CSC 347/ENS 211

Title and Experiment #	Lab5: Single NAND/NOR Chip Implementation
Name	Gianna Galard
Date Performed	12-Oct-21
Date Submitted	15-Oct-21

The student pledges this work to be their own *Gianna Galard*

Objective:

This week's lab aims to create a circuit using a minimum number of chips and gates used. Doing so will decrease the overall power usage throughout the board and increase the reliability of the circuit. Given a simple circuit implemented with AND and OR gates, we create another minimized circuit that only uses NAND or XOR gates, displaying the same results.

Equipment and Chip used:

- TinkerCad
- 74HC**00** Quad NAND gate
- 74HC**02** Quad XOR gate

Design Procedure:

The first task was to build an all NAND gate implementation of the following circuit function:

$$F = AB + BC'$$

First, the student derived a truth table for this function, shown below:

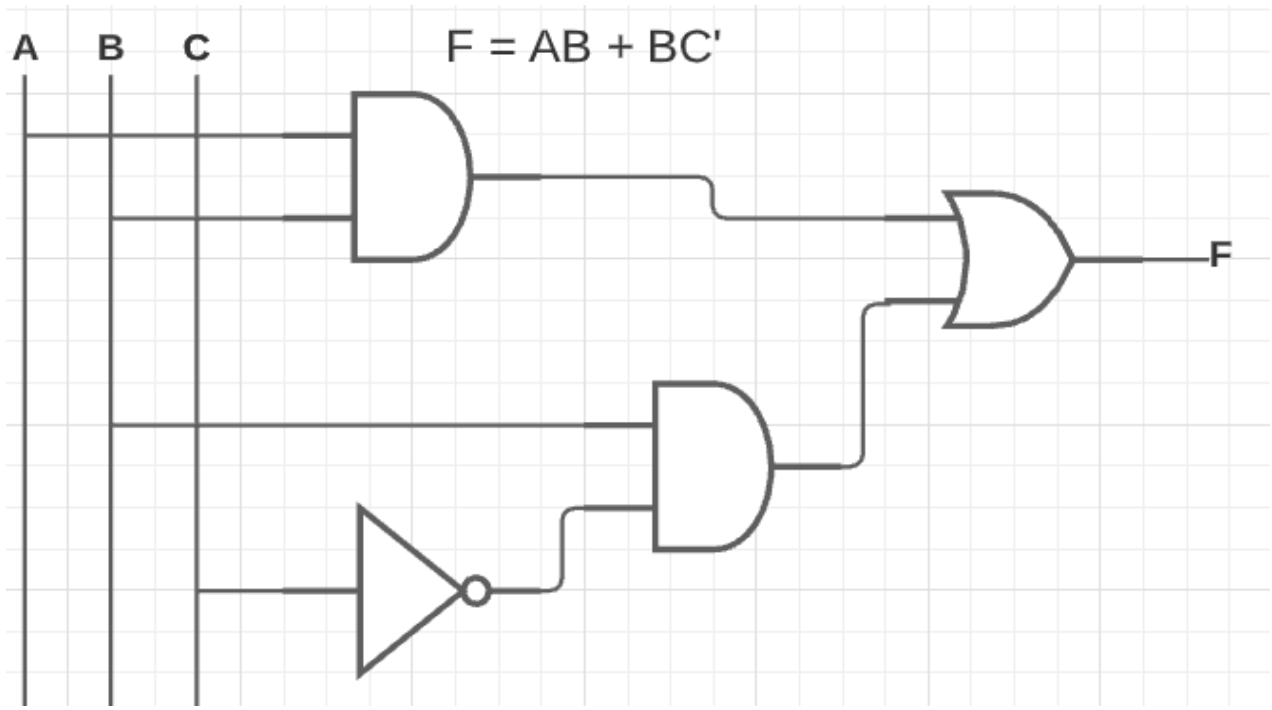
Figure 1. Truth Table

ABC	AB	BC’	F
000	0	0	0
001	0	0	0
010	0	1	1

011	0	0	0
100	0	0	0
101	0	0	0
110	1	1	1
111	1	0	1

Then the student drew a logic diagram that implements this circuit using AND, OR and inverter gates.

