## Assignment #1

**Problem Statement:**

Derive the architecture of a 6 bit unsigned combinational controlled Incrementer/Decrementer. The architecture has a control input **P**. When P=0, the designed architecture works as an incrementer which increments the input by 1. When P=1, the designed architecture works as a decrementer which decrements the input by 1. Try to make the architecture optimized by using basic gates only.

**Problem: Write the Verilog Code for the above controlled Incrementer/Decrementer circuit which accepts a 6 bit input X and a controlled input signal P. Demonstrate the functionality of the circuit with exhaustive post route simulation outputs for a varied possible set of input combinations that leads either increment or decrement of the input depending on the control signal input. Follow a structural style of Verilog coding.**

## Assignment #2

**Problem Statement:**

Consider a left barrel shifter that accepts two inputs: a 8-bit unsigned data input A and a second input B denoting the shift amount, using ONLY 2:1 multiplexers as the sole circuit building block. Modify this circuit to derive a left-right barrel shifter which accepts an additional control signal R. When R = 0, the modified circuitry performs the original left shift operation; otherwise when R = 1, it performs a right shift operation on the same input data A with the amount of right shift dictated by the same shift signal B. The MODIFIED left-right barrel shifter must also comprise of 2:1 multiplexers ONLY. COMPLETE CREDIT WILL BE AWARDED FOR THE DESIGN DERIVED USING LEAST NO: OF 2:1 MULTIPLEXERS FOR THE MODIFIED LEFT-RIGHT BARREL SHIFTER.

**Problem: Write the Verilog Code for the MODIFIED left-right barrel shifter accepting 8-bit data input A, shift amount B and the control signal R, with exhaustive post route simulation outputs. Comment on the bit-width of signal B. Ignore any overflow or underflow of data. Approach the problem by writing the Verilog code of a 2:1 multiplexer and instantiate it requisite number of times through implicit or explicit association. Use ONLY 2:1 multiplexers for the entire design.**