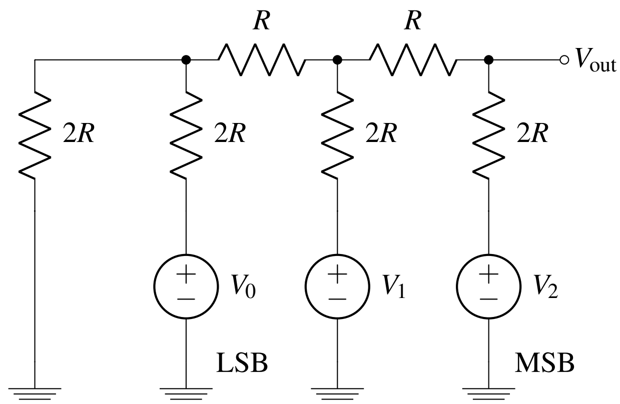
This homework is due on Wednesday September 5, 2018, at 11:59PM. Self-grades are due on Monday, September 10, 2018, at 11:59PM.

1. Digital-Analog Converter

A digital-analog converter (DAC) is a circuit for converting a digital representation of a number (binary) into a corresponding analog voltage. In this problem, we will consider a DAC made out of resistors only (resistive DAC) called the R-$2R$ ladder. Here is the circuit for a 3-bit resistive DAC.



Let $b_0, b_1, b_2 = \{0, 1\}$ (that is, either 1 or 0), and let the voltage sources $V_0 = b_0 V_{DD}$, $V_1 = b_1 V_{DD}$, $V_2 = b_2 V_{DD}$, where V_{DD} is the supply voltage.

As you may have noticed, (b_2, b_1, b_0) represents a 3-bit binary (unsigned) number where each of b_i is a binary bit. We will now analyze how this converter functions.

(a) If $b_2, b_1, b_0 = 1, 0, 0$, what is V_{out}? Express your answer in terms of V_{DD}.

(b) If $b_2, b_1, b_0 = 0, 1, 0$, what is V_{out}? Express your answer in terms of V_{DD}.

(c) If $b_2, b_1, b_0 = 0, 0, 1$, what is V_{out}? Express your answer in terms of V_{DD}.

(d) If $b_2, b_1, b_0 = 1, 1, 1$, what is V_{out}? Express your answer in terms of V_{DD}.

(e) Finally, solve for V_{out} in terms of V_{DD} and the binary bits b_2, b_1, b_0.

(f) Explain how your results above show that the resistive DAC converts the 3-bit binary number (b_2, b_1, b_0) to the output analog voltage V_{DD}.

2. Logical Comparator

In this problem, we are going to implement a two-bit comparator. We take in two numbers a, b, which are composed of bits a_1, a_0, b_1, b_0 respectively. For example, $a_1 = 1, a_0 = 0$ would mean $a = 10_2 = 2_{10}$.

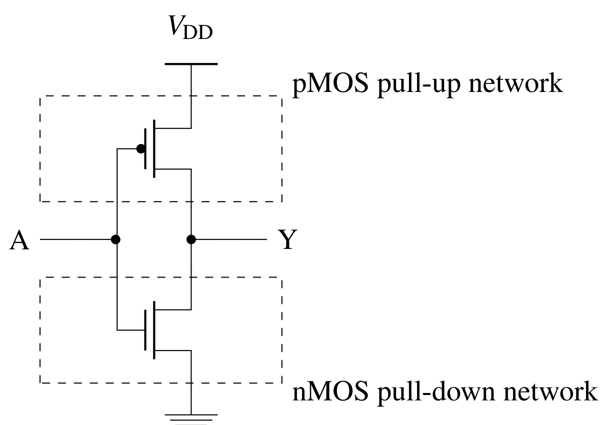
(a) Let's first implement a one-bit comparator. Give a logical expression for the $>$ operator. To be more specific, create a logical function such that $f(a_0, b_0)$ will be true if $a_0 > b_0$ and false otherwise.

(b) Now, let's extend this function to implement a two-bit comparator. More specifically, make a function $g(a,b)$ such that $g(a,b)$ is true if $a > b$ and false otherwise. Recall that a,b can be decomposed to two digits each. You may optionally use $f(a_0, b_0)$ to implement $g(a,b)$.

