Homework 9: Simulated Memory Bank

Your task is to create a functional 16-byte memory bank in Logisim. Each byte will be able to be read and written independently of the others. The final file should be called `memory.circ`. Here are the inputs:

- A 4-bit-wide address on the north side, which indicates the active byte.
- An 8-bit-wide data-in on the west side, where data to be written is received.
- A 1-bit write enable on the west side, that allows the data-in to be recorded to the memory bank when it is on. (If it is off, the data input is ignored.)

It has a single output:

- An 8-bit-wide data-out on the east side, which outputs the data stored in the active address.

As before, you should start by downloading the pre-made circuit. The "dummy-memory" is simply a wrapper for Logisim's built-in memory. Your final main circuit should be identical to the one provided, except with the dummy memory replaced by your memory.

As with the adder, the best way to do this task is to first design a sub-circuit. In this case, you should design a D-latch. The D-latch will take an 8-bit data-in and a 1-bit write-enable, and output an 8-bit data-out. The memory bank will then hold 16 of these D-latches, along with multiplexers and/or decoders to direct the data and the write-enable to the correct place. (You are free to use built-in multiplexers and decoders if you like.)

Also remember that this assignment requires feedback loops. As such, you will likely have bugs that shut down the simulation. You can reset it using commands in the *Simulate* menu.

Extra Credit: Many practical memory banks have two address inputs and two data outputs, allowing two addresses to be read simultaneously. Implement this. Data will only be written to the byte indicated by the frst address; the second is read-only. You will need to modify the main circuit if you implement this.