ECE 310
Lab 6 Report: PISO and SIPO Shift Registers

Ohm Patel

November 2, 2025

Abstract

This report presents the design, implementation, and verification of two 8-bit shift registers: a **Parallel-In Serial-Out (PISO)** and a **Serial-In Parallel-Out (SIPO)** architecture. Both modules were developed using synchronous Verilog logic and verified through timing simulations in Vivado. Each design demonstrates accurate bit propagation between parallel and serial domains under clock-driven operation. Simulation waveforms confirm correct functionality during reset, load, and shift phases, while synthesized RTL schematics validate hardware efficiency and alignment with theoretical design.

1 Introduction and Background

Shift registers are fundamental sequential elements used to convert data between parallel and serial formats. They consist of cascaded flip-flops that store and shift bits with each clock pulse, enabling controlled data flow in digital systems such as communication interfaces, memory buffers, and embedded pipelines.

- **Parallel-In Serial-Out (PISO):** Accepts eight parallel inputs simultaneously and shifts them out sequentially, one bit per clock cycle, starting from the least significant bit.
- **Serial-In Parallel-Out (SIPO):** Receives serial data one bit at a time and constructs an 8-bit parallel output after eight clock cycles.

Both designs employ synchronous, edge-triggered flip-flops with an active-high reset to ensure deterministic behavior from startup. Control signals are designed to be mutually exclusive, preventing conflicting load and shift operations.

2 Design and Block Diagrams

The conceptual architecture of both modules was first modeled using D flip-flops and multiplexers to represent load and shift control logic.

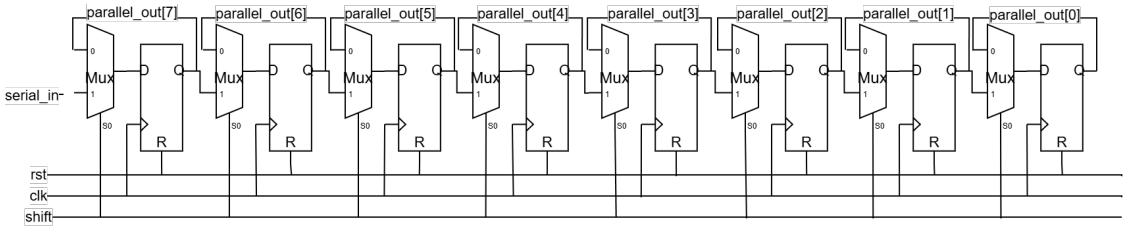

Figure 1: Block diagram of the 8-bit PISO register showing parallel load and serial shift logic.

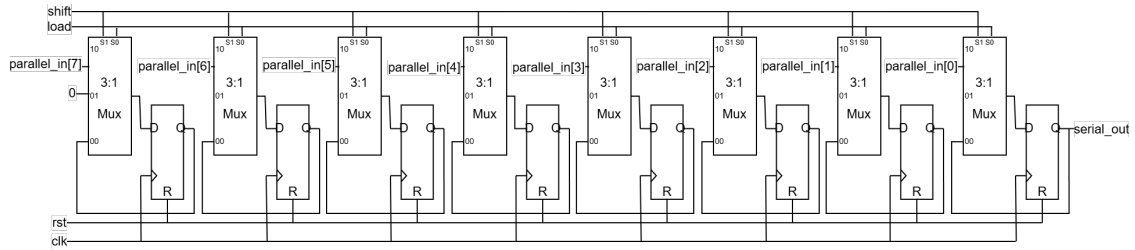

Figure 2: Block diagram of the 8-bit SIPO register showing serial data propagation and parallel output structure.

3 Verilog Implementation

3.1 PISO 8-bit Register

Listing 1: Verilog implementation of the 8-bit PISO register

```
module PISO_8bit(
    input [7:0] parallel_in,
    input clk, rst, shift, load,
    output reg serial_out
);

    reg [7:0] shift_register;

    always @(posedge clk) begin
        if(rst) begin
            shift_register <= 8'b0;
            serial_out <= 1'b0;
        end else if (load) begin
            shift_register <= parallel_in;
            serial_out <= 1'b0;
        end else if (shift) begin
            serial_out <= shift_register[0];
            shift_register <= {1'b0, shift_register[7:1]};
        end
    end
endmodule
```

3.2 SIPO 8-bit Register

Listing 2: Verilog implementation of the 8-bit SIPO register

```
module SIPO_8bit(
    input serial_in, clk, rst, shift,
    output reg [7:0] parallel_out
);

    reg [7:0] shift_register;

    always @(posedge clk) begin
        if(rst) begin
            shift_register <= 8'b0;
        end else if(shift) begin
            shift_register <= {shift_register[6:0], serial_in};
        end
    end

    always @(*) begin
        parallel_out = shift_register;
    end
endmodule
```

4 Testing and Verification

Thorough testing was conducted to validate the functional correctness, synchronization, and reliability of both modules. Each module was simulated in Vivado with clock generation, reset handling, and multiple data vectors to ensure robust behavior under all control conditions.

4.1 Testbench Methodology

- A 10 ns clock period was used for both designs.
- Active-high reset was asserted at the beginning of simulation to clear internal registers.
- For the PISO, two 8-bit input patterns (`10110110` and `01101101`) were loaded sequentially and shifted out bit-by-bit with intentional pauses between sequences.
- For the SIPO, the same two patterns were sent serially, with one bit transmitted per clock, verifying correct accumulation into the 8-bit parallel output.
- Edge cases were included to confirm that when neither `load` nor `shift` were active, internal states remained stable and unaltered.

