EE-230 Electronics
Project 2: Half Adder
May 16th, 2023

The Lab Rats

- Christopher Scott
- Cameron deLeeuw

Christopher Scott
Cameron deLeeuw

Abstract:

The goal of this project was to implement basic logic gates using the basic components we have been analyzing during class. Namely, we created an AND gate and an XOR gate using transistors, diodes, and resistors. Both of these circuits can be built from a smaller NAND gate, which we did implement, however, we decided to try out something new and implement one of the gates using NMOS components instead. So for our project, we built the XOR half as described in the assignment (though with altered resistor values), and implemented the AND half of the circuit using a MOSFET circuit that we came up with. This circuit does function as expected, as seen in our lab demonstration.

Introduction:

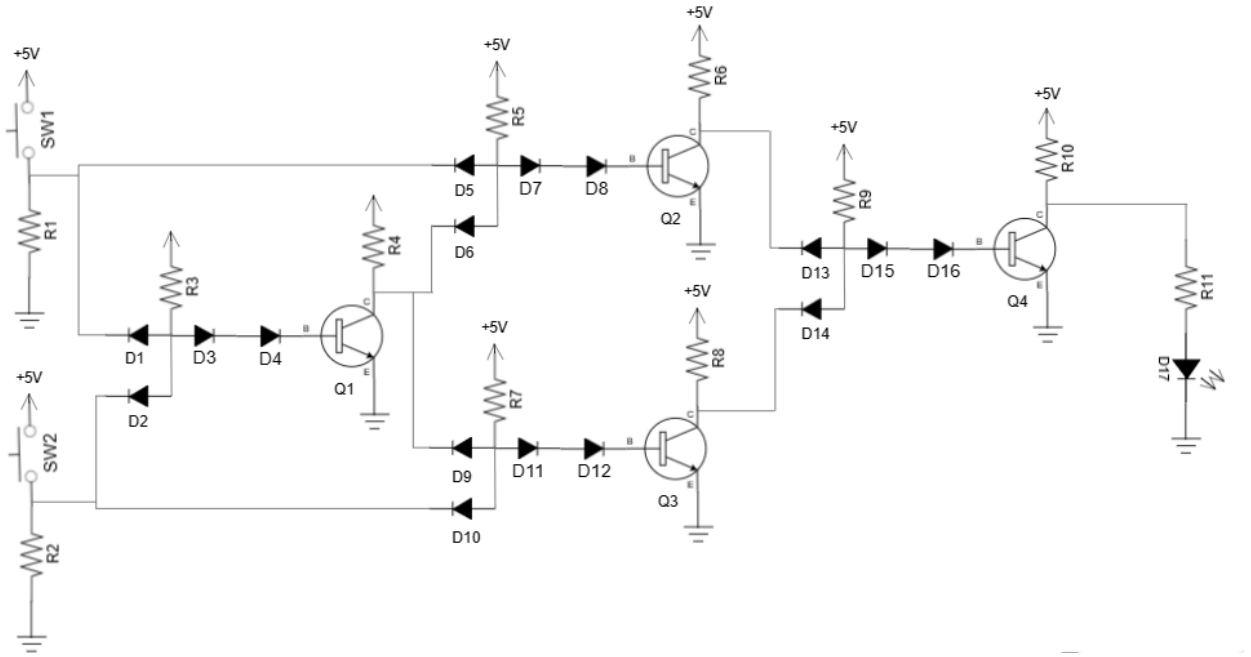
Objective:

To utilize transistors in a circuit to create a half adder circuit watch outputs a sum (XOR) and carry (AND).

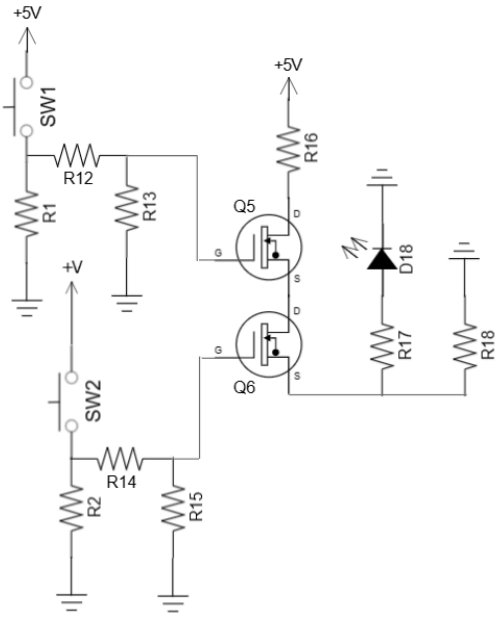
Materials:

Part	Schematic Name	Ideal Value	Measured Value	Percent Error
Resistor	R1	1500 Ω	1507 Ω	0.47% ± 0.03
Resistor	R2	1500 Ω	1509 Ω	0.6% ± 0.03
Resistor	R3	15 kΩ	14.85 kΩ	-1% ± 0.03
Resistor	R4	330 Ω	327.1 Ω	-0.88% ± 0.015
Resistor	R5	15 kΩ	14.83 kΩ	-1.13% ± 0.03
Resistor	R6	3300 Ω	3255 Ω	-1.36% ± 0.015
Resistor	R7	15 kΩ	14.87 kΩ	-0.87% ± 0.03
Resistor	R8	3300 Ω	3238 Ω	-1.89% ± 015
Resistor	R9	15 kΩ	14.82 kΩ	-1.2% ± 0.03
Resistor	R10	3300 Ω	3232 Ω	-2.06% ± 0.015
Resistor	R11	220 Ω	219.7 Ω	-0.136% ± 0.023
Resistor	R12	100 Ω	100.2 Ω	0.2% ± 0.05
Resistor	R13	1000 Ω	998 Ω	-0.2% ± 0.05
Resistor	R14	100 Ω	100.4 Ω	0.4% ± 0.05

Resistor	**R15**	**1000 Ω**	**997 Ω**	**-0.3% ± 0.05**
Resistor	**R16**	**100 Ω**	**100.7 Ω**	**0.7% ± 0.05**
Resistor	**R17**	**330 Ω**	**325.3 Ω**	**-1.42% ± 0.015**
Resistor	**R18**	**10 kΩ**	**10.05 kΩ**	**0.5% ± 0.05**
Diode	**D1 - D16**	**1N4001**	-	-
Transistor	**Q1 - Q4**	**2N3904**	-	-
Mosfet	**Q5, Q6**	**2N7000**	-	-
Pushbutton	**SW1, SW2**	-	-	-
LED	**D17, D18**	-	-	-



XOR Gate Schematic



AND Gate Schematic

Theoretical Analysis:

The AND Gate

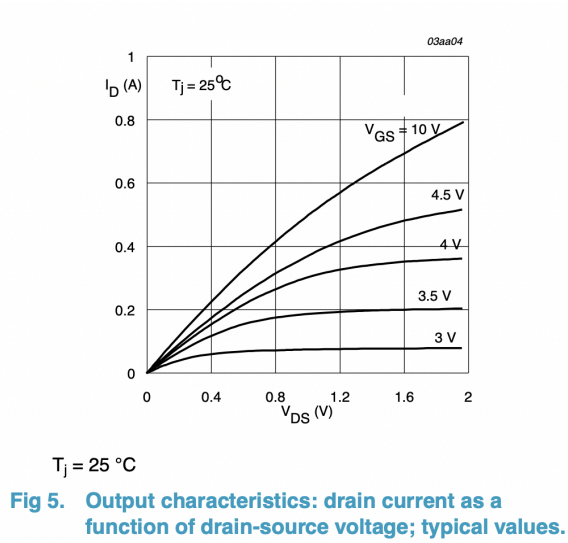
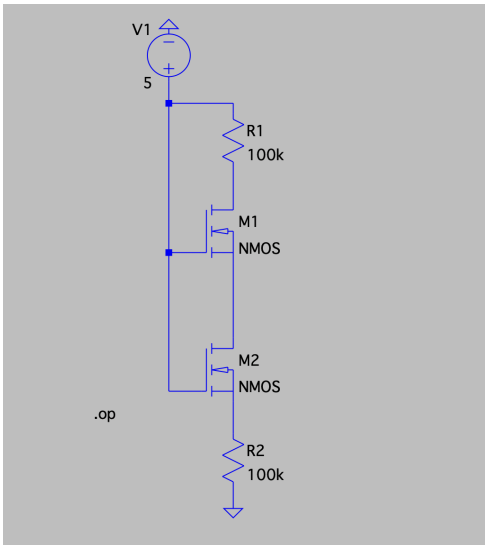
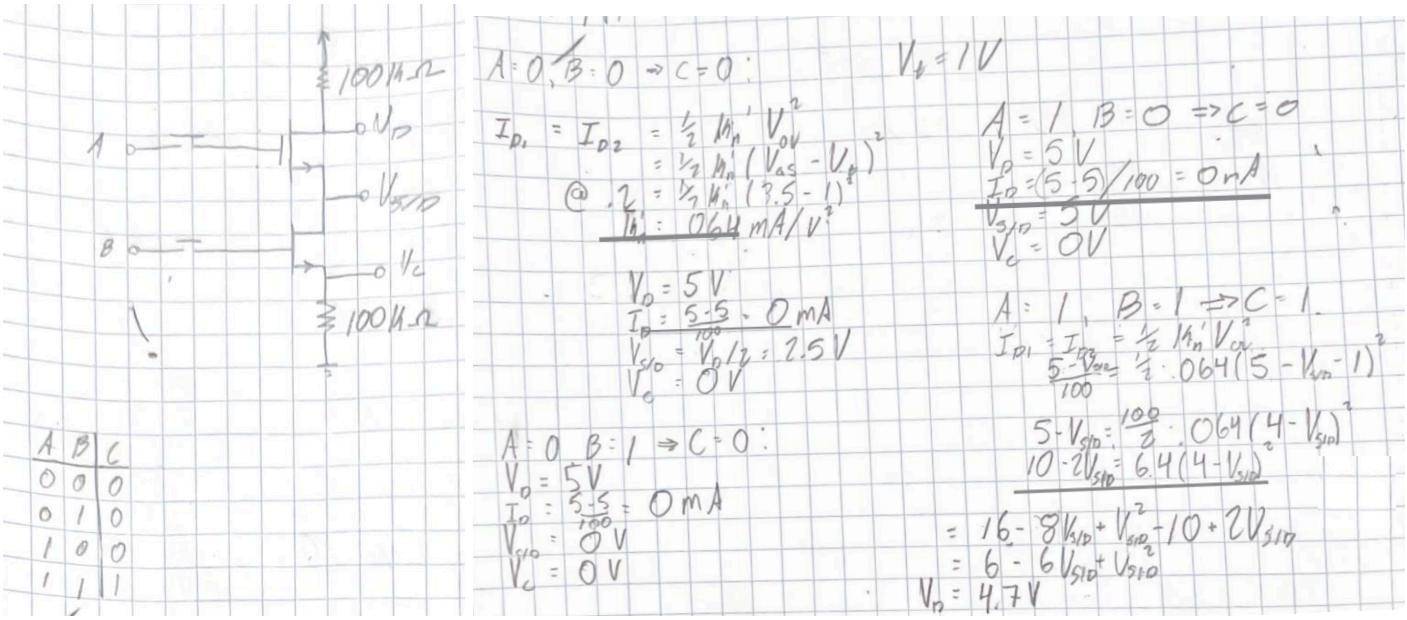


Fig 5. Output characteristics: drain current as a function of drain-source voltage; typical values.

Datasheet



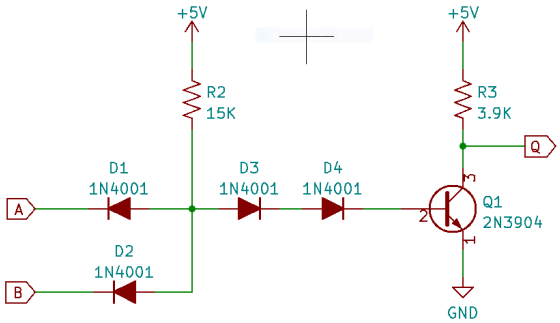
LTSpice Simulation



The AND gate portion of the circuit utilizes 2 MOSFETs in series. The output C only turns HIGH if both A and B buttons are pressed. In all other cases, C is LOW. Kn' was found from the datasheet of the 2N7000 NMOS transistors' I/V output characteristics. In our demonstration, we utilized a voltage divider network to bias the transistors but in theory, we can make the circuit operate the same with just two resistors.

The NAND Gates

While the concept of a NAND gate is not particularly complex (the output is high for any combination of inputs other than A * B = 1), understanding their implementation is somewhat more complex. A single NAND gate looks as follows:



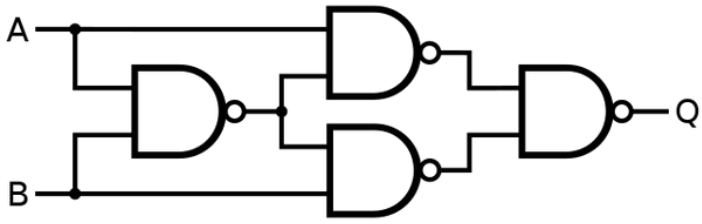
The output of this circuit, Q, entirely depends on the voltage at terminal 2, the base. If the voltage at the base is below the threshold, then the transistor is in cutoff mode, making output Q the only path for the 5V applied through R3, giving it a high voltage. If the voltage

at the base is above the threshold, then the transistor is in saturation mode and begins conducting and Q is effectively connected straight to ground, making its output 0 V.