3. Transistors and Boolean Logic

A boolean formula can be implemented in digital circuitry using nMOS and pMOS transistors. In circuits, the truth value 1 (*true*) is represented by a high voltage, called POWER (V_{DD}). The truth value 0 (*false*) is represented by a low voltage, called GROUND (GND). In this problem, we will only use the truth values in order to simplify notations. That is, if you see $A = 1$ for a point A, then it means the voltage of A is equal to V_{DD}. Similarly, if $A = 0$, then the voltage of A is equal to GND.

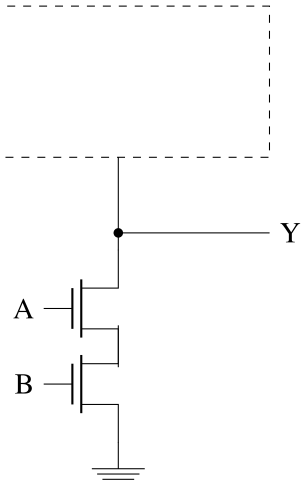
An inverter can be implemented with 1 nMOS and 1 pMOS, as shown in the figure below. When the input A is 0, then the nMOS is OFF and the pMOS is ON. Thus, the output Y is pulled up to 1 because it is connected to V_{DD}. Conversely, when A is 1, then the nMOS is ON and the pMOS is OFF, and Y is pulled down to 0. Therefore, the circuit implements the Boolean formula, $Y = \overline{A}$.



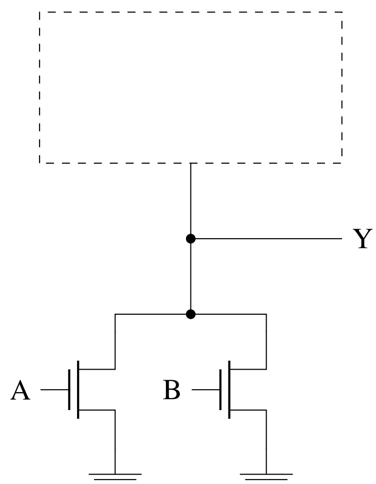
In general, a Boolean-formula circuit has an nMOS *pull-down network* to connect the output to 0 (GND) and a pMOS *pull-up network* to connect the output to 1 (V_{DD}). The pull-up and pull-down networks in the inverter example each consist of a single transistor.

In this problem, you will design pull-up networks given the pull-down networks.

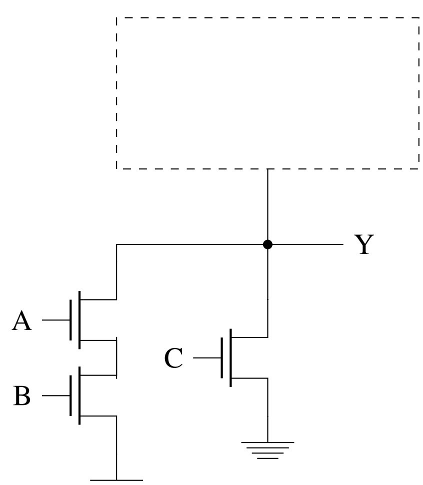
(a) The pull-down network of the Boolean formula (a 2-input NAND gate), $Y = \overline{(A \cdot B)}$, is given below. Design the pull-up network (the dashed box) with 2 pMOS transistors.



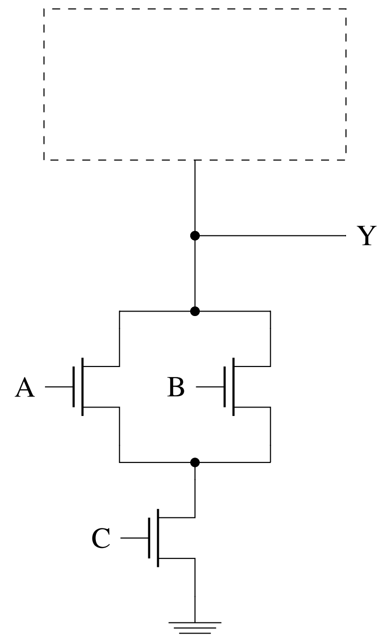
(b) The pull-down network of the Boolean formula (a 2-input NOR gate), $Y = \overline{(A+B)}$, is given below. Design the pull-up network (the dashed box) with 2 pMOS transistors.



