MAT 3670: Extra Lab
Digital Logic Structures: Part III

Background

This lab will give you an opportunity to explore how elementary storage devices can be constructed with digital logic components.

Pre-Lab Exercises

You should be familiar with the following ideas before starting the lab: the RS latch, the gated D latch, and the construction of a 4×3 memory module. These are discussed in the textbook and the relevant circuits also appear in this handout.

One of the lab exercises asks you to design and implement a circuit which can store 16 three-bit words. Time spent thinking about your design away from the computer is, as always, an excellent plan of attack.

Laboratory Exercises

1. Create a **labExtra** folder to store the circuits you will need for this lab.

2. Obtain the circuits for this lab from the course web site, placing them in your **labExtra** directory.

3. Figure 1 shows a block diagram view of an RS-latch. From within Logisim, open **rs-latch.circ**. Initially, set each of the R and S inputs to 1. By toggling these lines (from 1 to 0 and then back to 1), convince yourself that this circuit has the capability to set and reset one bit of storage.

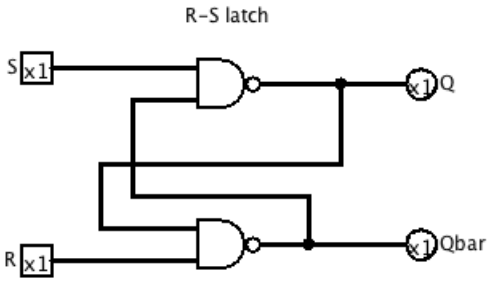


Figure 1: An RS latch constructed with NAND gates. The quiescent state is $R = S = 1$.

4. Figure 2 shows a block diagram for a gated D latch; a possible implementation of a gated D latch is shown in Figure 3. From within Logisim, open **gated-d-latch.circ**. Experiment with this circuit and convince yourself that one bit of data can be stored in this device.

5. Figure 4 shows the block diagram for a memory consisting of four words of three bits each; a possible implementation of this is shown in Figure 5.

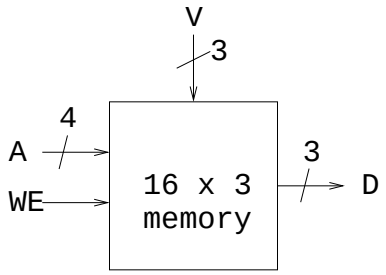
   Open **mem4x3.circ** and experiment with this circuit, convincing yourself that you can both read and write three-bit data words. Fill the memory with the following data values, then cycle through all addresses to verify these values can be retrieved.

Address	Value
00	000
01	010
10	100
11	110

6. Modify the 4×3 memory circuit by adding an `OE` (output enable) input pin. When `OE` is asserted, the circuit behaves as before; when `OE` is de-asserted, each of the three `D` lines should be 0. Put the `OE` input on the south side of the circuit.

   After making your modification, test the circuit by loading the memory with specific values, then cycle through all the addresses as before.

7. OK; here's your chance to design your own memory circuit, as promised earlier. What we want is a storage device with four times the capacity of the 4×3 storage module — sixteen words of three bits each. Here's a block diagram of the desired circuit:



   Notice the similarity of this to our earlier memory module — now, we simply have more address bits. With four address bits, there are $2^4 = 16$ addresses.

   Ultimately, your circuit will have $16 \times 3 = 48$ gated D latches. However, this is not the best way to think of it. Instead, you should use four of the 4×3 memory modules, which you can think of as modules M_0, M_1, M_2, and M_3. Of the four address bits, the upper two, A_3 and A_2 can be used to select the appropriate module. Within that module, use the lower two address bits, A_1 and A_0 to refer to the correct word. A two-input decoder can be used to select the appropriate module.

   Other than the 4×3 memory modules, your design should use basic gates (AND, NAND, OR, and NOT) only — since we are looking for a fundamental understanding of how such circuits can be constructed.

   You should aim to obtain a design with a pleasing layout on the page.

   Before you submit your work, thoroughly test your design, following a strategy similar to what was done for the 4×3 module. In particular, you should place a variety of three-bit values in the sixteen words of memory, then cycle through all addresses to verify the words of memory can be retrieved.

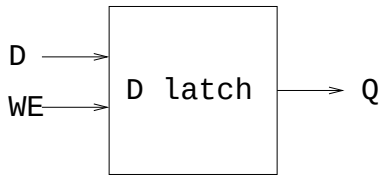


Figure 2: A block diagram of a gated D-latch.

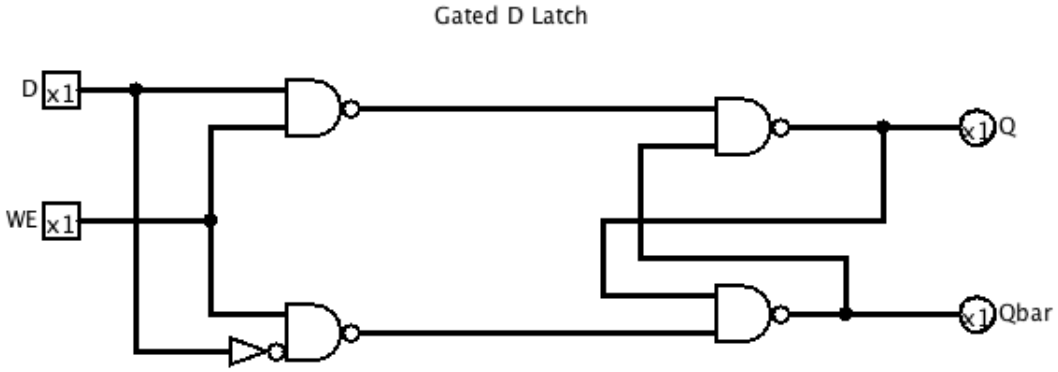


Figure 3: A gated latch constructed with NAND gates. The WE (write enable) control line allows the data value D to be captured and stored.