4.2 Verification Objectives

Testing was performed to ensure:

1. Proper synchronization between control signals and clock transitions.
2. Accurate serialization and deserialization of data in correct bit order.
3. Consistent operation following resets and idle periods.
4. Identical functional results between behavioral and synthesized implementations.

The simulation results validated every objective. All transitions occurred on rising clock edges, output signals remained stable during idle intervals, and data propagation matched theoretical timing expectations.

5 Simulation Results

5.1 PISO Simulation

The PISO testbench demonstrated successful loading and serial shifting. After each `load` pulse, data was correctly output in LSB-first order over eight clock cycles. The final output stabilized at zero once all bits had shifted out.

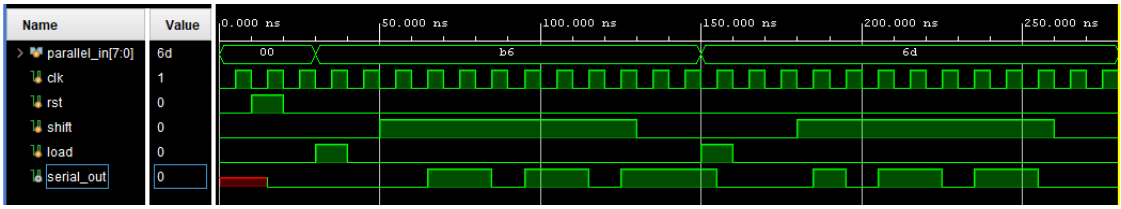

Figure 3: PISO simulation waveform showing correct parallel load, LSB-first shifting, and reset behavior.

5.2 SIPO Simulation

The SIPO testbench confirmed accurate reconstruction of serial input streams. Each new bit shifted in on the rising clock edge, and after eight cycles, the final parallel output reflected the complete byte sequence with no corruption or misalignment.

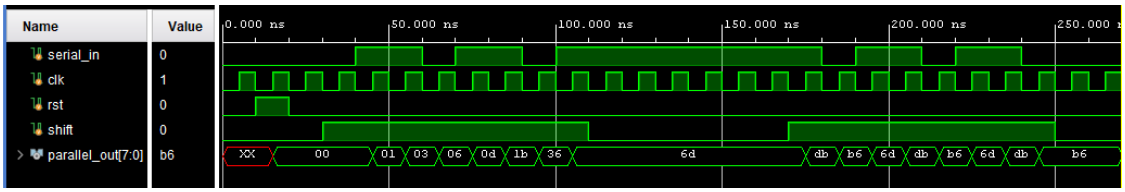

Figure 4: SIPO simulation waveform showing proper serial accumulation and final parallel output formation.

6 Results and Discussion

Both modules performed precisely as intended. Key observations include:

- Active-high reset correctly cleared all registers and outputs to zero.
- Clock synchronization ensured deterministic and glitch-free operation.
- Load and shift control signals were mutually exclusive, preventing logical contention.
- Bit order consistency was maintained across all test cases.

6.1 Synthesis Validation

Vivado synthesis produced RTL schematics matching the conceptual D flip-flop structures. Each design utilized eight flip-flops with minimal combinational logic, confirming efficient resource use and correctness of behavioral modeling.

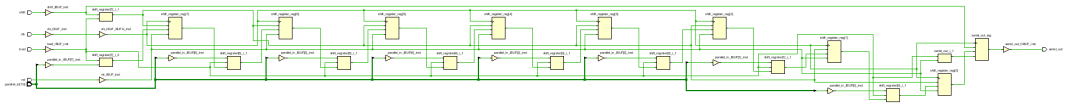

Figure 5: Synthesized RTL schematic of the PISO register.

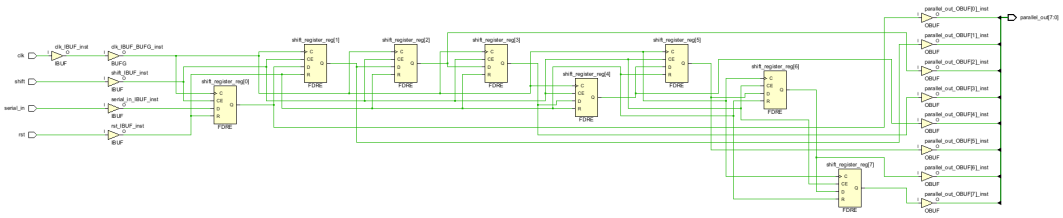

Figure 6: Synthesized RTL schematic of the SIPO register.

Synthesis analysis confirmed the absence of timing violations, with logic depth equivalent to a single D flip-flop per bit and simple control gating. Behavioral simulation and post-synthesis results were functionally identical, indicating full design integrity.

7 Conclusion

Two 8-bit shift registers, PISO and SIPO, were successfully designed, implemented, and verified through simulation and synthesis. Both designs exhibited stable, predictable behavior under clock-driven operation with synchronous control. Waveforms confirmed correct data flow, while synthesis results matched theoretical designs, validating both logic efficiency and correctness. These results reinforce fundamental principles of sequential digital design and demonstrate the reliable implementation of bidirectional data conversion between parallel and serial domains.

This report was typeset in LaTeX.