(c) The pull-down network of the Boolean formula, $Y = \overline{\left((A \cdot B) + C\right)}$, is given below. Design the pull-up network (the dashed box) with 3 pMOS transistors.

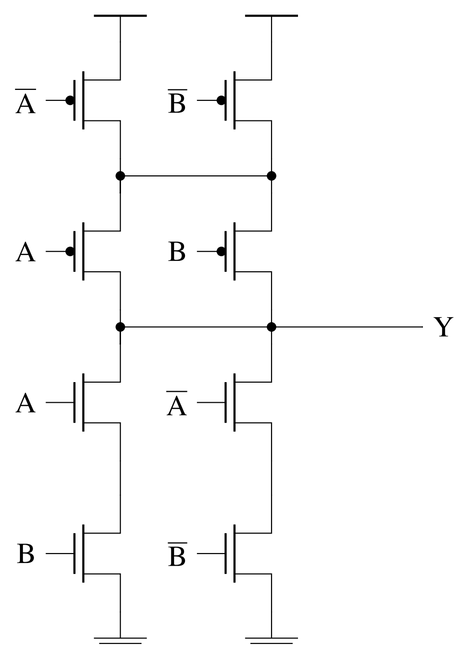


(d) The pull-down network of the Boolean formula, $Y = \overline{((A+B)\cdot C)}$, is given below. Design the pull-up network (the dashed box) with 3 pMOS transistors.



(e) For the circuit below, write the truth table for inputs A and B with output Y. What boolean operation is this?

Note some of the gate voltages are $\overline{A}$ and $\overline{B}$.

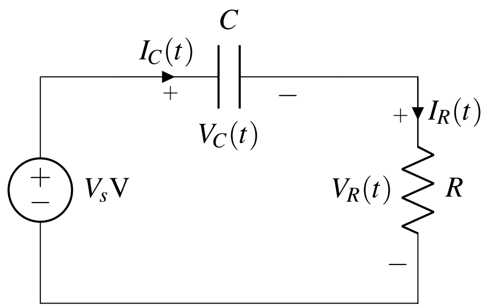


4. Mechanical: First-Order Differential Equation Practice

(a) Solve the equation $3\frac{d}{dt}f(t)+6f(t)=0$ given that $f(0)=7$

(b) Solve the equation $3f(t)-6f'(t)=4$ given that $f(2)=\frac{1}{3}$

5. RC Circuit



(a) Find a differential equation for $V_c(t)$ for $t \geq 0$. Solve the differential equation using the initial condition $V_c(0) = 1$V. Use component values of $C = 1$fF ($1\text{fF} = 10^{-15}$F), $R = 10\text{k}\Omega$, and $V_s = 2$V

(b) Instead of having an initial condition of $V_c(0) = 1$V, we now have an initial condition of $I_R(t) = 150\mu$A ($1\mu\text{A} = 10^{-6}$A). Find the new expression for $V_C(t)$ for $t \geq 0$. Use the same component values listed in part (a)

6. Write Your Own Question And Provide a Thorough Solution.

Writing your own problems is a very important way to really learn material. The famous "Bloom's Taxonomy" that lists the levels of learning is: Remember, Understand, Apply, Analyze, Evaluate, and Create. Using what you know to create is the top level. We rarely ask you any homework questions about the lowest level of straight-up remembering, expecting you to be able to do that yourself (e.g. making flashcards). But we don't want the same to be true about the highest level. As a practical matter, having some practice at trying to create problems helps you study for exams much better than simply counting on solving existing practice problems. This is because thinking about how to create an interesting problem forces you to really look at the material from the perspective of those who are going to create the exams. Besides, this is fun. If you want to make a boring problem, go ahead. That is your prerogative. But it is more fun to really engage with the material, discover something interesting, and then come up with a problem that walks others down a journey that lets them share your discovery. You don't have to achieve this every week. But unless you try every week, it probably won't ever happen.

Contributors:

- Edward Wang.
- Jaymo Kang.
- Yen-Sheng Ho.
- Kyle Tanghe.