18-100: Intro to Electrical and Computer Engineering

LAB03: MOSFET Lab

Writeup Due: Thursday, September 22nd 2022 at 10 PM

Name: Bing Bhakdibhumi

Andrew ID: bbraneeb

How to submit labs:

Download from this file from *Canvas* and edit it with whatever PDF editor you're most comfortable with. Some recommendations from other students and courses that use Gradescope include:

DocHub	An online PDF annotator that works on desktop and mobile platforms.
pdfescape.com	A web-based PDF editor that works on most, if not all, devices.
iAnnotate	A cross-platform editor for mobile devices (iOS/Android).

If you have difficulties inserting your image into the PDF, simply append them as an extra page to the END of your lab packet and mark the given box. ***Do NOT insert between pages.***

If you'd prefer not to edit a PDF, you can print the document, write your answers in neatly and scan it as a PDF. *(Note: We do not recommend this as unreadable lab reports will not be graded!).* Once you've completed the lab, upload and submit it to *Gradescope*.

Note that while you may work with other students on completing the lab, this writeup is to be completed alone. Do not exchange or copy measurements, plots, code, calculations, or answer in the lab writeup.

Your lab grade will consist of two components:

1. Answers to all lab questions in your lab handout. The questions consist of measurements taken during the lab activities, calculations on those measurements and questions on the lab material.
2. A demonstration of your working lab circuits and conceptual understanding of the material. This demo will occur during your small group meeting with your TAs.

Question:	1	2	3	4	Total
Points:	5	10	12	13	40
Score:					

Lab Outline

This lab aims to teach the "switching" property of MOSFET transistors and how MOSFETs can be used to build fundamental logic gates.

1. NMOS Inverter
2. CMOS Inverter
3. CMOS NOR Gate
4. CMOS SR Latch

Equipment Required

Breadboard
Digital Multimeter
Diagonal Cutters
Power Supply
Wire Strippers
Needle-nose Pliers

Bill of Materials

2x  470Ω Resistor
1x  1kΩ Resistor
3x  10kΩ Resistor
3x Pushbutton Switch
3x Red LEDs
5x p-MOSFET (ZVP3306A)
5x n-MOSFET (2N7000, labeled ZVN3306A)
Jumper Wire

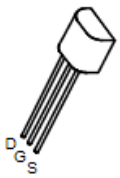
Small Group Check-off Circuits

☐ CMOS Inverter (pg. 5)

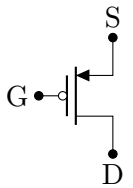
☐ CMOS SR Latch (pg. 7)

☐ CMOS D Latch (pg. 9; bonus; counts for both checkoff items above)

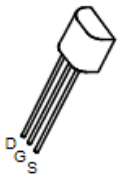
Pinouts



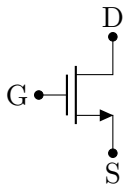
ZVP3306A P-Ch MOSFET Pinout



P-Ch MOSFET Symbol



2N7000 N-Ch MOSFET Pinout



N-Ch MOSFET Symbol

Introduction

Metal-Oxide Semiconductor Field-Effect Transistors or MOSFETs are the foundation of modern digital circuitry. They have three main pins: the gate, drain, and source. A MOSFET is a *voltage-controlled current source.* The voltage applied to the gate will determine the current between the drain and the source, which is known as the *drain current*, I_{DS}.

The MOSFETs we will be working with in this lab come in two flavors: N-Channel and P-Channel, or NMOS and PMOS for short. With N-Ch MOSFETs, when a voltage greater than the voltage at the source is applied to the gate, current will flow from the drain to the source. This is analogous to a switch being closed when a voltage is applied.

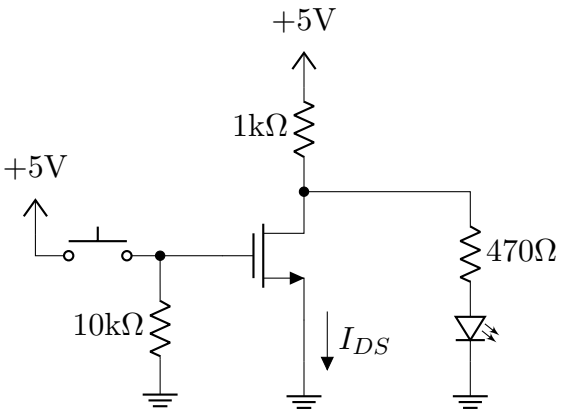
Conversely, a P-Ch MOSFET will allow current to flow from source to drain when the gate voltage is lower than the source voltage. This is analgous to a switch being opened when a voltage is applied. We can use these complementary behaviors to create Complementary MOS or CMOS.

