequency of 224.7 MHz and a power consumption for 0.34 mW.

## II. SYSTEM ARCHITECTURE

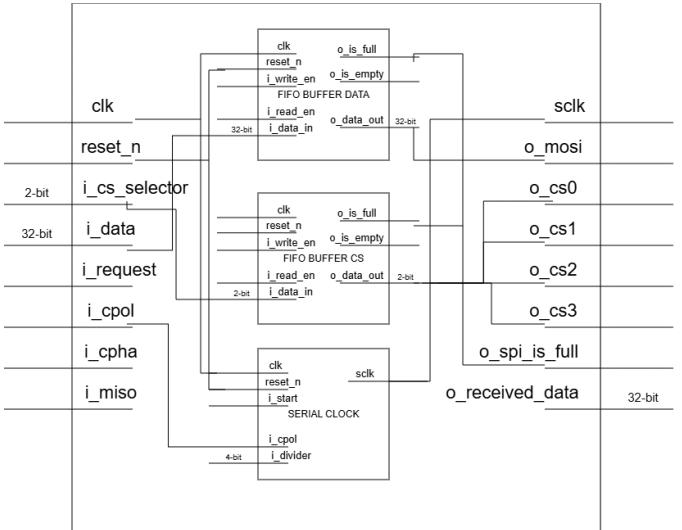


Fig. 1. SPI Master Block Diagram

The proposed SPI Master controller is designed as modular component. The overall architecture is divided into three primary sub modules to ensure scalability and ease of verification. The FIFO Buffers, the Serial Clock Generator and the SPI Logic.

### A. FIFO Buffers

To decouple the CPU from the serial transaction, we implemented two FIFO buffers, one for saving the 32-bit input data and the other buffer for saving the chip select using 2-bits.

- Function: The CPU enable the signal for the data and chip select for one clock cycle and then this data is written into their respective buffers. Our SPI master controller monitor the buffers for any data to transmit, if it found any data, it will start the transmission, right away.

### B. Serial Clock Generator

The serial clock generator allows us to derive a specific SPI baud rate from the system clock (clk).

- Configurability: It supports a configurable clock divider to match different slaves speed requirements.
- Mode Support: This module automatically adjust the idle state of the serial clock depending on the input clock pole signal, which is required to support all of the SPI Modes.

### C. Finite State Machine (Control Logic)

This is the core control logic of our SPI master controller, which will allow us to transmit and receive data at the same time.

- IDLE: In this state, the SPI is waiting for the i\_request signal to be high.
- WRITE\_FIFO: The data that was provided from the CPU is written into the FIFO buffers (Input data and Chip Select).
- SETUP: The SPI reads the data that is in FIFO and start the serial clock, it also assert the chip select line (o\_cs).
- TRANSMITTING: At this point we are sending and receiving data on a positive or negative edge of SCLK (Depending on the CPHA input), once it finish, it will move to the IDLE state.

## III. METHODOLOGY

### A. Design and Behavioral Simulation

The hardware logic of this module was described using Verilog HDL. The functionality was verified using a behavioral simulation with AMD Vivado 2025.1, ensuring that the data data was correctly serialized.

*1) Behavioral Simulation Results:* For this simulation, we adjust our module to handle a data width of 4 bits and we set the CPOL to be equal to 1 and CPHA equal to 0. With this configuration, we got our expected result from the waveform.

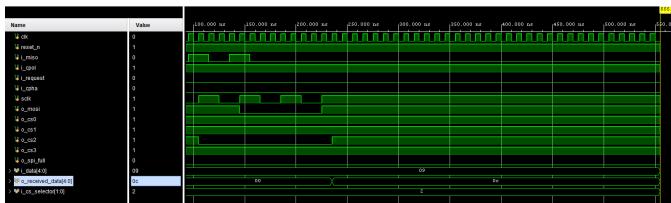


Fig. 2. Behavioral Simulation Waveform

### B. Logic Synthesis

The RTL code was synthesized using Cadence RTL Compiler (RC) targeting the TSMC 180 nm CMOS technology library. The operation voltage was set to 1.8 V. The synthesis process was conducted twice, the first one without

any optimization in place and the second with clock gating optimization enabled.