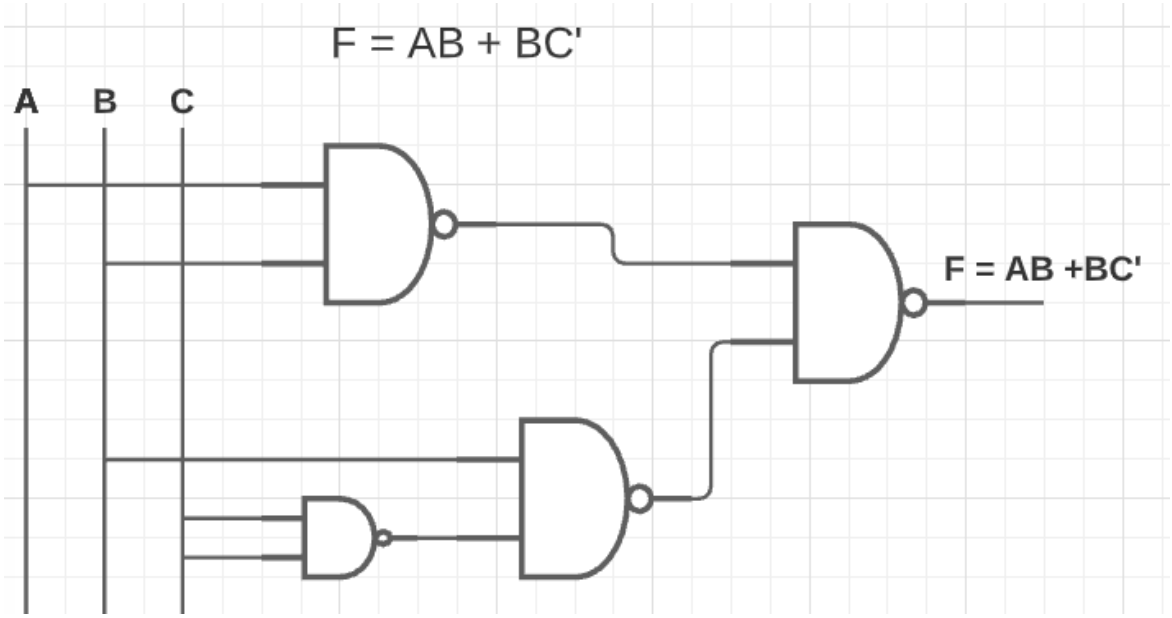
Figure 2. AND, OR, Inverter Gates Implementation Logic Diagram of F = AB+BC'



Then the circuit diagram was converted to one using all NAND gates. Converting the AND gate to a NAND gate is simple and requires only putting a NOT bubble in its front. However, this changes the circuit, and we must not do that; therefore, the student puts an inverter directly after the NAND gate to cancel it out. NAND is the complement of an AND gate. The two inverters following the NAND gates and leading towards

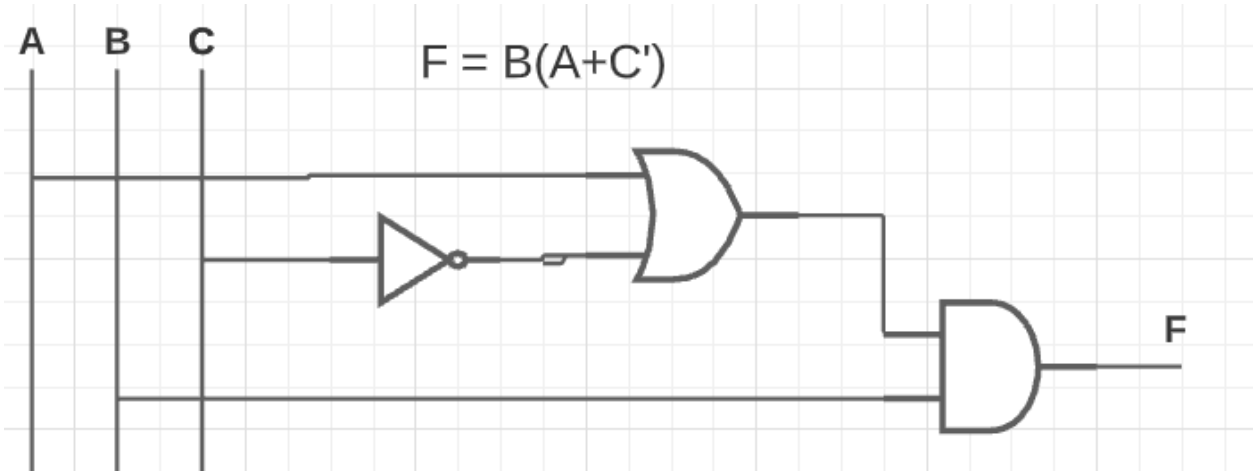
the final OR gate commutatively create another NAND gate as the final gate. By DeMorgan's law, x'+y' = (xy)' = NAND gate. For the inverter stemming from C, we change the inverter to a NAND gate by using C as both inputs to the NAND gate. CC = C, then through the NAND gate, it will equal C'. So x inverted and y inverted, together with leading up to a NAND gate, equals a normal OR gate. (x'y')' = x + y.

