|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 🔑 **Essential** 🔑 | | | |  | 🏆 **Enhanced** 🏆 | | | |
|  |  |  |  |  |  |  |  |  |
| \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ |  | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ |

**Score: \_\_\_\_\_**

**Lab05 – Dynamic Memory (DRAM)**

COMP256 – Computing Abstractions

Dickinson College

Spring 2022

Prof. Grant Braught

**Name(s):**

**Introduction:**

In the last lab we learned about an abstract memory cell, like the one shown below:



Recall, that this cell behaves as follows. When E=1 the cell is *write enabled* and the value of D is written into the cell and will be reflected on Q. When the value E=0 the value stored in the cell is still reflected on Q, but changes to the input D do not affect Q. Thus, to write a value into the cell we write enable the cell (i.e. set E=1) and then set D equal to the value to be stored. We then set E=0 to preserve the value of D in the cell.

In Lab 04 you built a *sequential logic circuit* called a *D-Latch* that implements the behavior of the above abstract memory cell. The D-Latch is an example of what is known as *static memory* or *SRAM*. When a value is written into a bit of SRAM the value of that bit will remain the same as long as the circuit has power (i.e. the value is static or unchanging).

In this lab we will consider a different way to implement the behavior of the abstract memory cell that is called *dynamic memory* or *DRAM*. Here dynamic means that while we will be able to store the value of a bit into the cell, it will not remain static. Instead, the properties of the circuit will cause it to change even while the circuit has power (i.e. the value is dynamic or changing). In particular, when a 1 is stored in the cell it will slowly decay over time until it is no longer a 1 and eventually until it becomes a 0. While this may seem like a bad design, we will see that DRAM has another advantage that makes it a good complement to SRAM in designing the memory hierarchy.

**SRAM vs DRAM:**

In class we saw that SRAM and DRAM are used for different parts of the memory hierarchy. This was a result of the cost/performance tradeoff between these two types of memory. Refer back to the class slides for the day on Memory Hierarchy and Caching to answer the following questions.

🔑 1. Which of the two types of memory (SRAM or DRAM) is faster?

🔑 2. Which of the two types of memory (SRAM or DRAM) has higher cost?

🔑 3. What is the name of the part of the memory hierarchy that is constructed using SRAM? What part is constructed using DRAM?

🔑 4. Of the two parts of the memory hierarchy that you identified in question 3, which has greater capacity (i.e. can store more information)?

**A DRAM Memory Cell:**

A DRAM memory cell will have the inputs D and E and the output Q, and its behavior will be the same as the abstract memory cell and the same as the SRAM implementation we saw in Lab 04. However, the circuit for a DRAM cell is quite different as shown below.



This circuit contains just one transistor, one resistor and one capacitor. Transistor used here is an NPN transistor instead of an NMOS transistor. The electrical characteristics of the NPN transistor make it more suitable for DRAM cells than the NMOS transistor we used before. The connections on an NPN transistor are labeled Collector (C), Base (B) and Emitter (E) instead of the Source (S), Gate (G) and Drain (D). But other than that, at our level of abstraction, the NPN transistor will behave just like an NMOS. For our purposes we can think of the capacitor as a small rechargeable battery. It is this capacitor that will actually store the bit in our DRAM memory cell. When the capacitor is charged it stores a 1 and when it is discharged (i.e. dead) it stores a 0.

Notice that one side of the capacitor is connected to ground. The other side (the non-ground side) will be used to *charge* (store a 1) or *discharge* (store a 0) the capacitor. To store a 1 in the capacitor we attach its non-ground side to 1 (i.e. a positive voltage) by setting D=1 and closing the “switch” in the transistor by setting E=1. To store a 0 in the capacitor we attach its non-ground side to 0 (i.e. ground) by setting D=0 and E=1. When the non-ground side of the capacitor is not attached to either a 0 or a 1 (i.e. the switch is open because E=0), it will remember what value has been stored – like a battery holding a charge.

🔑 5. The figures below show a sequence (left to right) of different inputs to our DRAM circuit. For example, figure (i) shows that when the write enable E=1 the NPN transistor will act like a closed switch. This connects the non-ground side of the capacitor to input D=0. This causes the capacitor to discharge, and thus stores a 0. Because the output is connected to the non-ground side of the capacitor, it will always reflect the value that is stored in the capacitor, so 0 in this case. Figure (ii) shows that if the write enable E is changed from 1 to 0, the “switch” in the NPN transistor will open. Thus, the capacitor is disconnected and will remember that it had the value 0 stored into it. Complete the remaining figures (iii-vi) showing the switch position, the value stored in the capacitor and the value of the output.



