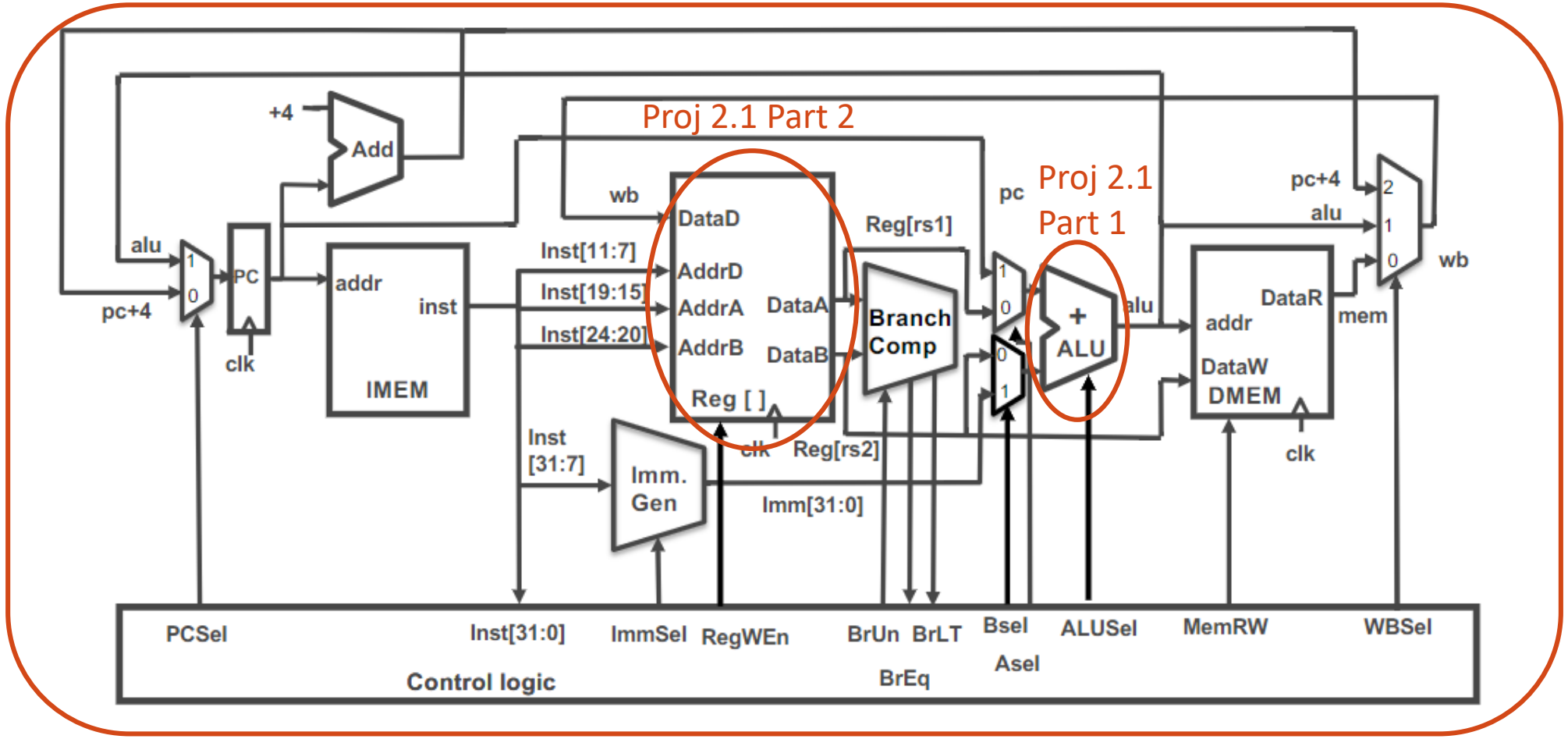
Discussion 10: Project 2

MENGYING WU

Project 2 Overview

Proj 2.2

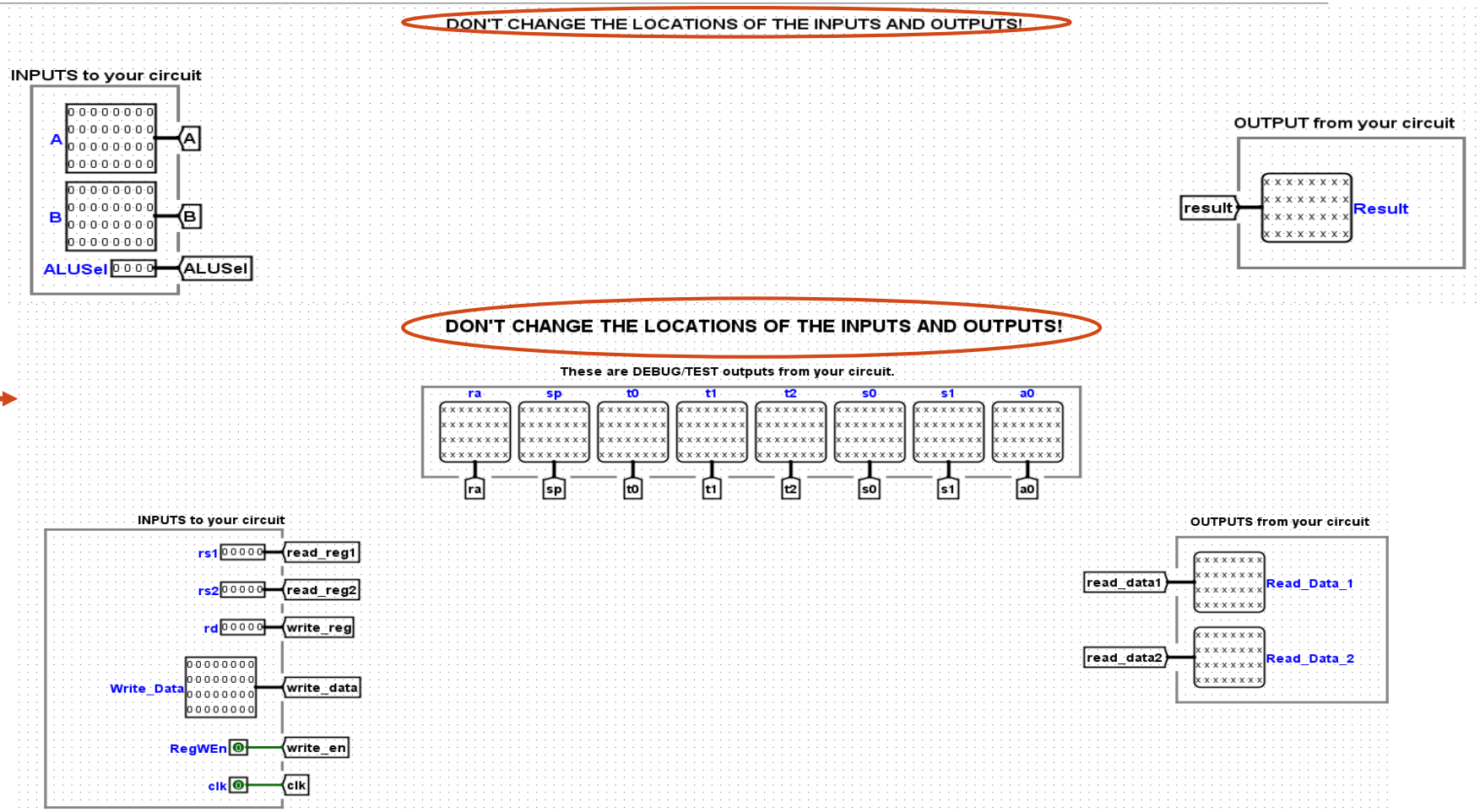


DON’T CHANGE THE LOCATIONS OF THE INPUTS AND OUTPUTS!!!!!!

Project 2.1: ALU and RegFile

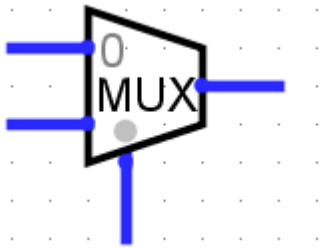
- Files you obtain:
  - tests
  - alu.circ
  - alu_harness.circ
  - autolab.txt
  - logisim-evolution.jar
  - README.md
  - regfile.circ
  - regfile_harness.circ
  - test.sh → Run tests

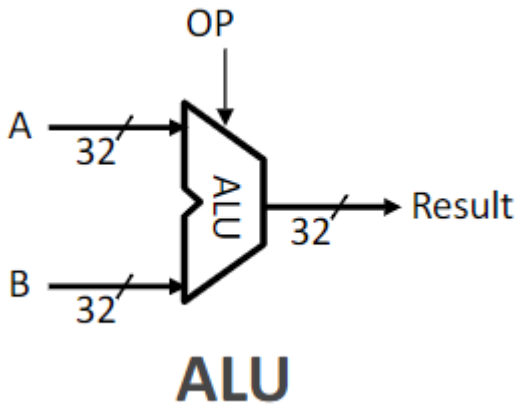
You can create your own tests in these two circuit.

