CS-382 Computer Architecture and Organization Fall 2022

Lab 7 · Mini Register File

Lecturer: Shudong Hao *Date: See Canvas*

1 Task 1: Mini Register File

1.1 Configuration

In this lab, we are going to use Logisim-Evolution to build a mini register file. The configuration is as follows:

- The register file contains **four** registers, one write port, and two read ports;
- Each register can store one byte;
- All registers are synchronized by a clock.

Based on Figure 3.14 from the textbook (pp.65), we need the following data signals:

- Input:
  - `WriteReg`: the register ID for writing data;
  - `RegWrite`: the switch to control if we need to write to a register;
  - `RegDataW`: the data we want to write to the register;
  - `ReadReg1`: the register ID for reading (register 1);
  - `ReadReg2`: the register ID for reading (register 2);
- Output:
  - `RegData1`: data read from register 1;
  - `RegData2`: data read from register 2.

1.2 Write-Enabled Registers

In Logisim-Evolution, we can use write-enabled registers (left panel → Memory → Register), shown below.

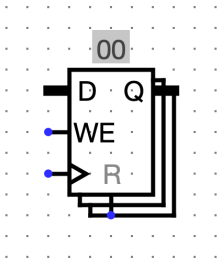

Figure 1

The grey box with number shows the data stored in the register; `D` is the input for data (write), and `Q` is the output for data (read). We can leave `R` unattached. The port with a small triangle is used to connect a clock. For this part you'd need to go to left panel → Wiring → Clock, choose the clock, and connect it with the register triangle.

The `WE` port means write-enabled. This is actually the "Clock" or the "C" signal we used in the textbook Figure 3.14, and should connect to an AND gate with input `RegWrite` and the output from a decoder. In this lab, you can certainly use the decoder provided by Logisim-Evolution: left panel → Plexers → Decoder.

2 Task 2: Simulation and Timing Diagram

Before we start exploring timing diagram, make sure you have labeled all the input and output signals, as well as register numbers (such as `X1`, `X2`, etc).

Once you have built the timing diagram, we can start simulating our register file. To set up a comfortable environment for simulation, do the following:

- Change the cursor to the finger (not arrow) to get ready for simulation;
- Go to the top menu, choose Simulate, and make sure the item "Auto-Tick Enabled" is chosen;
- Under same menu, choose "Auto-Tick Frequency", and choose 0.5Hz so it's not running too fast;
- Under same menu, choose "Timing Diagram" to open a new window, and click on Timing Diagram tab;
- Click the play button.

Then you'll see the clock starts ticking!

What you need to do from here: enable `RegWrite`, and change input signals as much as you

like as the clock ticks. Then take a screenshot of the timing diagram. Look closely to the timing, especially each clock cycle and how the output responds to your input change.

3 Requirements

- You must write your name and pledge in the `.circ` file that contains the register file;
- You must label all the input and output pins, the register numbers, **and the clock**;
- You must use the decorder, multiplexor, and the register components from Logisim-Evolution;
- Write a PDF file that contains the screenshot, and discuss what you notice from the timing diagram. The discussion needs to be around how output changes corresponding to input change, especially its relation to clock cycles. *Hint:* after you change the input `RegDataW`, can you read it right away from the output? When can you read it then if not?

Deliverable

One `.circ` files and one PDF file, zipped.