# Physical Design Implementation of SPI using Q-Flow

Maven Silicon Project Report

Kanchan Kumari

**Abstract**

This report presents the physical design and implementation of a Serial Peripheral Interface (SPI) controller using Verilog HDL and the Q-Flow toolchain. The project involved front-end RTL design, functional verification, and back-end implementation steps such as synthesis, floorplanning, placement, routing, and design verification. The final layout was validated with Design Rule Checks (DRC), Layout versus Schematic (LVS) checks, and the generation of the GDSII format suitable for fabrication. The methodology and outcomes demonstrate the complete ASIC design flow from RTL to GDSII using open-source tools.

## 1. Introduction

Serial Peripheral Interface (SPI) is a widely used synchronous serial communication protocol that enables high-speed data exchange between a master and multiple slave devices. In this project, the SPI controller was designed in Verilog and implemented using the Q-Flow toolchain to demonstrate a complete ASIC design cycle. The design focused on correctness, timing optimization, and physical verification to meet standard cell-based implementation requirements.

## 2. Design Methodology

The implementation flow followed in this project can be divided into two major phases: *Front-End Design* and *Back-End Implementation.*

### 2.1 Front-End Design

- **RTL Design:** An SPI controller was designed and coded in Verilog HDL.
- **Simulation and Verification:** The RTL functionality was verified using testbenches to ensure correct operation in different modes of SPI communication.

### 2.2 Back-End Implementation

- **Logic Synthesis:** The RTL was synthesized using Yosys, and the design was mapped to a standard cell library.
- **Floorplanning:** Defined the chip outline, power grid structure, and placed I/O pads.
- **Placement and Routing:** Standard cells were placed, followed by signal routing using the Q-Flow automated flow.
- **Timing Analysis:** Static Timing Analysis (STA) was performed to identify and optimize critical paths, ensuring timing closure.
- **Physical Verification:** DRC and LVS checks were conducted to confirm design compliance and consistency between the layout and schematic.
- **Final GDSII:** A manufacturable GDSII file was generated, suitable for fabrication.

## 3. Results and Observations

- Functional verification confirmed correct SPI data transfer between master and slave modules.
- The synthesis step successfully mapped the RTL to the chosen standard cell library with minimal area and power overhead.
- Floorplanning and placement achieved efficient utilization of chip area.
- Routing achieved complete signal connectivity with no congestion.
- Timing analysis showed that performance improved after optimization of critical paths.
- DRC and LVS checks passed, validating the correctness of the physical layout.

## 4. Conclusion

The successful implementation of the SPI controller using the Q-Flow toolchain demonstrates the end-to-end VLSI design process using open-source EDA tools. The project covered all stages of physical design, from Verilog RTL to GDSII, emphasizing the importance of synthesis, timing optimization, and verification. The generated GDSII layout can be considered a tape-out ready design for fabrication.

## References

- Yosys Open Synthesis Suite – `https://yosyshq.net/yosys/`
- Q-Flow Digital Synthesis Flow – `http://opencircuitdesign.com/qflow/`
- Maven Silicon Training Resources