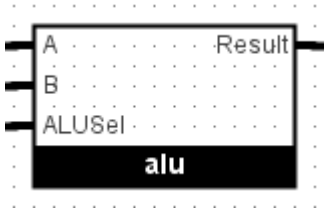


Project 2.1: ALU

- Input: two 32-bit numbers (A,B) and 4-bit ALUSel
- Output: one 32-bit number
- ALUSel decides what to output (A+B, A-B, or other operations)
- Just like a multiplexer
  - You should be familiar with it in Lab5

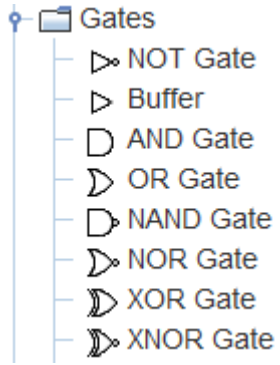


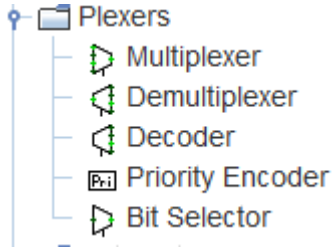


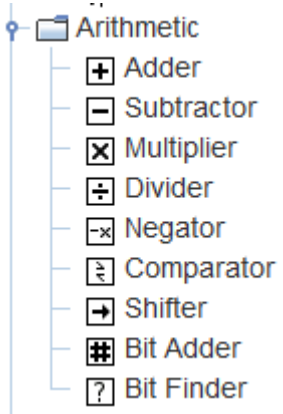


Project 2.1: ALU

- Your ALU should be able to execute 16 operations as required
  - Some operations may not be used in CPU
  - Some operations may not be included but used in CPU
  - Some operations may not be required but exist in real CPU
  - So you can modify your ALU in later implementation
- You can make good use of Gates, Plexers and Arithmetic to implement.



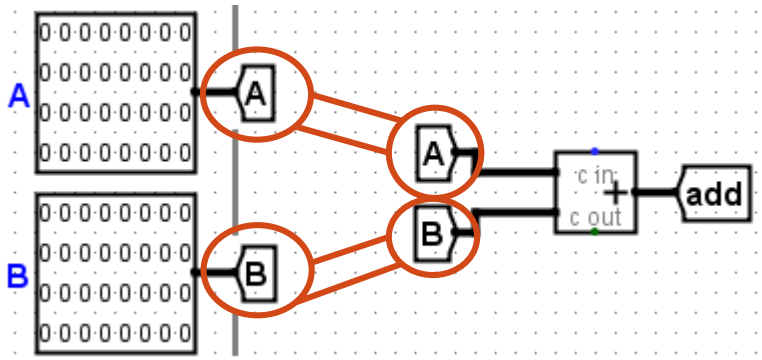


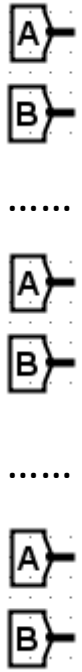


Of course you can build them by hand if you don't know what to use :)

Project 2.1: ALU

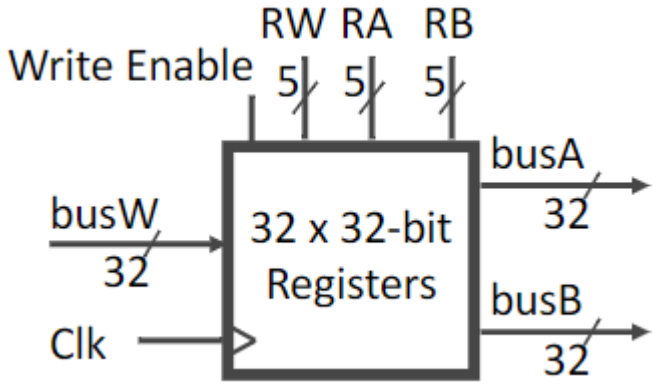
- Tunnel can transfer pins from one place to another without lines
  - Use it to tidy your circuit
  - Recommend if you are OCD