Figure 3. NAND Gate Implementation Logic Diagram of F = AB+BC'



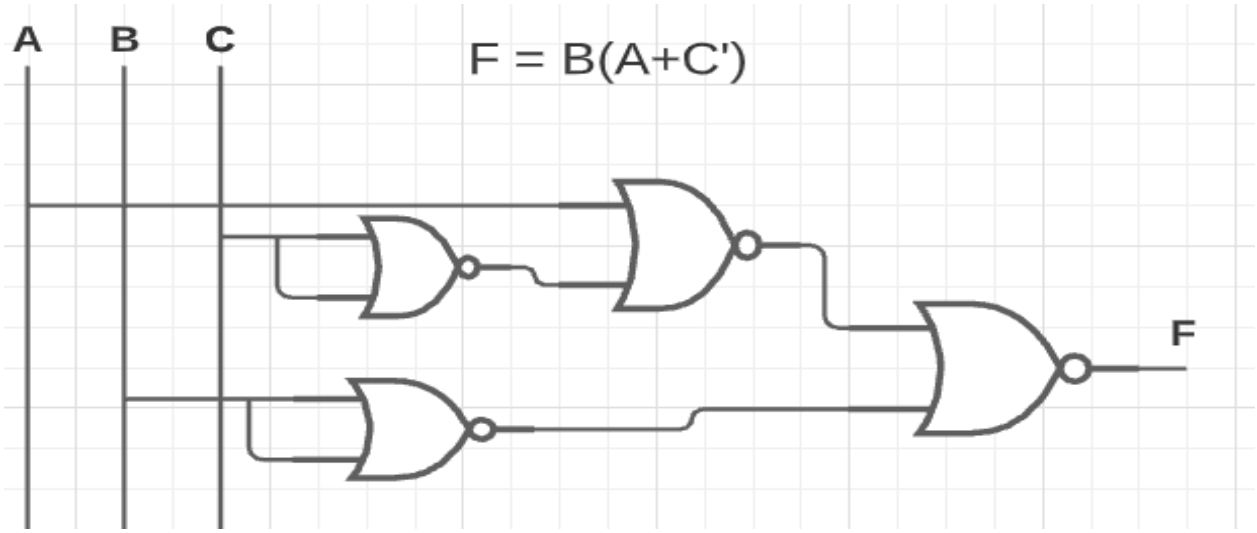
The second part of the lab has to do with converting the F = AB + BC' Boolean function from an SOP function to a POS form. The POS form is B(A+C').

