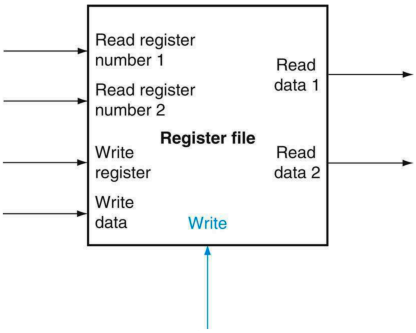
Name: _____

Lab for ITSC 3181, Introduction to Computer Architecture, Spring 2023

Lab #09: Logic design using Digital (https://github.com/hneemann/Digital): A register file with 32 32/16/8-bit registers. Due 03/29 and count for 1% of the final accumulated grade.

From Lab 07 to Lab 12, you will build a RISC-V CPU to support an essential set of instructions including at least add, and, or, sub, lw, sw, and beq instructions. While you are encouraged to design a 32-bit CPU (each register stores a 32-bit word, and the ALU operates on 32-bit data), you are ok to design 8-bit or 16-bit CPU (each register stores a 8-/16-bit data, and the ALU operates on 8/16 bits data). Designing 8-bit/16-bit is much easier comparing with designing 32-bit, but with same design principle. But regardless of the bitwidth you choose, each instruction is still encoded as a 32-bit word and the address for accessing data from memory are also 32-bit address. In this lab, you will design a register file with 32 registers.

1. Review the lecture slides (123 -131 of https://passlab.github.io/ITSC3181/notes/AppendixA_LogicDesignBasics.pdf) for how to design a register file with 32 32-bit registers. The input and output of the register file should be exactly as the design in the slide:



Tasks:

1. (10%) **Create register X0:** Create a new circuit named as *32-<bitwidthOfYourChoice>RegisterFile* in Digital and add a D-Flip-flop in the design, set this register to have the Data Bits of your choice (8, 16 or 32) for the CPU bitwidth, and Label it as X0. This will be the register X0.
2. (10%) **Create and arrange the other 31 registers for the register file:** Make a copy of this X0 register using Ctrl-C or Cmd-C depending on whether you are using Windows OS or Mac OS X, and then paste it 31 times using Ctrl-V or Cmd-V. This will easily create the other 31 registers that have the bitwidth set for you and labelled correctly from X1 to X31. Arrange the registers properly in one column and make sure leave some space between them for wires.
3. (20%) **Design the first read ports using a 32-1 32/16/8-bitwidth Mux according to slide 121:** You can use the Mux you designed before or use the one provided in Digital. Add and label the input (5-bitwidth) as “Read register number 1” and output (32/16/8-bitwidth) as “Read data 1”.
4. (20%) **Design the second read ports using 32-1 32/16/8-bit Mux according to slide 121:** You can use the Mux you designed before or use the one provided in Digital. Add and label the input (5-bitwidth) as “Read register number 2” and output (32/16/8-bit) as “Read data 2”.
5. (30%) **Design the write ports using the 5-32 decoder according to slide 124**: You can use the decoder you designed before or the one provided by Digital. Add and label Inputs (32/16/8-bit) as “Write data”, 5-bitwidth as “Write register” and 1-bit as “Write” enable signal.
6. (10%) Test your design: Save the design of your register file, and then simulate and test your design.

Important: when designing the circuit, please make sure you follow these rules for adding, changing components, input/output and wires. These rules are applied to all the design lab tasks of the course.

1) **Each input and output of a design MUST be properly and meaningfully labeled.**
2) **Each component, input and output should be correctly configured in terms of its bitwidth and signal control width.**
3) **Keep wires and components organized and layed out according to the circuit schematics rules, a) Inputs on the left (or top), b) Outputs on right (or bottom), c) gates flow from left to right, d) Straight (not angled) wires are best, e) Wires always connect at a T junction (only 90-degree turn or connection), f) A dot where wires cross indicates a connection between the wires, and g) Wires crossing *without* a dot make no connection. While you may not need to follow all those rules for creating correct and small circuit, it is very important when for creating complex circuit. So, make sure you properly organize the components and wires, make them structured and look good. That will help reduce errors.**
4) **For drawing straight wires using mouse, make one turn per each draw. You should not use one draw to make a connection that needs to have two or more 90-degree turns, which would create angled wires. For those wires, you have to make a wire that has a 90-degree turn, and then connect it with another wire that also make 90 turn, and so on. Try to minimize the turns as much as possible. If two wires have to be crossed, but should not be connected, make sure no dot is marked on where the wire is crossed.**

Submission: take screen shoot of your design of each task and save it to a single file of PDF format. Then submit the file on Canvas.
Your grade will be based on both the correctness of your design and the organization of the design, 60% and 40% respectively.

Task	1 (10%)	2 (10%)	3 (20%)	4 (20%)	5 (30%)	6 (10%)	Total
Correctness (components, input/output, bitwidth, connection, width of control, etc), 60%							
Organization (I/O labeled and positioned correctly, all straight wires and T junction, dot used correctly, readability etc), 40%							
Total							