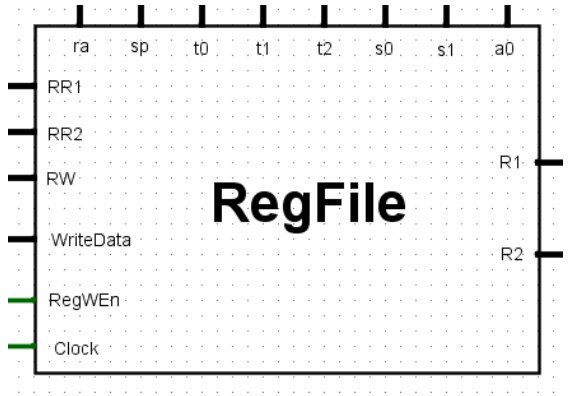




Project 2.1: RegFile

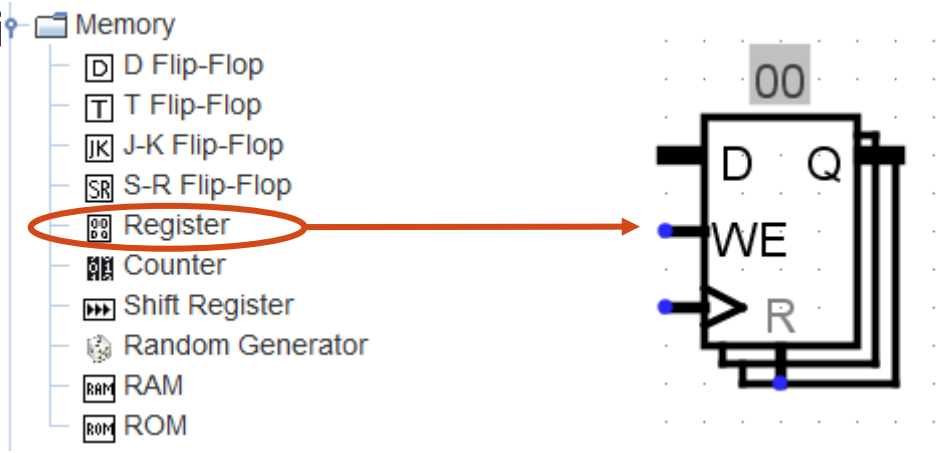
- Our RegFile only has 8 registers.
  - Both in Project 2.1 and Project 2.2
  - But a real CPU has 32 registers, don't make mistakes in exam.
- Input
  - rs1(5-bit), rs2(5-bit), rd(5-bit)
  - WriteData(32-bit)
  - RegWEn(1-bit)
    - Determines whether data is written to the register file on the next rising edge of the clock
  - Clock(1-bit)
    - It is a given clock, which means you shouldn't connect a clock here.
    - Please take a look at `regfile_harness.circ` to see how the clock is used.
- Output
  - Two regsiters' values, chosen by `rs1` and `rs2`





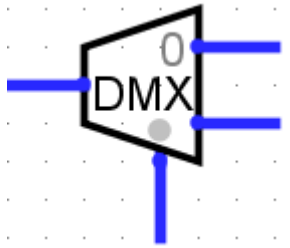
Project 2.1: RegFile

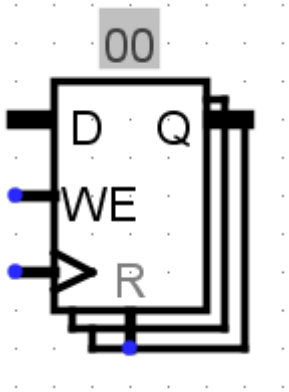
- Only to choose a register and simply output its value then pass RegFile tests?
- No!
- ALU is a Combination Logic, just calculate the current numbers then output.
- RegFile is a Sequential Logic, the most important thing is to maintain the data in your registers.
  - Otherwise you will have all registers equals 0 all the time!
- You can find regi