So what determines the voltage at the base? That is really the diodes D1 and D2 in the diagram above. The voltage across a diode is a known value (typically 0.7 V). So any time A, B, or both A and B are low, they are considered at 0 V, which means the node shared by D1, D2, D3, and R2 must be approximately 0.7 volts. Because D3 and D4 are in series, they have a collective voltage drop of 1.4 V, ensuring that the 0.7 volts supplied are not enough to reach the base of the transistor. Therefore, anytime A, B, or both are low, the transistor must be in cutoff mode, making Q approximately 5 V. The only other possibility is that A and B are both high, approximately 5 volts. There is no way in this circuit for any node to have more than 5 volts, so D1 and D2 are necessarily in a reverse bias as the voltage across them is less than 0.7 V. With both of them in reverse bias, the full 5 volts is applied only to R2 as well as D3 and D4. The two diodes drop the voltage slightly, but the voltage at the node is still kept above the threshold, putting the transistor into saturation mode, connecting Q to ground. All of this amounts to the following truth table:

Truth Table		
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

The XOR Gate



As we know from our Digital Logic class, all logic gates can be made from the simple NAND gate. Therefore, once we have one NAND gate working and understood, we need only chain them in the appropriate way in order to create the desired logic. Using the schematic shown above, we can create a NAND gate from the following logic:

Q = (((A*B)'*A)'*((A*B)'*B)')'

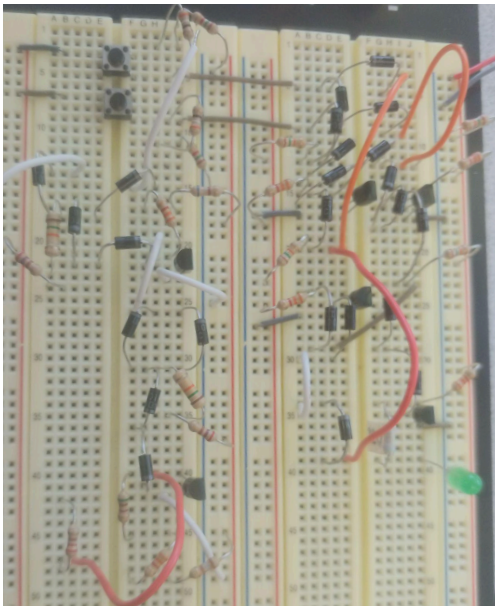
Which boils down to become:

$Q = AB' + A'B$

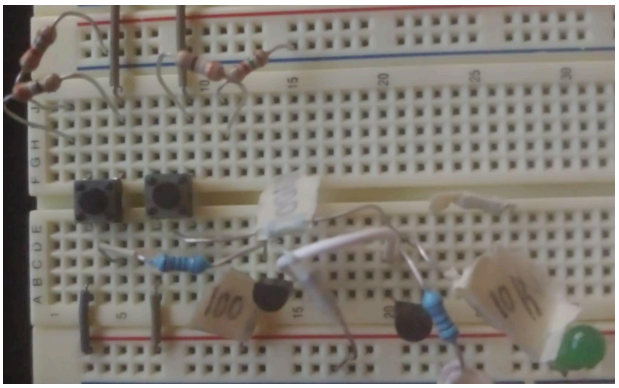
In other words, it becomes a simple Exclusive OR gate. One other change we decided to make to the circuit was to alter the resistor values somewhat. This was mostly done as a way of simplifying the circuit on our breadboard as we did not have any resistors of the exact value, so we found ourselves daisy-chaining resistors together to get as close as possible to the values listed in the project. After implementing the NMOS portion of the circuit, we found that we could change the resistor values to ones we already happened to have with no impact on circuit performance.

Results:

Ultimately, over the course of this project, we ended up making the functioning circuit twice. First, we made the circuit as described in the project assignment using a large number of diodes as pictured below.