🔑 6. Compare your results in question #5 to the figures i-vi in Lab 04 that show the behavior of the abstract memory cell. They should agree. If not, revisit your answer to #5. Be sure to figure out and understand any mistakes that you had made, rather than simply copying the expected results from Lab 04. *No answer is required here.*

🔑 7. Create a new circuit in TINKERCAD named Lab05-DRAMbit and setup a small breadboard as shown below. Be sure to use an NPN transistor instead of an NMOS transistor. Also**, be sure to click on the capacitor and set its capacitance to 5µF as shown**. You can think of the capacitance as a measure of how long it takes the capacitor (i.e. small rechargeable battery) to go dead.



🔑 8. Connect the switches, transistor and capacitor to create a DRAM cell as was shown in the schematic earlier in the lab. If you look closely in TINKERCAD, you’ll see the pins of the NPN transistor are labeled as C, B and E for Collector, Base and Emitter. Be sure those are connected as shown in the schematic. Also connect a voltmeter to the output (Q) of the DRAM cell so that you can see the value that is saved in the cell. *Your TINKERCAD circuit is your answer for this question.*

🔑 9. Now let’s store a 1 into the DRAM cell. When the simulation starts you should have E=0, D=0 and the output Q=0. Start the simulation and, complete the following steps, as described earlier, to write a 1 into the cell.

1. Set D=1 as the value to be stored.

2. Write enable the cell by changing E=1

What is the voltage at Q when you write enable the cell? Is that a logical 0 or logical 1?

🔑 10. Now disable the write to the cell by setting E=0.

a. Watch the voltage at Q for a little while. What happens to this voltage?

b. What will happen to the logical value stored in the cell if we wait long enough?

What you are seeing is the “dynamic” part of dynamic RAM (DRAM). Like a battery, over time the capacitor slowly loses its charge. In our model, we have a pretty big capacitor, so this happens relatively slowly. The capacitors used in real DRAM chips are very very small and thus this erosion of 1’s to 0’s happens much more quickly. To compensate for this, DRAM modules contain additional circuitry that *refreshes* the memory about every 60ms (or about 16 times per second). Each refresh rewrites all of the bits that are 1’s, bringing their voltage back up to the maximum to ensure that they remain 1’s and do not decay to 0’s.

🔑 11. Now let’s check that the 1 is stored in the DRAM cell. Start by *refreshing* the 1 in your DRAM cell. Set D=1 and then cycle E from a 1 (write enabled) and back to 0 (write disabled). That should recharge the capacitor and *top up* the voltage in the cell so it is a 1 again. Then with the write disabled (E=0), toggle the input D from 0 to 1 and back to 0 several times. Is the behavior of the output Q affected by these changes in D while E=0 (i.e. when write is disabled)?

🔑 12. We have now seen that we can store a 1 into the DRAM cell. We need to confirm that we can also store a 0. First refresh the 1 in the cell as we did earlier. Then write a 0 into the cell. Give the sequence of steps that you used to write the 0 into the cell.

Note that once the 0 is written into the cell, and writing is disabled (E=0), then changes to D should not affect the output Q, similar to what we saw in question #11 for when the cell was storing a 1.

**A DRAM Memory:**

DRAM memories are usually characterized by a number of rows and columns. For example, a 2Mx16 memory would have 220 rows (the M is Mega) and 16 columns of cells with each cell holding one bit. Each row of the DRAM has an address. So, for example, address 1510 would hold the 16 bits in the 16th row of the module. Note: That the address of the first row is 010, so address 110 is the 2nd row and address 1510 will be in the 16th row. This is just like the memory in the K&S and the way that arrays work in Java.

You’ll be building a very small 2x2 DRAM memory as shown below:



This memory has just two address, Address 0 (yellow) and Address 1 (green), and each address holds just two bits. For example, Address 0 holds bits bit0,1 and bit0,0. The subscripts 0,1 and 0,0 indicate the row and column of the bits. So, for example, if Address 0 contains the value 210 in unsigned binary (i.e. 102) then bit0,1 will be 1 and bit0,0 would be 0.

🔑 13. Which bits store the value that is saved at Address 1? What would the values of those bits be if the Address 1 contains the value 310 in unsigned binary?

🏆 14. Imagine we were going to build a DRAM memory module for the main memory of the Knob & Switch computer. How many rows and columns would there be in that module?

