Lab 1: ALU Lab

EE312 Computer Architecture

Professor: Myoungsoo Jung

TA (Lab 1): Seungkwan Kang,Head TA: Miryeong Kwon

(EMAIL: kangsk@camelab.org, mkwon@camelab.org)

Deadline: 3/19/2022 (Sat) 23:59

**1. Overview**

*Lab 1* is a toy project that teaches you the basics of Verilog language. You are going to learn how to write code in Verilog and how to simulate it. In *Lab 1*, you are required to implement an Arithmetic Logic Unit (ALU) in Verilog. You are able to complete *Lab 1* based on what you have learned from the Verilog tutorial. Have fun!

**2. Files**

You are given two files:

1. *alu.v*: This is where you implement the ALU module.
2. *ALU\_TB.v*: This is a testbench file which you can use to test and grade your ALU module with.

You only need to change *alu.v*. You don’t need to change *ALU\_TB.v*.

**3. ALU Lab**

In *Lab 1*, you are required to make an ALU that has the following specifications:

1. Inputs and outputs are 16-bit signed binary numbers.
2. Operations are 4-bit binary numbers.
3. Overflow must be detected.

For Add & Sub operation, the ALU should be able to handle overflow. *Cout* must be 1 if overflow happens; otherwise, *Cout* should be 0.

The detailed explanations of the operations that your ALU module is required to handle are shown in Table 1.

|  |  |  |
| --- | --- | --- |
| **OP** | **operation** | **description** |
| 0000 | A + B | 16-bit *addition* |
| 0001 | A – B | 16-bit *subtraction* |
| 0010 | A and B | 16-bit *and* |
| 0011 | A or B | 16-bit *or* |
| 0100 | A nand B | 16-bit *nand* |
| 0101 | A nor B | 16-bit *nor* |
| 0110 | A xor B | 16-bit *xor* |
| 0111 | A xnor B | 16-bit *xnor* |
| 1000 | A | *Identity* |
| 1001 | ~A | 16-bit *not* |
| 1010 | A >> 1 | *Logical right shift* |
| 1011 | A >>> 1 | *Arithmetic right shift* |
| 1100 | A[0]A[15:1] | *Rotate right* |
| 1101 | A << 1 | *Logical left shift* |
| 1110 | A <<< 1 | *Arithmetic left shift* |
| 1111 | A[14:0]A[15] | *Rotate left* |

**Table 1. Operations**

**4. Grading**

The TAs will grade your lab assignments with *ALU\_TB.v*, which is already given to you. If you pass all the tests in *ALU\_TB.v*, you will get a full score. Your score is determined by how many tests you pass.

**5. Lab Report Guidance**

You are required to submit a lab report for every lab assignment. You can write your report either in Korean or English.

Your lab report **MUST** include the following sections:

1. Introduction
   1. *Introduction* includes what you think you are required to accomplish the lab assignment and a brief description of your design and implementation.
2. Design
   1. *Design* includes a high-level description of your design of the Verilog modules (e.g., the relationship between the modules).
   2. Figures are very helpful for the TAs to understand your Verilog code.
   3. We recommend you to include figures, as drawing them helps you *design* your modules.
3. Implementation
   1. *Implementation* includes a detail description of your implementation of the design.
   2. Writing about the overall structure and meaningful information is enough; you don’t have to explain all minor issues you have solved in detail.
   3. **Do not copy and paste your source code.**
4. Evaluation
   1. *Evaluation* includes how you have evaluated your implementation together with the simulation results.
   2. ***Evaluation* must include how many tests you pass in *ALU\_TB.v*.**
5. Discussion
   1. *Discussion* includes any problems that you have experienced while you follow through the lab assignment or any feedbacks for the TAs.
   2. Your feedbacks are very helpful for the TAs to further improve the course !
6. Conclusion
   1. *Conclusion* includes any concluding remarks of your work or what you have accomplished through the lab assignment.

**6. Requirements**

You **MUST** comply with the following rules:

* You must implement the lab assignment in **Verilog**.
* You shall only implement the **TODO** parts of the given template.
* You must name your lab report as **Lab1\_YourName1\_StudentID1\_YourName2\_StudentID2.pdf**.
* You must compress the lab report and source code, name the compressed zip file as **Lab1\_YourName1\_StudentID1\_YourName2\_StudentID2.zip**, and submit the zip file on KLMS.