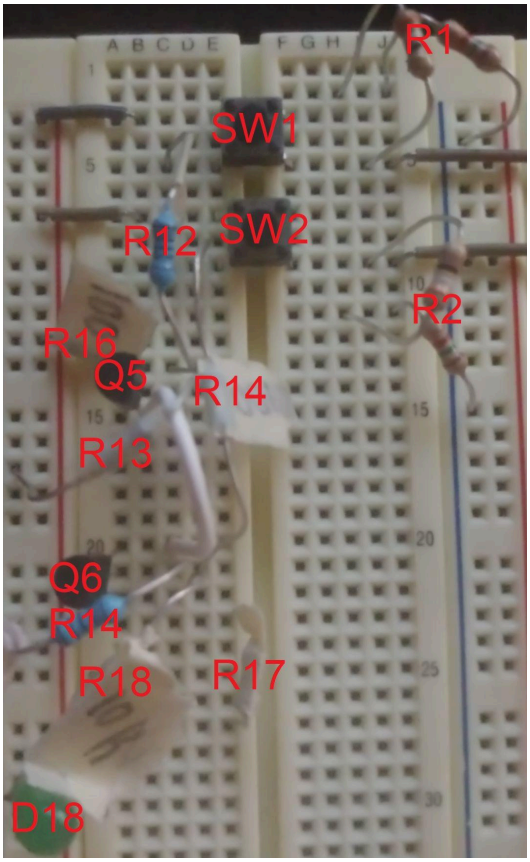


Then, we decided we could probably simplify this circuit somewhat using MOSFETS, and we came up with the following circuit to replace the AND gate.

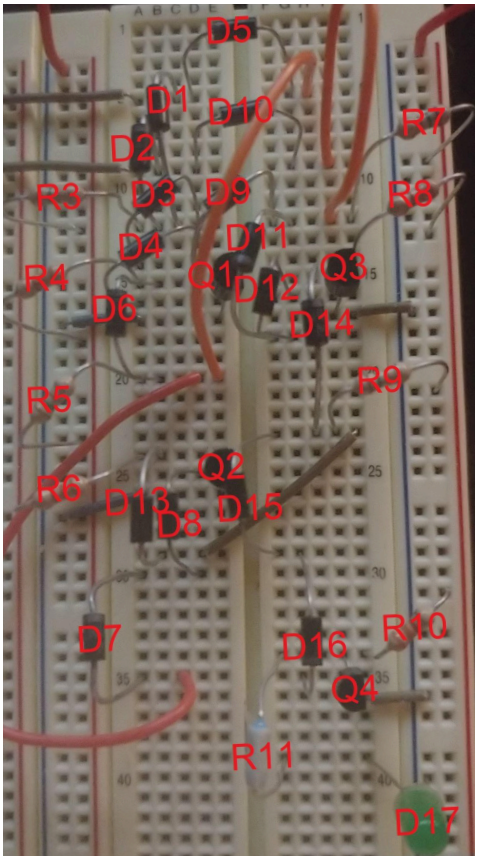


This change decreased the number of components significantly and still maintained the same functionality as the AND gate using BJTs. While we could likely have come up with a circuit to replace the XOR half as well, we found it to be satisfying to have a circuit that utilized both types of transistors at once, and so kept the circuit in this combined configuration. The final change we made was to swap out some of the resistor's values to decrease the board clutter.

And thus, we ended up with the final two halves of the circuit:

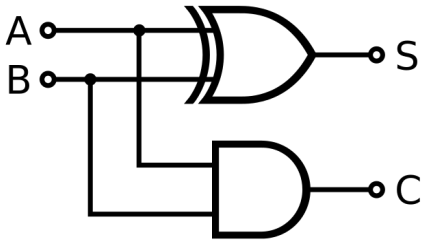


AND Gate



XOR Gate

This circuit logically creates the following diagram and truth table:



Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

It was actually very funny to us that we had completed this new version of the project using BJTs just a couple of days prior to these types of circuits being covered in class.

Conclusion:

In conclusion, ultimately we were in fact able to successfully implement a half-adder using basic electrical components. In fact, we were able to adapt half the circuit to function using fewer components by using NMOS instead of BJT NPN transistors. While not strictly necessary, given that we have spent a significant amount of time in class learning about them both, we found it to be very gratifying to be able to use both in our circuit. Perhaps the largest problem this project gave us was that it made us very aware of how sloppy we have a tendency to be when it comes to implementing circuits on a breadboard. With so many components involved, it becomes increasingly essential to keep your board organized in a way that keeps the circuit readable, which is something we struggled with somewhat, but it is a lesson we will keep in mind moving forward.