Project: Building an Integer ALU

CS3339/CS5339 Fall 2023

Step-1 Due: Monday, February 19, 11:59 PM
Step-2 Due: Monday, March 25, 11:59 PM
Final Step Due: Monday, April 22, 11:59 PM

Objectives:

The goal of this exercise is to introduce the process and methodology of designing a new computer circuit from scratch. Using the latest tools the students will learn about hardware descriptive languages and how they are used to build circuits from the micron level. As a secondary goal, this project will also introduce several languages. The first, Verilog, is a hardware descriptive language that defines the mathematics of the circuit which is used to generate the circuit modules. It is open source and runs on Linux, with Ubuntu suggested.

The second language is LaTeX which is a document markup language. It is used by most journals as the method for publishing pdf documents. The only acceptable method for making submissions to any IEE or Springer sponsored conference and such is mandatory for any published technical researcher to use. It is a requirement reports for this project be made in LaTeX. The student will need to find an editor/compiler for LaTeX docs, there are plenty of options (TeXstudio with MiTeX for Windows machines is suggested). A LaTeX template will be provided to use with a sample document.

Process:

This is a group project where the students will work together, all submitting a single submission. The group will provide a name and its members during one of the initial classes. They will then self-coordinate to accomplish the following tasks on resources of their choosing.

Step-1: Verilog & LaTeX Basics
1. Install Verilog.
2. Code 1-bit Not, Nand, Nor, and 1x4-bit input / 1x4-bit output Shift Circuit.
3. Run Circuit Tests.
4. Generate Simulation Waveforms.
5. Prepare and submit LaTeX report.

Step-2: ALU functions
1. Code 4-bit binary logic functions (And, Nand, Or, Nor, Xor, Xnor, Not, 2x4-bit input / 2x4-bit output Shifter).
2. Code 4-bit arithmetic integer operations (Addition, Subtraction, Multiplication, Division: include carry in, carry out circuits, second 4-bit output for Multiplication and Division Remainder).
4. Run Circuit Tests.
5. Generate Simulation Waveforms
6. Prepare and submit LaTeX report.

Final: Putting it together
1. Integrate All ALU functions as modules from the previous exercise.
2. Code control circuit.
3. Run Circuit Tests.
4. Generate Simulation Waveforms.
6. Prepare and submit LaTeX report. Include new modules and test not shared in previous step report. Make sure you include a list of operation codes.

Extra Credit:
1. Provide a circuit diagram and logic table for each module in the report (+10 points).
2. Include larger ALU options: 8 bit (+5 points), 16 bit (+5), and 32 (+ 5) on step-2 and final.

Report Preparation:

Every step and the final should have a report done in LaTeX and submitted on the due date.

The report should include:
1. Header with Group Name, date, class, and group members.
2. Introduction on the project step and group goals.
3. Section with and explaining Verilog code including test scripts.
4. **Graduate Students**: Section displaying and explaining circuit diagrams (+10 points).
5. Section sharing test and waveform responses with some explanation.
6. Final conclusion sharing results, perspectives, difficulties, and successes.

Submission:

The submission shall be the PDF document exported from LaTeX, a folder named latex with the LaTeX source files, and a folder for the project files name Verilog inside a zip file with the with the filename GroupName-ProjectStep#.zip. Submit through Canvas the zip file by midnight of the due date. One submission per group.

Project Rubic:

Completeness		Formatting		Code Issues	
No PDF	-35	No Title	-5	Wrong Modules	-10
No Latex Files	-15	No Group Name	-5	Joined Modules	-10
No Code in PDF	-20	No Member Names	-5	Wrong Shift Circuit	-5
No Wave Forms in PDF	-20	Code not in Algorithm mode	-10	Code in Images (screen shots)	-5
No Test Code in PDF	-10				
		Test Issues		Extra Credit	
Narrative		Single Test File (non unit test)	-3	Circuit Diagrams	+10
No Narrative	-20	Not enough test options	-5	8-bit	+5
Missing Intro or Conclusion (each)	-5			16-bit	+5
Weak Intro or Conclusion (each)	-3	Result Issues		32 bit	+10
Report Flow	-10	Non-cropped/zoomed images	-5		
		Indecipherable Images	-10		