COSE222: Computer Architecture
Design Lab #3

Due: May 26, 2021 (Wednesday) 11:59pm on Blackboard

Total score: 30

In the previous design lab you were requested to design the datapath elements: instruction memory, register file, ALU, and data memory. In this lab you are requested to complete the single-cycle processor design using the datapath elements. The single-cycle processor in this lab will support just 7 instructions listed in the textbook: `ld`, `sd`, `add`, `sub`, `or`, `and`, `beq`.

1. Single-cycle processor design

As you already designed the major datapath elements in the design lab #2, you can just instantiate your design blocks in the top design of the single-cycle processor. We also provide the complete designs for the instruction memory, register file, ALU, and data memory. If you failed to complete the design lab #2, you can use the provided design files. The required interfaces are also provided in the incomplete top design of the single-cycle processor. In the provided top design of the single-cycle processor, you are to include the following items.

The instruction memory already includes the following RISC-V assembly code. (Note that `lw` is used instead of `ld` in the code, but it is not critical for our single-cycle processor design.)

Assembly code	Binary data in imem
`start: lw x4, 0(x0)`	`00000000000000000010001000000011`
`lw x5, 8(x0)`	`00000000100000000010001010000011`
`lw x6, 16(x0)`	`00000001000000000010001100000011`
`T1: add x7, x5, x4`	`00000000010000101000001110110011`
`sub x8, x5, x4`	`01000000010000101000010000110011`
`and x9, x5, x4`	`00000000010000101111010010110011`
`or x10, x5, x4`	`00000000010000101110010100110011`
`beq x7, x8, T1`	`11111110100000111000100011100011`
`sw x7, 24(x0)`	`00000000011100000010110000100011`
`lw x11, 24(x0)`	`00000001100000000010010110000011`
`beq x7, x11, T1`	`11111110101100111000001011100011`

(1) Simple datapath elements such as MUXes, a shifter, and an immediate generator

(2) The main control unit as described in the textbook

(3) The ALU control unit as described in the textbook

(4) Connecting datapath elements and control signals

You can find the more detailed instructions in the incomplete top design file (single_cycle_cpu.sv) and the testbench file (tb_single_cycle_cpu.sv). Please complete the design and the testbench file. Verify your design using ModelSim.

2. What to do

(a) Complete the top design of the single-cycle processor (single_cycle_cpu.sv).

(b) Verify your design with testbenches. The incomplete testbench file is provided. Set the clock frequency as 100 MHz and instantiate the top design file as "dut" (design under test). (Please find the provided testbench "**tb_single_cycle_cpu.sv**").

(d) Create the ModelSim project "Lab3" in your workspace folder. Copy all design files and *.mem files to the project folder. The mem files include the initial memory data for the instruction memory and the data memory. Verify your design using ModelSim. Run the testbench for 240 ns, then you will find the generated "report.txt" file, which will print the contents of the register file and the data memory.

(c) Compress your completed all the design files (imem.sv, regfile.sv, alu.sv, dmem.sv, and single_cycle_cpu.sv), the testbench (tb_single_cycle_cpu.sv), and the report file (report.txt) in **one zip file**. You must name your zip file as "`FirstName_LastName.zip`". (e.g. `Gildong_Hong.zip` for Gildong Hong) If your submission file does not meet this rule, we will reduce 1 point from your score. ☹