In this lab, we'll take a look at why CMOS logic is so powerful and how we can use it to form the building blocks of modern computation.

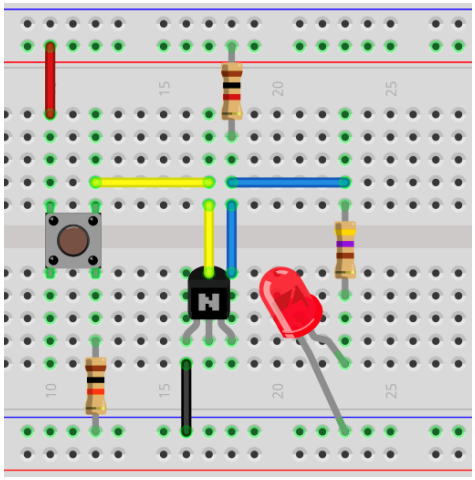
1. NMOS Inverter

Before we can understand why CMOS logic is so important, let's take a look at the most basic logic gate: the inverter.

Build the following circuit (in Figure 1) on a breadboard:



(a) NMOS Inverter Schematic



(b) NMOS Inverter Breadboard

Figure 1: N-Channel MOS Inverter

3 pts **1.1** Measure the gate, drain and source voltages with respect to ground (V_G, V_D, V_S respectively). Also measure drain-source current (I_{DS}) when the switch is open and closed. **For a more accurate measurement of I_{DS}, please remove the LED when measuring this current**. Then calculate the power consumed by the transistor and 1 kΩ pull-up resistor for both states.

SW	Open	Closed
V_D	2.77 V	0 V
V_G	0 V	4.98 V
V_S	0 V	0 V
I_{DS}	0 mA	5.0 mA
P	0 W	25 mW

Show work here:

$P = IV$

$P = 5\,V\ (5\,mA)$

$= 25\,mW$

2 pts **1.2** Identify one downside of this inverter design.

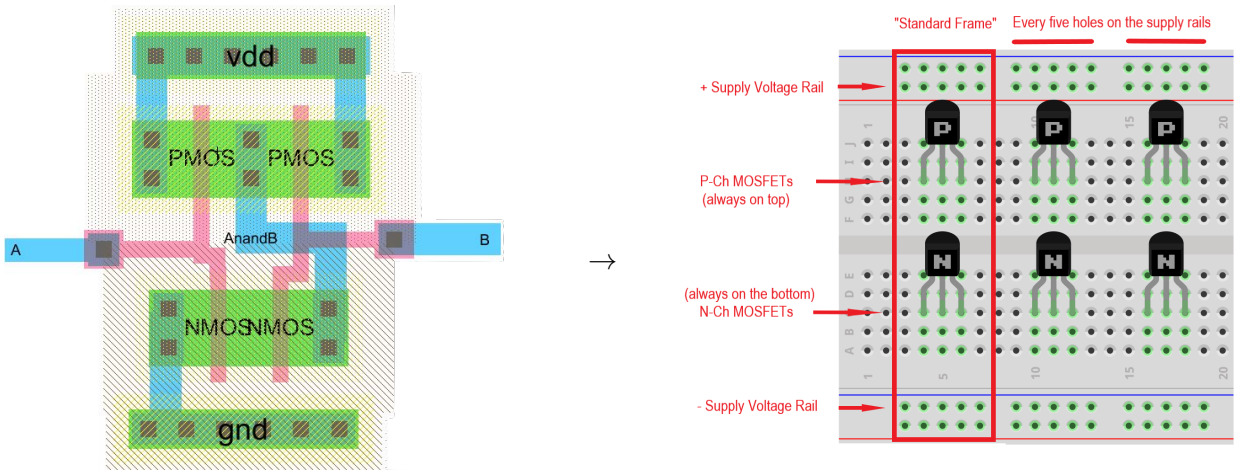
High power consumption in the context of a device.

2. CMOS Inverter

As we saw in Section 1, a single transistor is not perfect by any means. In this section we'll see how introducing a new type of MOSFET, the P-Channel MOSFET, will change the way we approach building logic gates.

However, before we begin working with P-Channel MOSFETs, let's establish a set of guidelines to help keep your circuits clean and easy to understand.

