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

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

**02 – Transistors to Logic Gates**

**Activities**

COMP256 – Computing Abstractions

Dickinson College

Spring 2022

Prof. Grant Braught

**Name:**

**Introduction:**

In today’s activities you will begin to build up through the computing hardware abstraction hierarchy. In class, I introduced the ideas of electrical signals representing 0’s and 1’s, transistors as electrically controlled switches, NMOS and PMOS transistor behavior and the basics of circuits. In the activities below you will expand and build upon this foundation to more fully develop an understanding of how CMOS transistors are combined into circuits that can perform logical operations. You’ll also have the opportunity to explore a little of the history of computing. In the next several classes and labs we will then see how the logical operations performed by circuits (i.e. logic gates) can be combined to produce circuits that do useful computations and also serve as computer memories.

**A CMOS Circuit:**

🔑 1. Today’s class explained the operation of NMOS and PMOS transistors and looked at the operation of the CMOS circuit shown in the box below. Use the drawing tools in Word (or your favorite drawing program) to show:

* The value of the input set to 0.
* The positions of the imaginary “switches” in the P and N transistors.
* The value of the output that would be produced.



🔑 2. The operation of circuits like the one above can be summarized concisely using a *truth table*. A truth table is a table that lists all possible input(s) to a circuit on left and the corresponding output(s) on the right. For example, the left column in the table in the box below shows the possible inputs to the circuit from question 1.

You can complete the truth table, applying the input in each row to the circuit, determine the output the circuit will generate and place it in the corresponding row of the output column. Use this process to fill in the Output column of the truth table below so that it represents the behavior of the circuit from question 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
|  | | **Input** | **Output** |  |
|  | | 0 |  |  |
|  | | 1 |  |  |
|  |  | |  |  |

**Logic Values:**

If you examine the truth table from question #2 you should see that an input of 0 is converted to an output of 1, while an input of 1 is converted to 0 (If not, you should revisit questions #1 and #2). Instead of 1’s and 0’s we sometimes will want to think in terms of the *logic values* of TRUE and FALSE. By convention, computer scientists usually will use 0 to represent FALSE and 1 to represent TRUE.

🔑 3. Translate the truth table from question #2 so that it shows logic values instead of 1’s and 0’s.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
|  | | **Input** | **Output** |  |
|  | |  |  |  |
|  | |  |  |  |
|  |  | |  |  |

🏆 4. In Java, and most other programming languages as well, the NOT (!) operator performs the same function as this circuit. To see the similarity, complete the table below by filling in two possible values for X, one that makes X>3 TRUE and one that makes it FALSE. Then apply the NOT operator to fill in the logic value in the final column.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | |  |  |  |
|  | | **X** | **X>3** | **!(X>3)** |  |
|  | |  | TRUE |  |  |
|  | |  | FALSE |  |  |
|  |  | |  |  |  |

**Logic Inverters and Not Gates**

From your truth table in question #3 you should see that the CMOS circuit we have been studying changes a TRUE input to a FALSE output and a FALSE input to a TRUE output. Because the circuit changes TRUEs to FALSEs and vice versa it is called a *Logic Inverter*, or simply an *Inverter*.

As we will see in the coming classes, an inverter is a common element in circuits that perform computations. In these circuits we will usually only be concerned with the inverter’s behavior (i.e. it inverts (i.e. flips) logic values) and we will forget about the particular configuration of transistors and their imaginary switches opening and closing. Because we only care about the behavior of the circuit and not its details, we can use abstraction.

5. A *NOT gate* is an abstraction for the particular configuration of transistors that make up a Logic Inverter. Use the information on the Inverter (logic gate) Wikipedia page to answer the following questions:

* <https://en.wikipedia.org/wiki/Inverter_(logic_gate)>

🔑 a. Compare the truth table for the NOT gate to the one you produced for our CMOS inverter in question #3. Does the NOT gate do the same thing as the CMOS converter? How can you tell?

🔑 b. What is the traditional symbol used to represent a NOT gate? Draw it here.

🏆 c. This page also shows a variety of different ways that a logic inverter can be created using transistors. They use a conventional symbol to represent NMOS and PMOS transistors that is different than the circles we have been using. What symbol do they use to represent an NMOS transistor? A PMOS transistor?

🏆 6. We now know that the circuit from question #1 performs logical inversion or computes the NOT function. What function would the circuit compute if the position of the NMOS and PMOS transistors are reversed? Give a truth table for this circuit.

**Another Logic Gate:**

Above you saw that the logical NOT function can be implemented using two CMOS transistors. In fact, CMOS circuits can also be built for the other logic functions with which you are familiar with from programming such as AND (&&) and OR (||). Because AND and OR require multiple inputs, these circuits will also have multiple (two or more) inputs.