Project 2.1: RegFile

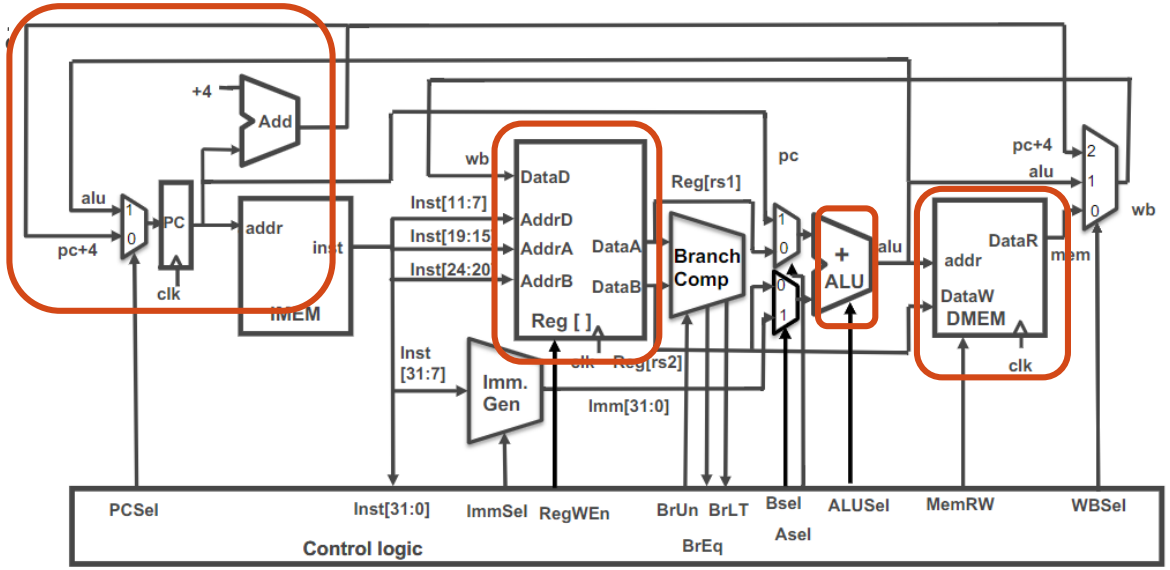
- Demultiplexer
  - A inverse multiplexer
  - Choose one way to output the given number
- You may find it useful when you choose a register to write data according to `rd`, `WriteData` and `RegWEn`
- Think of the 5 interfaces of this register, what should them be connected to?





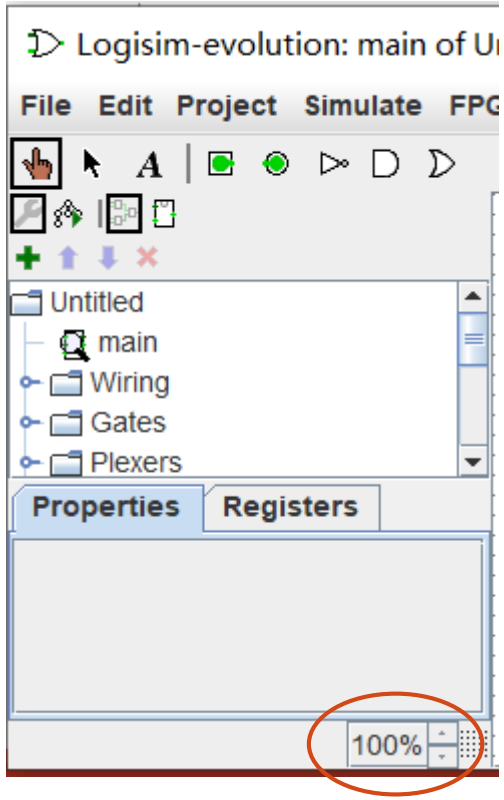
Project 2.2: CPU

- Files: `cpu.circ, mem.circ, run.circ`
- Target: given an address and an instruction, output the updated registers' status.
- `Cpu.circ`
  - You are given PC, DMEM, RegFile and ALU circuits.
  - Remember to move your `regfile.circ` and
  - Main task:
    - Implement a Control Logic, analysis the instruction
    - Connect the lines in this graph
    - And other small logics you need
- `Mem.circ`: DMEM
- `Run.circ`: tests in IMEM

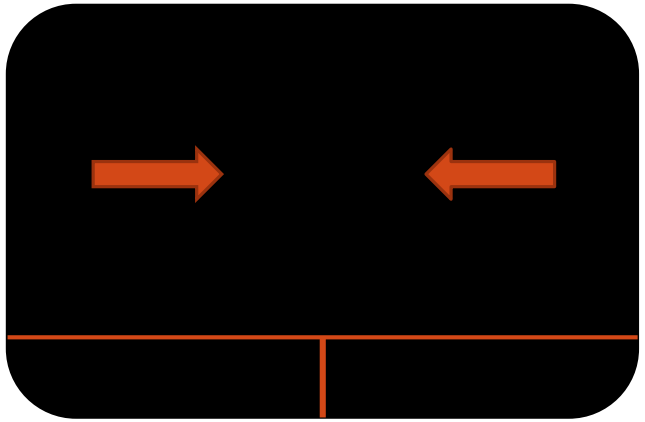


Tip on Logisim

- Zoom



But if you use Logisim on laptop…



Project 2

- Project 2.1 will be published tomorrow.
- You have two weeks to implement it.
- Project 2.2 will be published after it.