A common design paradigm when developing microelectronic circuits is the concept of a "standard cell." Due to the way PMOS and NMOS transistors are fabricated on integrated circuits, microelectronic circuit designers it's helpful to have each transistor type in its own row. Because the source of a PMOS transistor is often connected to the positive supply rail or the drain of another PMOS transistor, its helpful to have all the PMOS on the top row (see Figure 2). Conversely, the source of NMOS transistors are almost always connected to the drain of another NMOS or to ground. Hence why NMOS transistors are always on the bottom.



(a) "Standard Cell" CMOS Layout (NAND)

(b) Breadboard "Standard Cell"

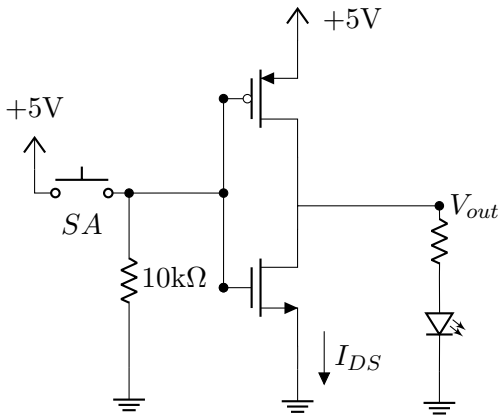
Figure 2: Standard CMOS cell and 18-100's Breadboard Approximation

Using the breadboard layout as our "standard cell" we can very easily create complex CMOS circuits.

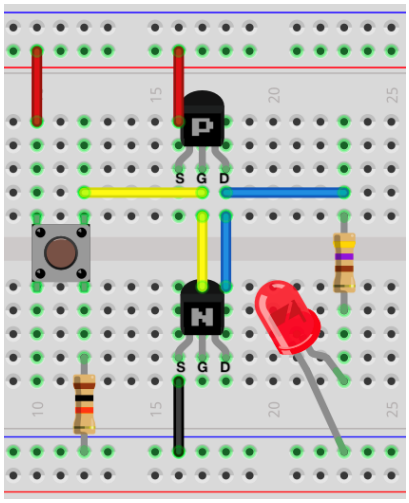
> ⚠ **We will ask that you build all of your circuits using this layout. Circuits that are messy/illegible may not receive credit.**

We'll build an example of this design on the next page.

5 pts **2.1** Modify your NMOS inverter to build a CMOS inverter on your breadboard. We've labeled the drain, gate, and source for each of the MOSFETs on the breadboard diagram below (Figure 3).



(a) CMOS Inverter Schematic



(b) CMOS Inverter Breadboard Layout

Figure 3: Complementary MOS Inverter

> ⚠ **Do NOT take your circuit apart yet! You will need it for lab checkoff!**

3 pts **2.2** Measure the voltage at the gates of the MOSFETs and at V_{out}. Also measure drain-source current (I_{DS}) when the switch is open and closed. For a more accurate measurement of I_{DS}, please remove the LED when measuring this current. Then calculate the power consumed by the MOSFETS for both states

SW	Open	Closed
V_G	0 V	4.99 V
V_{out}	4.88 V	0 V
I_{DS}	0 A	0 A
P	0 W	0 W

Show work here:

P = IV

P = 0 (5V) = 0 W

2 pts **2.3** Identify one improvement this inverter design has in comparison to the NMOS inverter.

much lower power consumption since there is no current from source to ground.

3. CMOS NOR Gate

A NOR gate can be built from 2 N-channel MOSFETs and 2 P-channel MOSFETs, as shown below (Figure 4).

