CSED311: Lab 1
ALU

Jiwoong Shin
jwshin0610@postech.ac.kr

Index

Team Announcement (1)

Team	Member 1	Member 2
1	황규현	이재승
2	유승재	
3	김서연	권민아
4	박형규	성창환
5	이해찬	채승현
6	황예찬	김태훈
7	유태형	
8	정혜일	최검기
9	김호영	함형규
10	양희민	김인호
11	이제훈	
12	이승주	남진환
13	김창은	오민현
14	홍정민	유선주
15	김우주	조준형
16	신용원	
17	송문경	하재현
18	최영빈	강민호
19	최정원	도승욱
20	류누리	이신범
21	한상혁	김정헌
22	김은채	조은찬
23	권동현	신현수
24	최은수	
25	권택현	박세하
26	김근우	
27	이화윤	이현준
28	정연우	신지훈
29	이성환	
30	김어진	박정훈

Team Announcement (2)

Team	Member 1	Member 2
31	박준영	
32	박진우	하석윤
33	김재진	한승목
34	정세형	
35	박강희	
36	안성현	
37	이희우	
38	전유진	김가람
39	홍지범	이지은
40	김형진	
41	김현지	김선표
42	최진수	백성빈
43	장수혁	F.Marie Jerome
44	이정은	오정택
45	최수아	박수빈
46	김홍기	김민경
47	김민태	문영태
48	유상우	유태현
49	주봉조	박규민
50	최성재	강동훈
51	이건규	이상윤
52	안광진	김찬영
53	국현호	
54	이하은	

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 teach yourself 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.pptx).

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 **code** to LMS

Due date: 2020/3/30 (Mon) 9:00 a.m.

You should submit the file following **this format** for your assignment

A single zip file with following filename containing two files

"Lab1_StudentID.zip" ex) Lab1_20150301.zip

The two files follow **these formats**

"Lab1_StudentID_Report.pdf": PDF file for your report

"Lab1_StudentID_Code.zip": Zip file for your code (*.v)

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:

https://postechackr.sharepoint.com/:x:/s/CSED311ComputerArchitectureSpring2020/EY4W64ByDbRNl0PBcq_VV10BW2JL4HsxJVep3Wn_wLv6Yg?e=5GhIdv

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.