CS 382
Fall 2023

Computer Architecture
and Organization

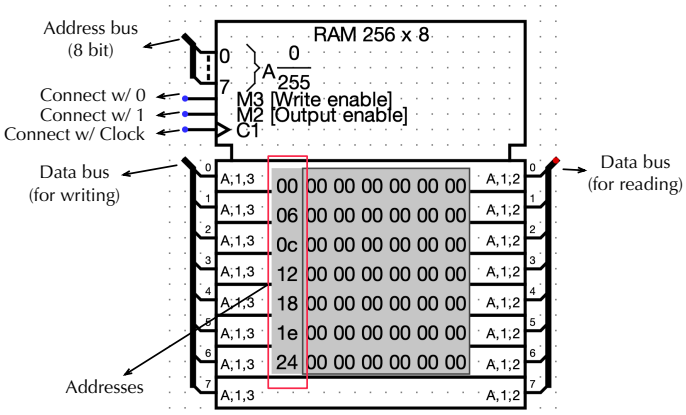
Lab 8

1 Task: Create a RAM

In this lab we will focus on random access memory (RAM) in Logisim-Evolution, and more specifically, we will create an instruction memory with program counter (PC).

On the left panel, choose `Memory`, and you will see both RAM and ROM, and please choose **RAM**. A typical RAM is shown as in the picture on the right. Note if your RAM looks different, go to attribute panel, and choose "Logisim-Evolution" for Appearance.

By default, this RAM can store 256 bytes, so it has 256 unique addresses, and therefore the address bus is 8 bit wide (why?). The address bus is at the left top corner with number "0" and "7". For port `M3`, please connect it with a constant `0`, and for `M2` with a constant `1`. Also, please connect `C1` with a clock.



The left is the data bus for write, and right for read. In this lab, you only need to connect the read port with an 8-bit wide output pin to show the content we read from the RAM, while leaving write port unattached.

1.1 Loading Content to RAM

Once you're done with the steps above, you can start simulation, and you'll see the RAM has data in it (in the grey box). There you see on the left these are addresses in hexadecimal, and each row has six bytes. If you're in the simulation mode, you can click on every byte, and type the data you want to store there. However, if you reset the simulation, all the data will be gone. A good idea is to store the memory data into a text file, called **image file**. To do this, right click the memory, and choose "Save Image", and store it somewhere. Next time when you want to load them into the RAM, just right click and choose "Load Image".

In this lab, the image file you submitted can contain any data, but they cannot be all zeros. Instead, you are encouraged to store a wide range of values, to make sure your RAM works successfully.

1.2 Automatic Reader

In this task, we will want to read every byte at a constant rate from RAM. Therefore, we need a register that connects to the address bus of the RAM and can update itself. Let's call that PC (get it? 😊).

Consider for each clock cycle, we need to read from the memory, and also update PC, so RAM and PC should be controlled by the same clock. Also, because we want to advance the memory address by one byte, we need to connect the PC with an adder, and add constant 1 as the other input of the adder.

In sum, you need the following components:

- Clock;

- Adder;
- Register;
- Output pin;
- RAM;
- Constants.

All of them can be found in the left panel of Logisim-Evolution.

2 Grading

The lab will be graded based on a total of 10 points.

- **-5:** the image file is missing or cannot be loaded into the RAM;
- **-5:** the circuit is missing or completely broken;
- **-5:** PC does not update itself;
- **-5:** not every byte is read;
- **-2:** used ROM instead of RAM;
- **-1:** any other minor glitch that can be fixed quickly;
- **-1:** no pledge and/or name in `.circ` file.

Earlybird Extra Credit: 2% of extra credit will be given if the lab is finished by Wednesday 11:59PM EST (1 day before the lab deadline). For specific policy, see syllabus.

Attendance: check off at the end of the lab to get attendance credit.

Deliverable

One `.circ` file, and one image file.