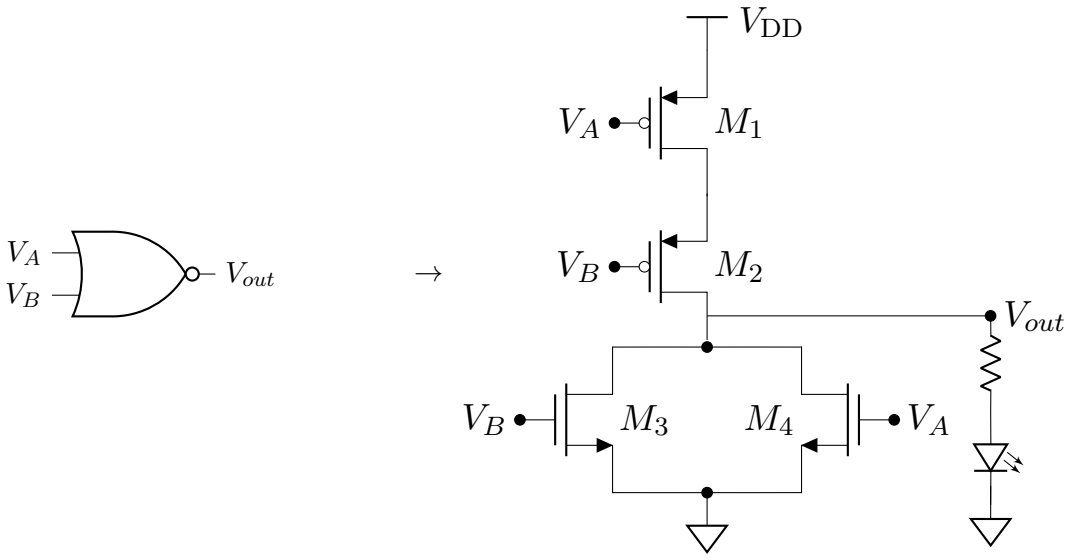


Figure 4: CMOS NOR Circuit

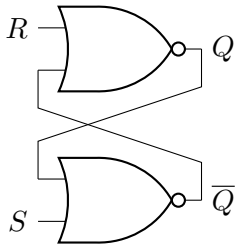
10 pts **3.1** Build the NOR gate from N/P-channel MOSFETs on a new section of the breadboard. You'll need to wire two switches, one for each of the inputs A and B. Keep in mind that while the pull down-resistors are not shown in the diagram, they should be included in your circuit, for both V_A and V_B.

2 pts **3.2** Take measurements of V_{out} and fill out the table with your readings. Verify that your 2-Input NOR gate behaves correctly by comparing the output voltage levels with the logic levels in the truth table of a NOR gate.

V_{A}	V_{B}	V_{out} (V)
0 V	0 V	5 V
0 V	5 V	0 V
5 V	0 V	0 V
5 V	5 V	0 V

4. CMOS SR Latch

Consider the following schematic and truth table of an SR latch:



(a) SR Latch Schematic

S	R	Q	$\overline{Q}$
0	0	latch	latch
0	1	0	1
1	0	1	0
1	1	meaningless	

10 pts **4.1** Build the SR latch out of discrete MOSFETs by expanding upon the NOR gate you built in the previous section. Add a second indicator LED to the $\overline{Q}$ output.

> ⚠ **Do NOT take your circuit apart yet! You will need it for lab checkoff!**

3 pts **4.2** We call the input combination of S = 1 and R = 1 as 'meaningless.' Test this input combination on your SR latch a few times and describe what you observe. While there is no law against doing this, why is this input combination one we should avoid?

There is no guarantee what the output will actually be. Pretty much depends on which signal arrives first.

5. Bonus: CMOS D Latch

As discussed in lecture, latches are *asynchronous*, meaning that their inputs change as soon as they receive the appropriate signal, and that signal could come at any time. In this section you will build a D latch.

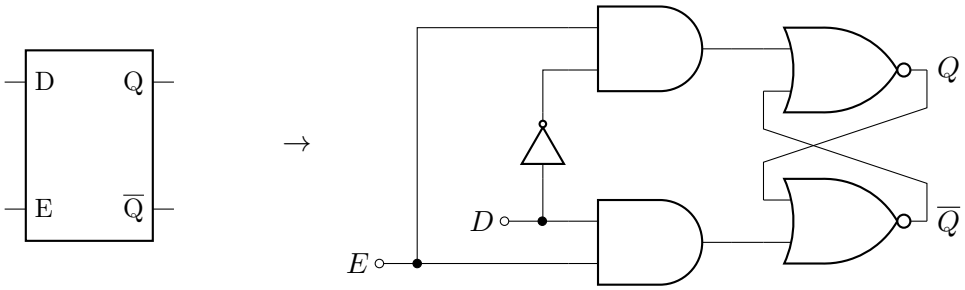


Figure 6: D-flop-flop symbol and equivalent logic circuit

As seen in lecture, a D latch is represented by the symbol above and can be built with the equivalent logic circuit to the right. D is the data input, E is the enable line and $Q/\overline{Q}$ is the output.

