Computer Organization 2019

HOMEWORK 3 Floating Point Adder

Due date:

Overview

The goal of this homework is to help you understand the floating point addition circuit. You need to implement 32-bit floating point addition circuit using Verilog hardware description language (Verilog HDL). In addition, you also need to verify it by Modelsim.

General rules for deliverables

- You need to complete this homework INDIVIDUALLY. You can discuss the homework with other students, but you need to do the homework by yourself. You should not copy anything from someone else, and you should not distribute your homework to someone else. If you violate any of these rules, you will get NEGATIVE scores, or even fail this course directly
- When submitting your homework, compress all files into a single **zip** file, and upload the compressed file to Moodle.
  - Please follow the file hierarchy shown in Figure 1.

F740XXXXX (your id) (folder)
src (folder) * Store your source code
report.docx (project report. The report template is already included. Follow the template to complete the report.)

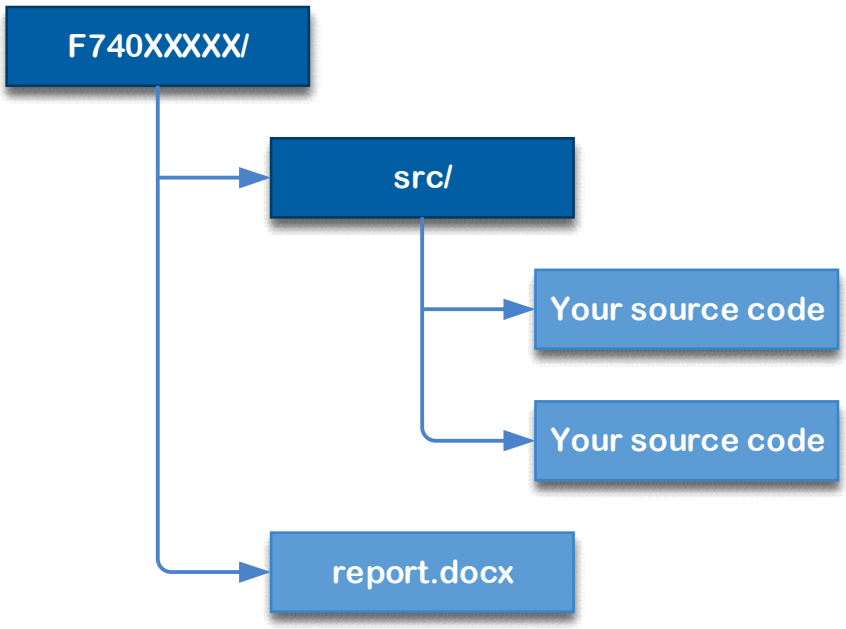

Figure 1. File hierarchy for homework submission

- **Important!** DO NOT submit your homework in the last minute. Late submission is not accepted.
- You should finish all the requirements (shown below) in this homework and Project report.
- If your code can not be recompiled by TA successfully using Modelsim, you will receive NO credit.
- Verilog and SystemVerilog generators aren't allowed in this course.

Circuit diagram:

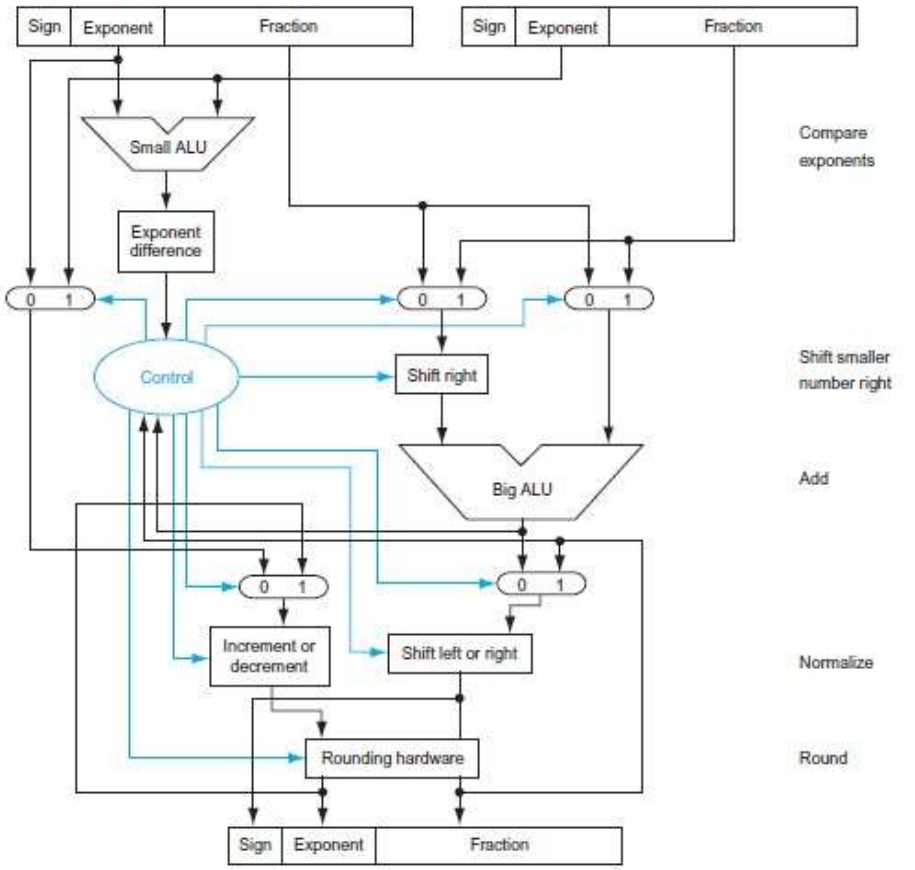

FIGURE 3.15 Block diagram of an arithmetic unit dedicated to floating-point addition. The steps of Figure 3.14 correspond to each block, from top to bottom. First, the exponent of one operand is subtracted from the other using the small ALU to determine which is larger and by how much. This difference controls the three multiplexors; from left to right, they select the larger exponent, the significand of the smaller number, and the significand of the larger number. The smaller significand is shifted right, and then the significands are added together using the big ALU. The normalization step then shifts the sum left or right and increments or decrements the exponent. Rounding then creates the final result, which may require normalizing again to produce the actual final result.

IEEE 754 32-bit floating point format

Signed	Exponent		Fraction	
31	30	23	22	0

Exponent	Fraction	Value
0	zero	±0
0	non-zero	$\pm 2^{-126} \times 0.(\text{Frac})_2$
1~254	Any	$\pm 2^{\text{Exp}-127} \times 1.(\text{Frac})_2$
255	zero	±∞
255	non-zero	special condition

※ The above 2 cases of exponent = 255 don't exist in the testbench, so you don't have to take them into consideration.

Homework Description

- **Module**
  a. **<u>fpadder_tb module: This is the testbench.</u>**
     ※You do not need to modify this module.
  b. **<u>fpadder module: Please refer to the above diagram and design a 32-bit floating point addition circuit.</u>**
     ※You need to modify this module.

Homework Requirements

1. Complete the 32-bit floating point addition circuit.
2. Verify your circuit with the testbench and take a snapshot (e.g. Figure 2)

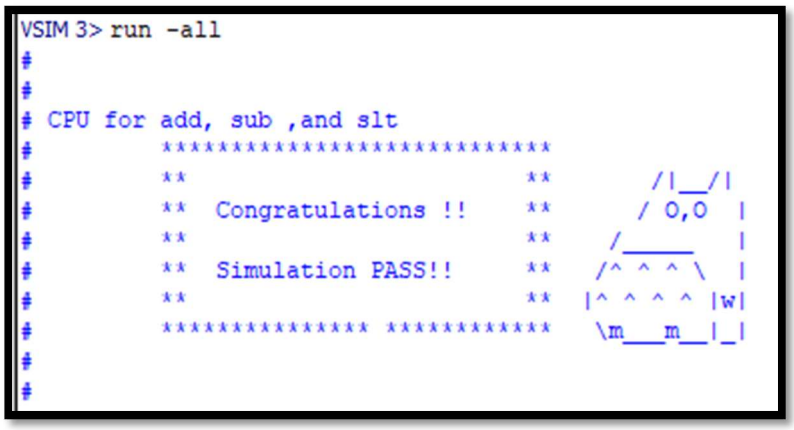

Figure 2. **S**napshot of correct simulation

a. Use **waveform** to verify the execute results.
b. Please annotate the waveform

3. Finish the Project Report.
   a. The report template is provided.

Important

When you upload your file, please make sure you have satisfied all the homework requirements, including the **File hierarchy**, **Requirement file** and **Report format**.

If you have any questions, please contact us.