# EE 3120: Digital Circuits Laboratory
# LAB 5: ARITHMETIC AND LOGIC UNIT DESIGN

**Objectives**

The objectives of this laboratory assignment are:

- To design a 1-bit full adder module.
- To use the above full adder module for the design of n-bit adder and subtractor circuits.

**Laboratory Instructions**

- Create the Verilog source file(s) for your designs before coming to the lab. You can use the Xilinx HDL editor or any text editor to create your files. Remember to bring these files to the lab.
- Use the Xilinx software to create a new project and then copy your Verilog (.v) to that new project directory. Remember to Add your Verilog file to this new project (see the EE 3120 Laboratory Tutorial for details).
- Perform functional simulation of your design and have it checked by the lab instructor or your TA before proceeding with the implementation.
- If the circuit works as expected, implement it using the prototyping board assigned to you.
- Use switches and LEDs available on the breadboard to test and demonstrate your circuit to the lab instructor or your TA.
- Before you leave the lab please remove any files or directories you created on your lab PC and leave your workplace at least as clean and tidy as you found it.

**Pre-lab Work**

Complete your design and its Verilog implementation and bring your Verilog program on a floppy disk. You can use any text editor to create an Verilog file. If you have any problems with Verilog syntax and other pre-lab related issues, please resolve them with the instructor or your TA before coming to the lab. Your TA may not be able to help you with these issues during the lab session.

**Laboratory Report Instructions (due at the start of next lab)**

Your report should be typed and prepared as per the guidelines given on the EE 3120 web page. For each design of this lab, submit the following:

- Detailed state diagram for each sequential logic circuit.
- Documented listing of your Verilog source file(s) with appropriate pin assignments for the top level files in the same or separate file(s).
- Simulation waveforms: Label the waveforms to indicate proper operation of your circuit(s).

**DESIGN PROBLEMS**

Using the Xilinx CAD tools, design, test and demonstrate arithmetic and logic circuits which implement the following functions. Your circuits should be as small as possible.

1. **2's Complement Adder/Subtractor**: Design a 1-bit full adder module in Verilog

and use it to implement a 3-bit arithmetic circuit for 2's complement represented numbers. The arithmetic unit has two 3-bit signed inputs ($A=a_2a_1a_0$ and $B=b_2b_1b_0$), an operation input (OPR) and produces a 3-bit output result ($S=s_2s_1s_0$), and an overflow flag (OVFL). The circuit should perform 2's complement addition (A+B) when OPR=0 and 2's complement subtraction (A-B) when opr=1.

2. **Sign-Magnitude Adder/Subtractor**: Use the above full-adder to implement a 3-bit arithmetic unit for sign-magnitude represented numbers. The arithmetic unit has two 3-bit sign-magnitude inputs ($A=a_2a_1a_0$ and $B=b_2b_1b_0$), an operation input (OPR), and produces a 3-bit output result ($S=s_2s_1s_0$), and an overflow flag (OVFL). The circuit should perform sign-magnitude addition (A+B) when OPR=0 and sign-magnitude subtraction (A-B) when OPR=1. You may use the n-bit sign-magnitude addition algorithm described below.

EAC = End Around Carry;

- **begin**
  OVFL = 0;
  **if** ($a_{n-1}$ = $b_{n-1}$) **then** /*operands have the same sign*/
    **begin**
      $s_{n-2}\ s_{n-3} \ldots s_0 = (a_{n-2}\ a_{n-3} \ldots a_0) + (b_{n-2}\ b_{n-3} \ldots b_0)$;
      $s_{n-1} = a_{n-1}$;
      **if** (EAC =1) **then** OVFL =1;
    **end**
  **else begin**
      $y_{n-2}\ y_{n-3} \ldots y_0 = \overline{a_{n-2}}\ \overline{a_{n-3}} \ldots \overline{a_0}$ ; /*complement A*/
      $x_{n-2}\ x_{n-3} \ldots x_0 = y_{n-2}\ y_{n-3} \ldots y_0 + b_{n-2}\ b_{n-3} \ldots b_0$ + EAC;
      **if (**EAC = 0) **then**
        **begin**
          $s_{n-2}\ s_{n-3} \ldots s_0 = \overline{x_{n-2}}\ \overline{x_{n-3}} \ldots \overline{x_0}$ ;
          $s_{n-1}=a_{n-1}$;
        **end**
      **else begin**
          $s_{n-2}\ s_{n-3} \ldots s_0 = x_{n-2}\ x_{n-3} \ldots x_0$;
          $s_{n-1}=b_{n-1}$;
        **end**
      **endif**
  **end**
  **endif**

**end**

**3.** Compare the 2's complement and sign-magnitude adders designed above. Estimate the area in terms of number of 1-bit full-adders and additional 2-input logic gates. For performance, use the timing analysis tool provided in the Xilinx Tools as follows: In the Process Window, go to Implement Design → Map Report → Timing. Double click on Timing to perform timing analysis on your circuit. A ( √ ) near that indicates successful completion of the analysis. Now double click on Timing Report in the submenu to pull

up the Timing Report. This report is a textual version of your circuits timing characteristics. It gives the timing paths from (all) inputs to (all) outputs if a path exists between them. Take time to analyze the report generated in the context of the design you completed and determine the worst case delay through your circuit. (You may request the TA for assistance in using the timing analysis tool).