However, since the two CMOS AND gates would require two CMOS NANDs followed by two CMOS NOTs, we would require more transistors than necessary. In general, it is good practice to minimize the number of transistors in your design to save physical space and reduce propagation delay (transistors do not switch instantantly so adding transistors adds to the time it takes for your circuit to stabilize).

Thus, we will switch to NOR gates in an attempt to reduce our number of transistors. The circuit below is logically equivalent to the circuit above (As an exercise, try verifying this yourself!).

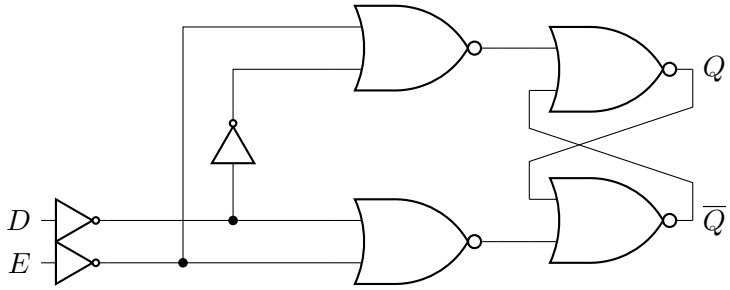
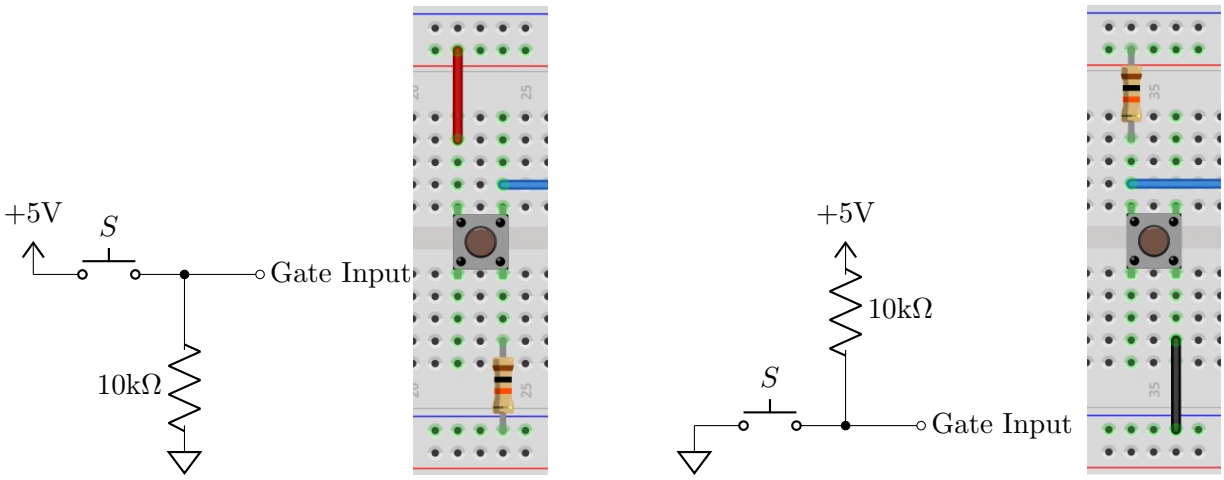


Figure 7: Alternative equivalent logic circuit for a D latch

You may notice that both the D and E inputs are inverted before being fed into the rest of the circuit, so we still have the problem of requiring additional transistors for those NOT gates. This is where 'active low' inputs come to the rescue! Instead of constructing NOT gates, we will create the necessary inverting logic at the inputs with 'active low' buttons that make use of pull-up resistors (as opposed to the usual pull-down resistors) as shown in 8.



(a) Active High Button (b) Active Low Button

Figure 8: Active Low and High Button Circuits

Earlier in this lab, we used 'active high' buttons, as shown in 8a. When the button is pressed, the output will rise to the positive supply voltage, and when released, the output will be grounded. By switching from using a pull-down resistor to a pull-up resistor, we can make it so that pressing the button will ground our output and releasing the button will raise it to the supply voltage. This circuit is shown in 8b and **using this type of input will remove the need for the two NOT gates on the D and E inputs.**

5 bonus

5.1 Using your CMOS Inverter from Section 2 and your SR Latch from the previous section, construct a D latch based on the logic circuit in Figure 7, using active low buttons as a transistor-free way to emulate the behavior of the NOT gates on the D and E inputs.

Be prepared to show a TA your functioning D Latch for check off.

> ⚠ **Do NOT take your circuit apart yet! You will need it for lab checkoff!**