CSED311: Lab 1
ALU

Sungjun Cho
allencho1222@postech.ac.kr

Index

Team Announcement

- Refer to PLMS

Learning Verilog HDL (1)

Throughout the labs of this course, you will be using Verilog HDL (Hardware Description Language) for assignments.

You will have to study Verilog. Here are some useful Youtube videos.

- Short and high-level introduction (<20 mins): link1, link2, link3
- More detailed tutorial (~50 mins): link

Learning Verilog HDL (2)

In case you are looking for lectures in Korean with more detailed explanations, look at this playlist: link

- See lectures 1-11 from the playlist. You can skip lecture 1-2 if you remember what you learned from the Digital System Design course

We also provide PPT slides (lab1_verilog.pdf).

Assignment (1)

Implement ALU (Arithmetic Logic Unit) in Verilog

Input: (A, B, FuncCode)

- A: left operand (16-bit signed binary)
- B: right operand (16-bit signed binary)
- FuncCode: operator (4-bit binary)

Output: (C, OverflowFlag)

- C: operation result (16-bit signed binary)
- OverflowFlag: overflow flag (1-bit binary)

Assignment (2)

ALU Specification

FuncCode	Operation	Comment
0000	A + B	Signed Addition
0001	A – B	Signed Subtraction
0010	A	Identity
0011	~A	Bitwise NOT
0100	A & B	Bitwise AND
0101	A | B	Bitwise OR
0110	~(A & B)	Bitwise NAND
0111	~(A | B)	Bitwise NOR

FuncCode	Operation	Comment
1000	$A \oplus B$	Bitwise XOR
1001	$\sim(A \oplus B)$	Bitwise XNOR
1010	A << 1	Logical Left Shift
1011	A >> 1	Logical Right Shift
1100	A <<< 1	Arithmetic Left Shift
1101	A >>> 1	Arithmetic Right Shift
1110	~A + 1	Two's Complement
1111	0	Zero

Assignment (3)

Overflow Detection

For **addition** and **subtraction**, you should detect overflow and set the flag.

Overflow Flag	Singed Addition	Signed Subtraction
0	Correct Result	Correct Result
1	**Wrong Result**	**Wrong Result**

For other operations, OverflowFlag is always zero.

Submission

You should submit **report** and **codes** to PLMS

Due date for codes: 2020/3/8 (Mon) 9:00

Due date for the report: 2020/3/8 (Mon) 24:00

The files for report and codes follow **these formats** for your assignment

"Lab1_TeamID_StudentID1[_StudentID2].pdf": PDF file for your report

ex) Lab1_20_20180001_20180002.pdf, Lab1_21_20180003.pdf (for one-man team)

"Lab1_TeamID_StudentID1[_StudentID2].zip": Zip file for your codes

Report (1)

Submit as **PDF file**

“Lab1_StudentID_Report.pdf”

You can write your report in **Korean** or **English**.

READABILITY is important!

Please help TAs to read it.

I encourage you to write a **simple** and **clear** report.

The **length of the report** is **not** related to the score.

Report (2)

Introduction

You should describe **following contents**

What you have to **design & implement**
What you have to **learn**

Report (3)

Design

You can save a lot of time with a **careful design**!

You should **make an effort** to write this section. For example,

How to divide a large module into submodules?

How does each submodule operate?

How to interconnect them?

If necessary, you should add a diagram to your report (handwritten diagram is also allowed)

Report (4)

Implementation

You should **explain your Verilog code**. For example,

The overall structure of your implementation

A short, but meaningful description for non-trivial modules.

The interaction between modules when they run a given scenario.

Do **NOT** explain too much details of your implementation.

Do **NOT** show the waveform results.

TAs will check your waveform results during the demonstration.

Report (5)

Discussion

You can write anything **valuable** that you want to inform.
For example:

Important decisions you made

Difficulties in designing and implementing, and your solutions for these

Differences between your design and implementation, and the reasons for these.

Feedback to TAs

Report (6)

Conclusion

You don’t need to repeat the contents of the introduction section.

You just need to answer the **following question**:

Did you succeed in achieving the goals described in the introduction section?

If not, which goals could not be achieved? Why?

Demonstration (1)

All you need **to do** is to

Show the TA that your implementation works

We will provide a testbench code for testing your implementation.

Answer some questions about your design and implementation

If you did your assignment well, you don’t have to worry!

Demonstration (2)

Because of **COVID-19**, we will use **MS Teams** for **online** demonstration

Procedure:

1. Remote-control TA's PC (using Teams) to download your source code from LMS
2. Demonstrate and explain your implementation to the TA
3. Answer questions from the TA

Demonstration (3)

Use the following link to **reserve a time slot** for your demo:

link will be uploaded on Teams

You should be connected to Teams before the appointed time.

There will be -10% point penalty if you do not show up on time, resulting in having to schedule another time slot for your demo.

Evaluation criteria (source code)

- Modelsim 작동 여부 (이외의 프로그램도 시뮬레이션 되면 인정)
- Testbench pass 수
- ALL PASS 여부
- 전체적인 기능 이해 – always, case, output C 계산과 관련된 부분
- Overflow detection 이해
- Modularization (add & subtraction, bitwise, shift, others)

Evaluation criteria (report)

- Introduction
- Design
- Implementation
- Discussion & conclusion
- Readability