Project: Part 1

Adders Mania

Overview:

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

Requirement:

1. Design and implement using verilog the following **32-bit signed** integer adders
    - Verilog (‘+’) version of adders
    - Ripple Carry Adder
    - Carry Look-Ahead 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 test cases.
    - 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 report the total number of success test cases.
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 6 adders (5 Integer + 1 float)
- Code file for testbenches.
- Excel Sheet containing the reported results in a tabular form. For example

	Adder 1	Adder 2	Adder 3	…	…	…
Min Delay						
Max Delay						
Clock						
Total Power						
Area						
Utilization						
Number Latches/Flip Flops.						
…						

- The Original reports of each design.
- A presentation containing the following
    - Your fullNames and codes (9XXXXXX)
    - Explanation of each adder design (including floating point).
    - Justification of your choice of adder used with floating point.
    - Additional challenges you faced.
    - Roles of each team member and estimate time s/he worked.
- A video for each showcasing your understanding of the work done and results

Due Dates:

- All files should be zipped together and uploaded on GoogleClassroom by only one team member, videos can be provided as drive links only (make sure of open access).
- Due Date is **14th of December**.

Final Remarks:

- Teams will be ranked according to the smallest area (using **the same timing constraints** and that is the design is **full functioning**).
- feel free to ask google for ideas **NOT CODE** .
- **COPYING from the internet will be strongly penalized. (max of 10% of the code is allowed to be copied given you understand it).**
- Don’t use clocks in combinational circuits.
- Don’t generate unnecessary latches.