Laboratory Assignment #3

Due: 29/11/2021, 11:30 am

16-bit Adder-Subtractor with Overflow Detector Circuit

Requirements:

You are required to design a circuit that can both add and subtract two signed 16-bit integers and realize whether there is an overflow or not. You will design two different adder-subtractors and compare their performances in terms of area requirements. You need to design and implement your adder-subtractor circuits on Verilog, simulate the designs and verify their correctness using Vivado Xilinx 2018.2 tool. Finally, you need to write a report for your lab. You can find two different adder-subtractor circuits below (you can create two different projects for two adder-subtractor designs).

1. 16-bit ripple-carry adder-subtractor using full adders
2. 16-bit hybrid adder-subtractor using four 4-bit carry lookahead adders (CLAs)

All of them must detect overflow.

You are also required to use hierarchical design method for these adder-subtractors, in which small building blocks are used to construct the entire circuitry. For example, to design a 16-bit ripple-carry adder-subtractor, you first design a full-adder, then use it with three-bit input and two-bit output, and finally duplicate the full-adder as many times as needed (For hierarchical design, see Verilog-I document page 26-35 (under Lab Materials on SUCourse+)).

You are also required to report the performance (area) of your designs. After you finish your design (and verify its correctness), you must synthesize and implement (no constraint file needed) your design for FPGA board (Artix-7, xc7a100t, csg324, -1). Then, you should include your implementation results in your report in terms of area. The area will be expressed as amount of resources on the FPGA device (i.e., number of LUTs). For reading synthesize and implementation results, see the document Xilinx Vivado 2018.2 Design Tool Guide (Synthesis, Implementation & Programming FPGA) on SUCourse+. Finally, compare two designs by answering following questions in your report using implementation results:

1. Which one of the two is better in terms of area?
2. Which one of the two is better in terms of time?

You are also required to write a report which should include:

1. For each design, include the screenshots of your simulation waveform window showing that your design is calculating correct output (for at least 4 different input combinations). For example:

- Input A: 784, Input B: 53, Operation: Addition→Output C: 837, Output Overflow: 0
- Input A: -4, Input B: -70, Operation: Subtraction→Output C: 66, Output Overflow: 0
- ...

2. Include your implementation results and answer the questions above.

We are going to have in-lab sessions in the week of 29 November. During lab hours, you are going to simulate your designs for the input combinations given by your TA.

Notes:
In order to save time and avoid delays on demo schedule, please use the following input and output port names in your designs.
Inputs:
For the augend: A15, A14, A13, A12, ... , A2, A1, A0 (A15 is the MSB,A0 is the LSB)
For the addend: B15, B14, B13, B12, ... , B2, B1, B0 (B15 is the MSB,B0 is the LSB)
For addition/subtraction input: SUB
Outputs:
For the result: C15, C14, C13, C12, ... , C2, C1, C0 (C15 is the MSB,C0 is the LSB)
For the overflow: OVF