**Laboratory Assignment #3 – All Sections**

**Due:** 24/11/2024, 21:59

**15-bit Adder-Subtractor with Overflow Detector Circuit using Verilog**

**Requirements:**

You are required to design a circuit that can both add and subtract two 15-bit integers and realize whether there is an overflow or not. You will design an 15-bit hybrid adder-subtractor circuit using carry lookaheads (e.g., you can use five 3-bit CLAs) and simulate behavioral model using **Icarus Verilog**.

Adder-subtractor circuits must detect overflow.

You are also required to use a hierarchical design method for these adder-subtractors, in which small building blocks are used to construct the entire circuitry. For example, to design a 15-bit carry-lookahead adder-subtractor. You will instantiate the 3-bit-CLA module to the 15-bit-CLA module (note that 15-bit-CLA is your top module). You may want to use the generate-propagate method to implement this module.

Since this assignment aims to help you to practice with Verilog language, you will perform both **bitwise operations** and **operations on buses** for computation. You are **not allowed** to use **direct arithmetic operations** (e.g. addition). For CLA, you **MUST precisely follow** the order below for the computations:

1. Compute the carry information using bitwise operations,
2. Compute any other desired value using arrays.

Note that there are three inputs (**`A, B, mode`**) and three outputs (**`S, OVF, Cout`**). This naming convention is already given to you so please utilize these modules in your implementation. You are required to design "CLA_3bit" and "CLA_15bit" modules using Verilog and you will test your implementation by a testbench, called "CLA_15bit_tb".

**Report Requirements**

1- Explain building blocks in your design.
2- Explain your module and submodule structure of your design.
3- Explain your testbench and give concrete examples for each testing option.
4- Include a screenshot for each testing option and show that it is working correctly.

The following summarizes what you should do before and during the lab session.

*Before the lab*

1. Work on your designs.
2. Design your circuits using Verilog.
3. Simulate your design using Icarus Verilog. For better visualization, you are required to group the inputs and represent them as signed decimal numbers.
4. Simulation inputs must contain one example for each of the following:
   a. addition without a carry and without an overflow,
   b. addition without a carry and with an overflow,
   c. addition with a carry and without an overflow,
   d. addition with a carry and with an overflow.
   e. subtraction without a carry and without an overflow,
   f. subtraction without a carry and with an overflow,
   g. subtraction with a carry and without an overflow,
   h. subtraction with a carry and with an overflow.
5. *Write a report* that explains the circuit you designed. Please clearly explain your hierarchical design and each small building block.
6. Submit your work, as much as you've done, through SUCourse as a zip file. Name of the zip file must be "{SUID}_lab3.zip" (e.g. If your SUID is 26797, your zip file must be "26797_lab3.zip").

*At the lab*

1. Simulate your designs for the input combinations given by your TA.
2. Demonstrate Step 5 for the TA.

**Notes**

- You will work individually. No group work allowed.
- You will be required to demonstrate your work to the lab assistant as (s)he instructs.
- No late submissions.
- **Do not change the module, input and output signal names.** Each (signal name, module name) change reduces your grade by five points.
- **Do not use any verilog features that you did not see in the lectures.** This will be seen as a source of plagiarism.