Lab 2A: 32-bit Barrel Shifter

Introduction

A barrel shifter is a combinational circuit capable of shifting or rotating data by a specified number of positions in a **single cycle**. Unlike serial shifters, which require multiple cycles to shift data bit by bit, a barrel shifter uses a network of multiplexers to perform the shift in parallel.

The design in this lab implements a **32-bit barrel shifter** that supports:

- Left and right shifts
- Logical shifts (with zero fill) and rotates (with wraparound bits)
- Shift amounts ranging from 0 to 31 (controlled by a 5-bit input)

This makes it suitable for high-performance data paths such as processors and DSP units, where low-latency shifting is critical.

Design Requirements

- **Data width**: 32 bits input and 32 bits output
- **Shift amount**: 5-bit control (0–31 positions)
- **Direction control**:
    - 0 = left shift/rotate
    - 1 = right shift/rotate
- **Action control**:
    - 0 = shift
    - 1 = rotate
- **Single-cycle operation**: purely combinational, result available without a clock delay

Design Methodology

The barrel shifter is implemented using **five stages of multiplexers**, corresponding to shifting by **1, 2, 4, 8, and 16 bits**. Each stage either shifts/rotates the data if the corresponding bit of the shift amount is set or passes the data unchanged.

- Stage 0: Input data
- Stage 1: Shift by 1 if shift_amt[0] = 1
- Stage 2: Shift by 2 if shift_amt[1] = 1
- Stage 3: Shift by 4 if shift_amt[2] = 1
- Stage 4: Shift by 8 if shift_amt[3] = 1
- Stage 5: Shift by 16 if shift_amt[4] = 1

The output of Stage 5 is the final result. This network ensures **O($\log_2$N)** mux depth (5 levels for 32-bit), minimizing delay compared to a long chain.

Design Analysis

- **LUT usage**: Estimated ~160–200 LUTs for a 32-bit shift/rotate network. Exact number depends on FPGA family and synthesis optimizations.
- **Critical path**: Input of 5 bits and levels of 2:1 multiplexer then goes to Output.
- **Pipelining**: Registers can be inserted between stages to reduce delay per stage, achieving higher frequency at the cost of latency.

References Attached:

- **Report**
- **Synthesis Report**
- **Simulation Screenshot from QuestaSim**

Conclusion

The 32-bit barrel shifter was successfully implemented in System Verilog using **five stages of muxes with always_comb blocks**. The design supports shift/rotate in both directions, meets single-cycle requirements, and synthesizes efficiently for FPGA implementation.