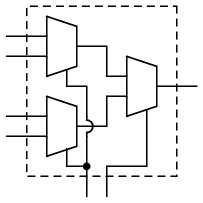
Lab 2

**In this lab, you should only use structural modelling, in other words you shouldn't use always@ .... , loops etc.**

1. a) First, build a 2x1 multiplexer (2 input, each input is 1 bit). Then by using this, design a 4x1 multiplexer (4 inputs, each input 1 bit). **( /2.5)**

You must use the same module that you created for 2x1 mux, the goal here is to learn how to combine modules and use them again, like the functions in C/C++



b) Build a 4 bit 2x1 multiplexer (2 inputs, each input is 4 bit). **( /1.5)**

**Example Code For Using The Same Module**

FullAdder s0( .A( a[0] ), .B(b0), .C( op ), .Sum( sum[0]), .Cout( ripple0 ) );

FullAdder s1( .A( a[1] ), .B(b1), .C( ripple0 ), .Sum( sum[1]), .Cout( ripple1 ) );

...

...

...

FullAdder sN( .A( a[N] ), .B(bN), .C( rippleN ), .Sum( sum[N]), .Cout( carry ) );

1. In this problem, you will implement an 8 bit ripple carry adder/subtractor **( /4)**

The inputs will be two 8 bit numbers in 2’s complement form, and a SEL signal that is used to select between addition (SEL = 0) and subtraction (SEL = 1). The output should be of 8 bits, along with an overflow detector OFLO.

1. Implement a comparator. **( /4)**

A comparator takes two binary 2 bit numbers a, b as inputs and outputs 1 if a >= b and zero otherwise.

1. Carry look ahead adder(4 bit) **( /4)**

The fundamental reason why the ripple carry adders are slow is that the carry signals must propagate through every bit in the adder. Carry look ahead adder aims to eliminate this problem (at the cost of increased circuit complexity).

1. 4 bit multiplier **( /4)**

In this problem, you will design a multiplier based on the given reference design. Your circuit must takes 2 inputs A and B, each input is 4 bits. your Output will be 8 bit.

Reference: [piazza-resourece-multiplier.pdf](https://s3.amazonaws.com/piazza-resources/i4tbv8xatvp6iu/i5ewlsx2fd12hn/multiplier.pdf?AWSAccessKeyId=AKIAJKOQYKAYOBKKVTKQ&Expires=1422350960&Signature=kq%2FvlJZFYhJmYG2I7EN%2BggmzFys%3D)