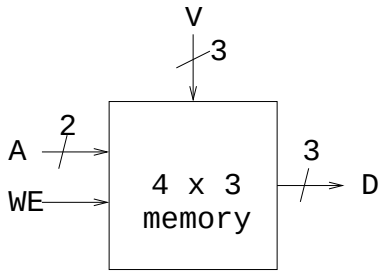


Figure 4: Block diagram of a memory with four words of three bits each. Two bits of addressing information, `A`, can be used to access the four words. To read a value from a given word, the appropriate address bits are specified and the data appears on the data lines, `D`. To store a value, the address and desired value are specified on the `A` and `V` lines; asserting `WE` causes the value to be stored.

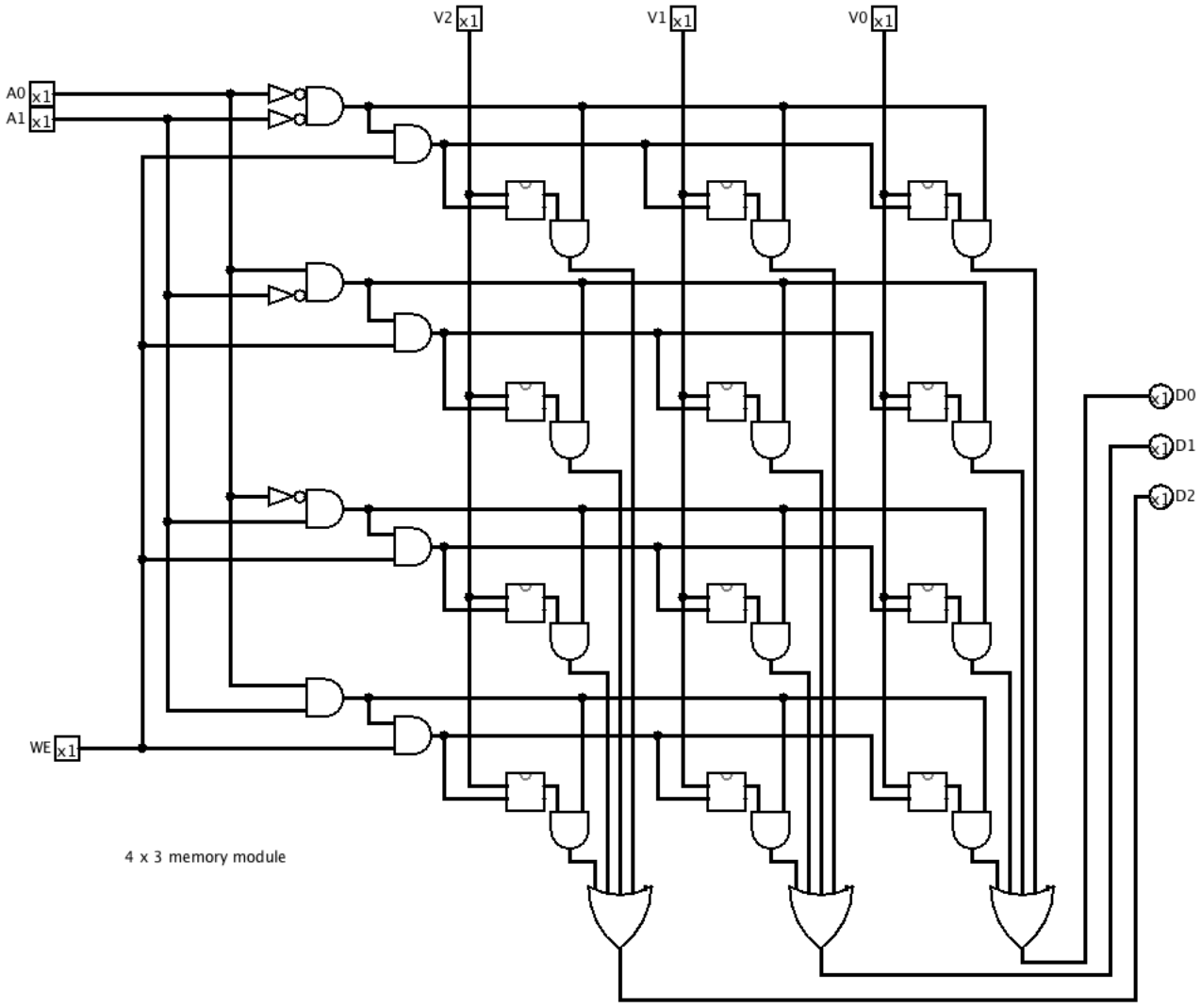


Figure 5: A small memory, organized as four rows of three columns. Each bit of the memory module is implemented with one gated D latch.

Submissions

When you have completed the lab, submit your `labExtra` folder by dragging it onto the EIU submit icon.