**Lab 1 Report**

*Matthew Dragotto, Luke Moore, David Chen*

1. **Implementation:**
2. **Questions:**

1) What is the difference between structural and behavioral Verilog? Please provide an example of a structural and behavioral implementation of a multiplexer.

Structural verilog is implementing the desired function using logic gates. Behavior Verilog uses behavioral code such as if statements to perform the desired the function. For a multiplexer, behavioral Verilog uses the select signal as the condition for the if-else statement, whereas for a multiplexer using structural style, it is implemented with AND gates, NOT gates, and OR gates

2) What is the difference between an asynchronous and synchronous Multiplexer? Please provide a brief explanation on how you could implement both using behavioral Verilog.

The difference between an asynchronous and a synchronous multiplexer is that the synchronous mux selects its output from its inputs at the clock edge, where an asynchronous mux can select at any time regardless of the clock being high or low.

To implement an asynchronous mux, you could use the statement: always @ (sel or in0 or in1 or in2 or in3 or in4) to test whether any of the inputs change, followed by the if-else statements checking the new values of the inputs.

To implement a synchronous mux, you could use the statement: always @ (posedge clk), followed by if-else statements to test the different input combinations.

3) What is the difference between an arithmetic and logical shifter?

An arithmetic shifter preserves the sign bit and a logical shifter always shifts in zeros.

4) Assuming that you did NOT use Behavioral Verilog to implement an arithmetic shifter, how could you design one from scratch? Please include a simple diagram.

To design an arithmetic shifter from scratch, we need multiplexers for each bit of the shifter.