Cairo University
Faculty Of Engineering
Computer Engineering Department

CMP3020
Fall 2022

Mini Project#1

Adders Mania

Overview:

Adders are the building blocks of all “compute” units. Their performance impact the whole chip. This why in this mini project, we will explore different implementations of adders and study their charachteristics.

Requirement:

1. Design and implement using verilog the following **32-bits** **signed** integer adders
    - Verilog (‘+’) version of adders
    - Ripple Carry Adder
    - Carry Save Adder
    - Carry Look-Ahead Adder
    - Carry Increment adder
    - Carry Skip Adder
    - Carry Bypass Adder
    - Carry Select Adder
2. Implement a testbench to test the above adders: Covering 8 cases:
    - Overflow of positive numbers.
    - Overflow of negative numbers.
    - Addition of positive and negative number
    - Addition of positive and positive number
    - Addition of negative and negative number
    - Additional 3 random testcases.
    - Your testbench should print “TestCase#1: success” on success and should print the “TestCase#1: failed with input X and Y and Output Z and overflow status N”, elements in blue should be replaced by your values.
    - Your testbench should also report the total number of success and failure testcases at the end.
        i.
3. Synthesis the adders and add the following constraints
        i. Set (virtual) clock to 20ns.

ii. Set Input delay to 1ns.

iii. Set load to 10

iv. Set output load to 0.5ns.

v. Set Utilization to 60%

vi. Enable usage of all library cells.

- Report: Total Area, Max Delay, Max Slack, Min Slack, Total Power, clk.
- If a design suffers from -ve slack, adjust the timing constraints.
- Make sure that all designs work on the same constraints after modification.

4. Apply post-synthesis simulation using your previously made testbench.
5. Using the result you got from synthesis, use the most appropriate adder (from your point of view) to create a **32-bit floating point adder (IEEE-standard)**
6. Repeat the steps from 2-4 for the floating point adder.

Deliverables:

- Code files for the 9 adders (8 Integer + 1 float)
- Code file for testbenches.
- Excel Sheet containg the reported results in a tabular form.
- The Original reports of each design.
- A presentation containing the following
  - Your fullNames
  - Explanation of each adder design (including floating point).
  - Justification of your choice of adder used with floating point.
  - Additional challegnes you faced.
  - Roles of each team member and estimate time s/he worked.

Due Dates:

- All files should be zipped together and uploaded on GoogleClassroom by only one team member
- Due Date is **5th of November** midnight. Discussion will be on the 6th of November.

Final Remarks:

- Attached a helper file for adders design, feel free to ask google as well.
- Take care that implementations on the internet might be wrong, always use your mind.
- You will be asked to explain the adders, make sure you do understand what you implement.
- Don’t use clock in combinational circuits. (please)
- Don’t generate unnecessary latches. (please)