Faculty of Engineering and Technology
Electrical and Computer Engineering Department
Digital Systems ENCS2340

Verilog HDL Project

Project title: *Design and Implementation of a Simple ALU using Verilog HDL*
Due date: Tuesday, January 23rd, 2024
This project is to be done individually.

1) You need to submit your codes.
2) Write a **report** for your results by providing the code and the simulation results of every component as well as the whole system.

NOTE:

- The grading of the project will be via **discussion**.
- This project should be implemented in Verilog HDL using **Quartus software**

DO NOT:

- Give/receive code or proofs to/from other students

DO:

- Meet with other students to discuss the project (it is best not to take any notes during such meetings, and to re-work project on your own)
- Use online resources (e.g. Wikipedia) to understand the concepts needed to solve the project

Project Description:

In this project, you will design a simple Arithmetic Logic Unit (ALU) using Verilog Hardware Description Language (HDL). The ALU should be capable of performing four basic arithmetic and logic operations: addition, subtraction, bitwise AND, and bitwise OR.

Objective:

- To develop a comprehensive understanding of Verilog HDL for hardware description.
- To design and implement a simple ALU with four basic functionalities.
- To explore structural, dataflow, and behavioral modeling techniques in Verilog.

Tasks:

1. ALU Design:
    - Define inputs and outputs of the ALU.

    - Select four essential arithmetic and logical operations to implement (e.g., addition, subtraction, AND, OR).
    - Create a block diagram representing the ALU's structure.
2. Verilog Modules:
    - Develop a top-level ALU module with appropriate input/output ports.
    - Design separate modules for each of the four chosen operations, utilizing a combination of structural and behavioral modeling approaches.
        - Modules: `Adder`, `Subtractor`, `AndGate`, `OrGate`, and `ALU` (top-level module).
        - The ALU design should be modular, comprising separate modules for each functionality.
        - Implement the modules using structure, dataflow, and behavioral modeling as below:
            1. Utilize *structural modeling* for Adder, Subtractor, and logic gates.
            2. Implement *dataflow modeling* for connecting the modules.
            3. Utilize *behavioral modeling* for the top-level ALU module.
3. Input/Output:
    - The ALU should take two 4-bit inputs (A and B).
    - Include a 3-bit control input (`OpCode`) to select the operation:
        - 3'b000: Addition
        - 3'b001: Subtraction
        - 3'b010: Bitwise AND
        - 3'b011: Bitwise OR
    - Output the result (`Result`) of the selected operation.
4. Testing and Simulation:
    - Test various combinations of input values and control codes to ensure the ALU operates as expected. Apply a variety of input test cases to cover all possible combinations.
    - Simulate the ALU Components/Modules using a Verilog simulator and analyze the results
    - Simulate the ALU using a Verilog simulator and analyze the results.
5. Documentation:
    - Provide clear and concise comments within the Verilog code for better understanding.

    - Prepare a well-formatted project report detailing the design process, code implementation, simulation results, and conclusions.

Evaluation:

- Correctness of the ALU's functionality.
- Efficiency of the Verilog code.
- Quality of the simulation results.
- Clarity of documentation and project report.

Additional Guidelines:

- Adhere to Verilog coding conventions and best practices.
- Consider using a design hierarchy for better organization.
- Thoroughly test the ALU under various input scenarios.
- Document any assumptions or design choices made during the project.

Submission Requirements:

You need to submit the following files as one compressed folder (e.g., .zip, .rar) for your project by January, 23rd, 2024 (up to midnight). Then, name your folder as (hdlProject_LastName_FirstName_StudetnsID.zip).

- Verilog HDL source code for each module. Verilog code for the ALU and its modules.
- Simulation files. The simulation results (waveforms) of every individual component.
- The schematic files for the whole system (that shows the structural modeling of the final system design)
- Comprehensive project report (in PDF format), including design details, simulation results, and conclusions.
- The project discussion will be based on the above bullets, as well as, you will be asked to slightly modify the system. This way, we can assess your understanding of the project.

Best of luck with your ALU design!