Parametrized Multi-Function Barrel Shifter

This lab is based on 3.12.1 and 3.12.2 from the textbook.

Parameterized Barrel Shifter

The barrel shifter covered in the lecture accepts 8-bit input signals only. We want to construct a parameterized version. A parameterized barrel shifter accepts input width specified by a parameter. To simplify the design, we assume the width of an input is 2^N and the N is used as the parameter

1. Design a parameterized rotate-**right** barrel shifter and derive the SystemVerilog code (call the module `param_right_shifter`).
2. Design a parameterized rotate-**left** barrel shifter and derive the SystemVerilog code (call the module `param_left_shifter`).
3. Write a testbench and use simulation to verify the operation of both shifters. (You can test both shifter with a single testbench i.e., instantiate both shifters to display both output together)

Multi-Function Barrel Shifter

A multi-function barrel shifter can rotate-right or rotate-left based on the value of a 1-bit control signal, `lr`.

1. Design and implement a parameterized multi-function barrel shifter using one rotate-right circuit, one rotate-left circuit, and one 2-to-1 multiplexer to select the desired result (call the module `multi_barrel_shifter_mux`).
2. The multi-function shifter can also be implemented by one rotate-right shifter with pre-and post-reversing circuits. A reversing circuit either passes the original input or reverses the input bitwise (e.g., if an 8-bit input is $a_7a_6a_5a_4a_3a_2a_1a_0$, the reversed result becomes $a_0a_1a_2a_3a_5a_5a_6a_7$). (call the module `multi_barrel_shifter_reverser`)

FPGA Implementation

1. Verify the functionality of the `multi_barrel_shifter_mux` by synthesizing it and programming it on the FPGA board (Specify `N = 3` and use 8 switches to specify the input and another switch to specify the control bit `lr`).
   (Check the report files and record the number of logic cells used and any other interesting metrics i.e., propagation delays…etc.)
2. Repeat the verification process for `multi_barrel_shifter_reverser` and record the number of logic cells used.
3. Upload a short video (no more than 5 minutes) demoing a functioning circuit. Include a link to the video in the `README.MD` file.
4. Include the number of logic cells for both implementations in a table inside `README.MD`.