**Storing Values in the DRAM:**

The way this DRAM memory works is as follows. The bits to be stored into the memory are placed on the data lines (D1 and D0). So, to store the unsigned number 210 (102) we would set D1 = 1 and D0 = 0. Then we write enable the Address into which we want to store the value. So, to store this value into Address 1 we would bring E1 to 1 and then back to 0.

🔑 15. Give the steps you would need to use to store the value 110 into Address 0 in unsigned binary.

🔑 16. Create a new circuit in TINKERCAD named Lab05-DRAM2x2 and setup three small breadboard as shown below. Note that I have placed labels and color coded this diagram to correlate with the abstract example just above. **Be sure to use NPN transistors and to set the capacitors to each be 5µF.**



🔑 17. Wire this circuit to operate as a 2x2 DRAM memory as described earlier. Connect each of the voltmeters to the output of a cell so that you can see the stored values. **Note that each of the transistors has a resistor attached to its Base (B). The write enable line for the bit must be connected to this resistor and not directly to the Base,** as we did earlier when making the 1 bit DRAM memory cell**.** *Your TINKERCAD circuit is your answer for this question.*

🏆 18. Store the value 102 in Address 0 and the value 112 in Address 1. Then take a screen shot and paste it below as your answer to this question. You’ll have to be quick to set the values and get the screenshot before the voltages decay too much. If you find that the values in the memory degrade to quickly as you work with the circuit, increasing the values of the capacitors will slow that down for you (e.g. try 10µF or 20µF).

**Comparing SRAM and DRAM:**

In discussions of the memory hierarchy we have identified SRAM faster but being more expensive than DRAM and DRAM as being slower but less expensive than SRAM. This leads to designs that balance the cost/performance tradeoff by using small amounts of SRAM for cache and registers and large amounts of DRAM for main memory. Let’s compare the cost and speed of these technologies.

*Cost*

🏆 19. In Lab 04 you built a D-Latch, which is an SRAM that can store one bit of information.

a. Which logic gates (e.g. AND, OR etc) did the D-Latch circuit use and how many of each type of gate did you use?

b. Use your favorite search engine to determine how many CMOS transistors are needed to build each of the types of logic gates you identified in part a.

c. If we count each transistor as having a cost of 5 units, what is the total cost of building 1 bit of SRAM memory (i.e. a single D-Latch)?

🏆 20. In this lab you built a DRAM cell that can store one bit of information. If we assume that creating a capacitor or a resistor on an integrated circuit each cost 1 unit, and a NPN transistor costs 5 units what is the total cost of building 1 bit DRAM memory?

🏆 21. When building memories there are two types of costs to consider. The financial costs and the amount of chip space that is required. Designs that have higher financial cost and/or required more chip space cost more and thus we will tend to use less of them. Designs that have lower financial cost and/or require less chip space cost lest and thus we can have more of them. Use the information from questions #19 and #20 to answer the following questions.

a. Which costs more to build a DRAM cell or an SRAM cell? Briefly explain your answer.

b. Which would take up more space on a chip, a DRAM cell or an SRAM cell? Briefly explain your answer.

*Speed:*

Thinking back to the Lab 04, the SRAM circuits were just made up of logic gates, which are made up of transistors. Thus, the speed of an SRAM cell will be determined by how fast the transistors can switch on and off and the speed at which the electricity flows around the circuit.

🏆 22. From class we know that SRAM is faster but more expensive than DRAM. The previous few questions looked at why SRAM is more expensive than DRAM. Now, think about the DRAM cell and what you have observed about its behavior in this lab. Why might reading and writing data to DRAM be slower than reading and writing data to SRAM?

**Final Thoughts:**

Between Lab 04 and this lab you have gained some insight into how data is stored in computers and why there is a cost/performance tradeoff between SRAM and DRAM. Before that we saw how to build a 1-bit “programmable” computer using the basic logic gates (Lab 03). And before that we saw how logic gates can be built from transistors and how electrical signals represent 0’s and 1’s (Lab 01, Lab 02). In class we saw how those 1’s and 0’s can represent numbers, letter and machine instructions and how those instructions can be fetched, decoded and executed. So really, at this point you’ve seen it all. The rest is just details. Of course, there are a lot of those details, but you should have a good solid intuition about how computer hardware performs computations using just electricity and transistors!

Optional: To help me improve and scope these activities for future semesters please consider providing the following feedback.

a. Approximately how much time did you spend on this Lab (include the 2-hour lab period in your total)?

b. Please comment on any particular challenges you faced in completing this Lab.

**Acknowledgements:**

Box images from Open Clip Art and used under a CC0 license.