🔑 7. Consider the CMOS circuit shown in the box below. Notice that this circuit has two inputs (labeled A and B) and one output labeled (Z). Also note that the diagonal line does not connect to either of the two wires that it crosses (i.e. there is no thick dot at the crossing). Use the drawing tools in Word (or your favorite drawing program) to show:

* The value of the inputs set to A=0 and B=0.
* The positions of the imaginary “switches” in each of the four transistors.
* The value of the output (Z) that would be produced.



🔑 8. Complete the truth table below for this circuit by figuring out what the output Z will be for each possible combination of the inputs A and B.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | **A** | **B** | **Z** |  |
|  | 0 | 0 |  |  |
|  | 0 | 1 |  |  |
|  | 1 | 0 |  |  |
|  | 1 | 1 |  |  |
|  |  |  |  |  |

🔑 9. Compare your truth table from question #8 to those on the following page <https://www.electronics-tutorials.ws/boolean/bool_7.html>. Based on that comparison which logic function does the CMOS circuit from question #7 compute?

🔑 10. Like with the NOT gate earlier the CMOS circuit in question #7 can also be represented abstractly using a logic gate. Using the link from question #9, draw the symbol for the logic gate that corresponds to the CMOS circuit.

**All the Logic Gates:**

As you can see from the page we were using above ([https://www.electronics-tutorials.ws/boolean/bool\_7.html](https://www.electronics-tutorials.ws/boolean/bool_7.htmlt)) there are logic gates for a quite a few different logical (a.k.a. Boolean) functions. It is not required at this point, but if you would like to read a little more about the Boolean functions you can review the page: <https://ryanstutorials.net/boolean-algebra-tutorial/boolean-algebra.php>, which gives more information about each of them.

🔑 11. Each logic gate serves as an abstraction for a circuit using transistors that computes the corresponding Boolean function (e.g. like the two given earlier). Give the logic symbol and truth table for each of the logic functions listed below.

* Note that the different input combinations are always listed in the same order. This is a convention that is used by computer scientists. If you have taken MATH 211 and seen truth tables there, you will note that the ordering is different. Please use the computer science convention for ordering in this class.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | **Function** | **Logic Gate Symbol** | **Truth Table** |  |
|  | OR |  | |  |  |  | | --- | --- | --- | | **A** | **B** | **Z** | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |
|  | NOR |  | |  |  |  | | --- | --- | --- | | **A** | **B** | **Z** | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |
|  | AND |  | |  |  |  | | --- | --- | --- | | **A** | **B** | **Z** | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |
|  | XOR |  | |  |  |  | | --- | --- | --- | | **A** | **B** | **Z** | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |
|  | XNOR |  | |  |  |  | | --- | --- | --- | | **A** | **B** | **Z** | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |
|  |  |  |  |  |

🏆 12. At the end of the slides from class there was a challenge question about a different CMOS circuit. What logic function is computed by that circuit? Briefly explain in a few sentences how you determined your answer.

🏆🏆 13. Consider the CMOS circuit shown below.

A picture containing text, clock

Description automatically generated

This circuit uses the NMOS and PMOS symbols from question #5c, so you will need to complete that question before this one.

a. Give a truth table that shows the value of the output Q for all possible combination of the inputs A and B.

b. What logic function does this circuit compute?

**Reflecting on Abstractions:**

🔑 14. Returning to the definition of an abstraction from the first class, briefly explain in a few sentences of your own words why logic gates are an abstraction. Be sure to clearly identify the relevant information that is revealed and the irrelevant information that is hidden when working at the logic gate level of abstraction.

**A Little History:**

Watch the following video that covers modern computing history:

* <https://www.youtube.com/watch?v=LN0ucKNX0hc> (10:43)

🏆 15. Write a sentence or two in your own words that provide an answer the following questions: What was the first computer “bug”? In which computer was it found? Who reported it? How was that “bug” related to electromechanical relays?

🏆 16. Write a sentence or two in your own words that provide an answer the following questions: Which machine is recognized as the first programmable electronic computer? What type of switching technology did it use? Where was it built? What was its primary purpose?

🏆 17. Write a sentence or two in your own words that provide an answer the following questions: Which machine is recognized as the first *general purpose* programmable electronic computer? Who designed it? Where was it built?

🏆 18. Write a sentence or two in your own words that provide an answer the following questions: Where was the transistor invented? Who invented it?

**Additional Perspective:**

The following videos are not required viewing, but if you are interested, they will provide additional information, different perspectives and alternative explanations of the material that you may find helpful.

* A more conceptual discussion of Boolean Logic & Gates with Carrie Anne from Crash Course Computer Science.
  + <https://www.youtube.com/watch?v=gI-qXk7XojA> (10:06)
* A Coursera lecture with a detailed coverage of CMOS Logic Gates from Dr. Bonnie Ferri’s Introduction to Electronics at Georgia Tech.
  + <https://www.coursera.org/lecture/electronics/6-3-cmos-logic-gates-TA8qw> (10:40)

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 activity outside of class time?

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