

Assigned: Tuesday 21 September

Due: Wednesday 30 September 10:30 pm

Collaboration For this assignment, you may work with one or two partners from the class section. That is, you may work alone or in teams of up to three students. Teams of two are preferable. **Solutions submitted by teams of four or more students or with students from the other section will receive an automatic zero.**

This assignment should be a learning experience for everyone in the team, so do not “divide and conquer.” You should use a pair programming approach as you build your circuits.

Work submitted by your team must be your team’s own work. You may discuss (with words) your design with other groups, but you must not share or show your circuit files for any of these exercises.

Getting Started The starter file includes the inputs and outputs you need; do not change them. Copy it from `~weinman/courses/CSC211/assignments/memory/registers.circ`

Problem 1: A 4-bit register

In the subcircuit called “4-bit register”, build a four-bit register constructed of four D flip-flops. (Use Logisim’s built-in D flip-flop components, which may be found in the explorer pane’s **Memory** category.) The current four bits stored in the register should be displayed in **output**. The register has the following inputs:

data the four bits of data to store

clock the clock signal

write_enable a control line indicating data should be stored in the register

Writing to the register should be controlled by the `clock` signal with `write_enable` connected to the flip-flop’s “Enable” pin `en`. The data should be stored in the register only on the falling clock edge—this is *not* the default and must be changed in the attribute panel—when the `write_enable` selector is set.

Problem 2: A 4x4 register file

In the subcircuit “4x4 register”, build a register file consisting of four, 4-bit registers. The register file has additional inputs

read register number (two bits) to display on output

write register number (two bits) to store data into when `write_enable` is set

Use Logisim’s built-in Decoder and Multiplexor components and *your* 4-bit register as a component (don’t copy/paste).

What to turn in

- Completed Logisim circuit file `memory.circ`.
- Each subcircuit from the file—clearly labeled—as an image in a single PDF

I will use scripts to run tests on your `memory.circ` file and print your PDF to write comments. Failure to include either will result in significant grade penalties. Each group should submit only one copy on PioneerWeb. Every group member’s name and box number should be in the PDF.

Acknowledgments This derivative work of Janet Davis, used under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License was developed in collaboration with Charlie Curtsinger.