- Baseline Synthesis: The initial synthesis yield a maxF = 285.5 MHz and the design was left unconstrained regarding the area to determine the maximum theoretical speed of the logic paths.
- Synthesis of Clock Gating Power optimization: To improve the power consumption of our SPI controller, we implemented an automated clock gating. We implemented this by enabling the synthesis attribute *lp\_insert\_clock\_gating*, This instructed the compiler to look for registers with an enable condition (Registers that depends on an If condition), specially within FIFO and state registers and replace standard multiplexers feedback loops with integrated Clock Gating (IGC) cells.

### C. Area and Power Analysis

Post synthesis reports were generated to quantify the results. The area was measured in square micrometers  $\mu\text{m}^2$  and converted to equivalent NAND2 gate counts ( $10 \mu\text{m}^2$  per gate for this technology, see [2]). Power analysis was conducted by estimating switching activity.

## IV. SYNTHESIS RESULTS AND ANALYSIS

### A. Baseline Performance Analysis

The initial synthesis run was performed without any power optimization in place.

- Operation Frequency: The design achieved a critical path delay of 3.502 ns which yields a maximum frequency of ( $F_{\max}$ ) of 285.5 MHz. This significantly exceeds the initial design requirement of 200 MHz. Also, this demonstrate that our critical path (Which has 7 levels) was highly optimized for speed.
- Area Utilization: The baseline design has a total area of  $27,699 \mu\text{m}^2$ . Based on the standard NAND2X1 gate area of  $10 \mu\text{m}^2$  [2] has an approximately of 2,770 gates.
- Power Consumption: The estimated total power was 0.77 mW. The dynamic power accounted for the 99.9% of the total power, additionally the FIFO buffers were consuming 48% of the total energy due to the constant clock switchin activity in th registers.

### B. Clock Gating Optimization analysis

In order to reduce the power consumption of our module, we implemented a secondary synthesis with automated clock gating enabled (*lp\_insert\_clock\_gating*). This process replaced multiplexer based loops with Integrated Clock Gating (IGC) cells.

Metric	Baseline Design	Clock Gating Optimization	Change
Total Power	0.77 mW	0.37 mW	52.2% Reduction
FIFO Power	0.37 mW	0.10 mW	72.7% Reduction
Cell Count	1,062 Cells	614 Cells	42% Reduction
Gate Count	2,769 Gates	2,342 Gates	15.4% Reduction
Total Area	$27,699 \mu\text{m}^2$	$23,425 \mu\text{m}^2$	15.4% Reduction
Max Frequency	285.5 MHz	224.7 MHz	21.3% Decrease

TABLE I  
COMPARATIVE SYNTHESIS RESULTS

## V. CONCLUSION

This project successfully presented the VLSI design and implementation of a configurable SPI master controller using TSMC 180 nm technology. The proposed architecture met all the function requirements, support all the four SPI modes, manage independent slave devices with full duplex capability and decouple the CPU from the serial transmission using FIFO buffers.

The synthesis analysis highlighted the tradeoffs between the two synthesis version of our design. While the baseline synthesis has higher maximum frequency, most of the peripheral devices for SPI slaves usually operates between 10-50 MHz range, therefore, the final optimized version with clock gating will be more valuable to us due to its lower power consumption and smaller area which could play a critical role for modern low power System-on-Chip environments.

## REFERENCES

- [1] TSMC, “TSMC 0.18  $\mu\text{m}$  component parameters” pp. 129–130, Oct 1. 2001.
- [2] A. Ghanekar, B. Kishor, and S. Bandewar, “Design and implementation of SPI bus protocol,” Int. J. Adv. Res. Electr. Electron. Instrum. Eng., vol. 5, no. 5, pp. 4155–4157, May 2016.