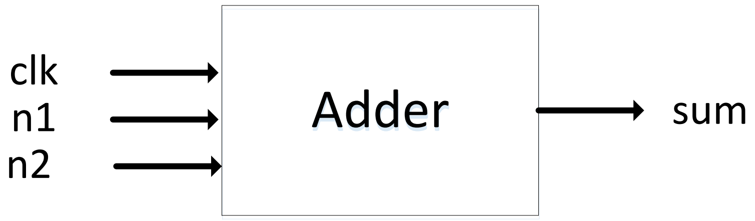
# ECE 426/516 Implementation of VLSI Systems with HDL
# Lab 6 Arithmetic Circuit Design with Pipelines

**Due date: May 5th, 2024, by 11:30pm**

## Section A

In this section, you will code and test a design for adding two signed numbers of width 16 bits each at a time without any pipelining.



## Lab Procedure

1. Use the following module template for this adder design.

```
// Addition of two 16 bits, 2's complement numbers, n1 and n2.  All the 16
bits addition at one stroke without any pipeline. Result is 17 bits.

module adder_a (clk, n1, n2, sum) ;
input clk ;
input [15:0]  n1 ;
input [15:0]  n2 ;
output [16:0] sum ;

reg    [16:0]  sum;
wire   [16:0]  sum_i;

assign sum_i = ________;  // Add after sign extension - Result (i).
always @ (posedge clk)         // Pipeline 1, clk (1), register final result (i).
          begin
                    sum <= __________;
          end
endmodule
```

2. Write a test bench to verify the functionality. Synthesize this design and report the number of gates and operating clock cycle.

## Section B

### Lab Procedure

1. In this section, you add pipeline stage. **8 bits are used at every pipeline stage**. Use the following template to complete your Verilog coding.

```
// Addition of two 16 bit, 2's complement nos., n1 and n2. 8 bits addition at a time.
Result is 17 bits.

module adder_b (clk, n1, n2, sum) ;

input   clk ;
input   [15:0]  n1 ;
input   [15:0]  n2 ;
output  [16:0]  sum ;

reg [16:0]      sum ;
wire [8:0]      sum_LSB ;
reg  [8:0]      sum_LSB_1 ;

reg  [15:8]     n1_reg1 ;
reg  [15:8]     n2_reg1 ;

wire    [16:8]  sum_MSB ;
wire    [16:0]  sum_next ;

assign  sum_LSB = ____ ; // Add least 8 significant bits. sum_LSB [8] is the carry.

always @ (posedge clk)  // Pipeline 1, clk (1), register LSB to continue addition of
MSB.
          begin
            sum_LSB_1 <=  _________;              // Preserve LSB sum
               n1_reg1   <= __________;              // Preserve MSBs of n1
               n2_reg1   <= __________;              // Preserve MSBs of n2
                   end
```

```
// Extend sign & add msbs with carry.

assign  sum_MSB = ____________________; // Add MSBs with carry.
assign sum_next = ______________________;

always @ (posedge clk)          // Pipeline 2, clk (2), register result.
                    begin
                              sum <= ___________;
                    end
endmodule
```

2. Write a test bench to verify the functionality. Synthesize this design and report the number of gates and operating clock cycle.

## Demo

You should demo the following aspects of your design to TA.

1. Verilog code of your design
2. Your simulation waveforms and your synthesized results.