RAM
COMP 273 Assignment 5
Due: March 1, 2024, at 23:55 on myCourses

Submission instructions

All work must be your own and must be submitted to myCourses. Include your name and student number in a comment at the top of your Logisim Evolution circuit. **Submit only one file: a5.circ**. Check your submission by downloading it from myCourses and verifying that it was correctly submitted. You will not receive marks for work that is incorrectly submitted.

Purpose

- Learning about the basic circuitry in RAM.
- To get used to using flip-flops.
- To get used to using addresses.
- To get used to splitting and merging signals.
- To learn that we need to respect the direction of a signal.

Helpful

- Tutorial E should be completed before you start this assignment.
- Notes: 14 – RAM and Bus.pdf

Overview

This assignment asks you to use Logisim Evolution to create an 8-nibble RAM. To make this an easier assignment, we are using nibbles (4 bits) instead of bytes (8 bits). Your RAM only has 8 nibbles, this means that your address register only needs to be 3 bits long. To simplify this assignment further, you will implement two data registers, one for reading and the other for writing (see figure 1). Optionally, you can merge these two data registers as in real RAM implementations.

Your goal is to create a RAM circuit that is able to do two things: (1) by placing write in the mode register (1 bit long), placing a value in the data register (4 bits long), placing an address in the address register, and then turning the clock on the value in the data register is copied into the nibble of the RAM at the address you selected. When you turn the clock off, the value stays in the RAM nibble. (2) Placing read in the mode register, then placing an address in the address register, and then turn on the clock. The data register should now contain a value from the nibble pointed to by the address register.

This should work for every nibble of RAM. The TA will "run" your circuit with multiple reads and writes. Make sure your circuit is robust enough to handle multiple **reads**, **write**, and **overwrites**.

Implementation

Using Logisim Evolution create the circuit following this diagram:

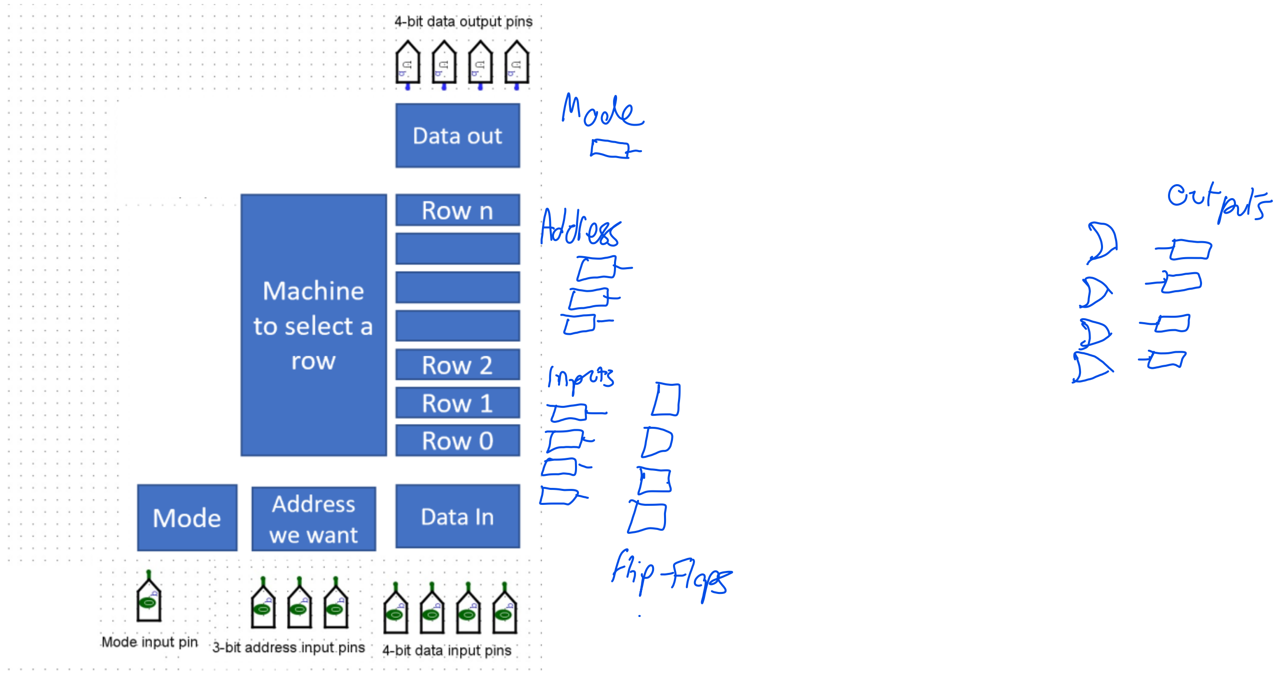

Figure 1: High-level view of RAM

- Task: Build the RAM of figure 1.
- Inputs: Mode, Address, Data In
- Outputs: Data Out
- Memory: 8 nibbles of RAM, and row selector circuit, data-in/data-out unidirectional buses

NOTE: Your final circuit does not need to look exactly like figure 1. However, the input and output pins must look like figure 1.

NOTE 2: Your final circuit must use designs we covered during class. You cannot use any outside (other sourced) circuit designs.

Note 3: You do not need to build your own flip-flops. You can use the flip-flops provided in Logisim Evolution. **You must build your memory from flip-flops**. For example, you cannot use Logisim Evolution's Register object and RAM object.

Steps

- Begin by placing the input and output pins onto the circuit. These pins will be used to enter and see values.
- Mode, Address, Data In, and Data Out are registers. Create these registers from scratch using flip-flops. Do not use the Logisim Register pre-built unit. You can pick any flip-flop you want. Group and synchronize the flip-flops for one register as a single unit. As seen in class.
- Row 0 to Row 7 are the eight nibbles of RAM. Also create them from scratch using any flip-flop you want. Do not use the Logisim Byte pre-built unit. Group and synchronize them into units.

- Build the row selector circuit connecting the Address and Mode registers with the Nibbles.
- Use the row selector and Mode register with the Data-In and Data-out registers (proper read/write gating circuitry).
- Add a clock to control the execution of this circuit.

Execution

Your RAM circuit must be able to do the following:

1. TA will input an Address and a Data-In value and will place 0 in Mode for write to nibble.
2. TA will turn the clock on to save the information into the nibble.
3. TA will turn the clock off.
4. TA will input an Address and a 1 in Mode to read a nibble to Data Out.
5. TA will turn the clock on to read the information from the nibble into Data Out.
6. The TA will turn the clock off.
7. The TA will then be able to see the Data-Out pins displaying the expected value.
8. The TA should be able to repeat this process as often as they want. Data previously stored in the nibbles should still be present unless they are overwritten.

Marking

- Maximum 20 points
    - +1 Mode register
    - +2 Address register
    - +1 Data In register
    - +1 Data Out register
    - +3 Row selector circuit
    - +3 8 nibbles of RAM
    - +3 Saves data to nibble at address register
    - +3 Reads data from nibble at address register
    - +3 Repeatable infinitely
- -10% per day late.
- -3 points for not following instructions (removed proportionally).
- -5 points for not using the clock (removed proportionally).
- Assignment must execute to be graded.