Figure 4. POS form of function with OR, Inverter, and AND gates



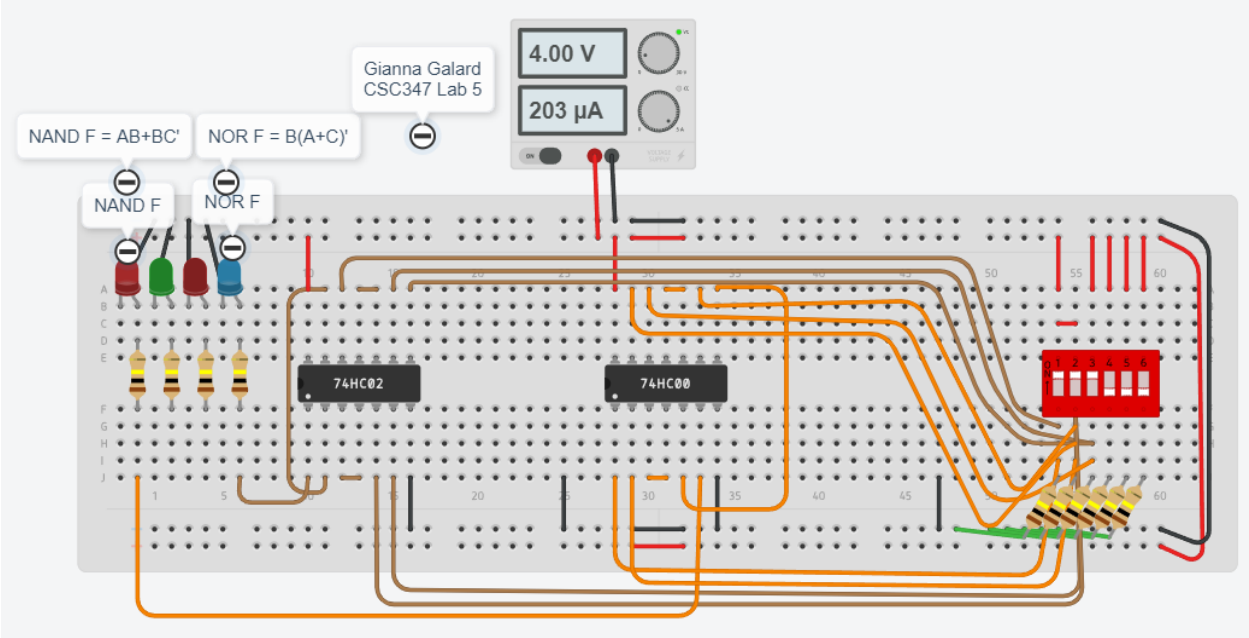
Then, the student transformed the function into an all NOR gate diagram.

Figure 5. POS form of function NAND gate implementation



To turn an OR gate into a NOR gate, you run the inputs through a NOR gate then invert the output by running it through the two inputs of a NOR gate. To make an AND gate with NOR gates, you invert both inputs and run them through a NOR gate.

Figure 6.1 1 1 = 1. Breadboard implementing both the SOP and POS function

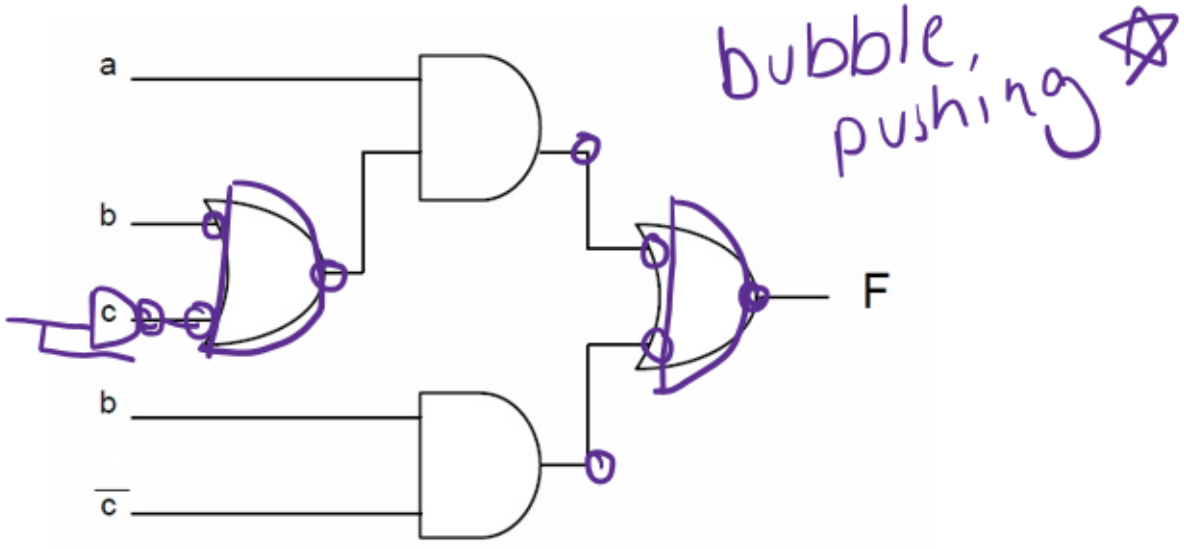


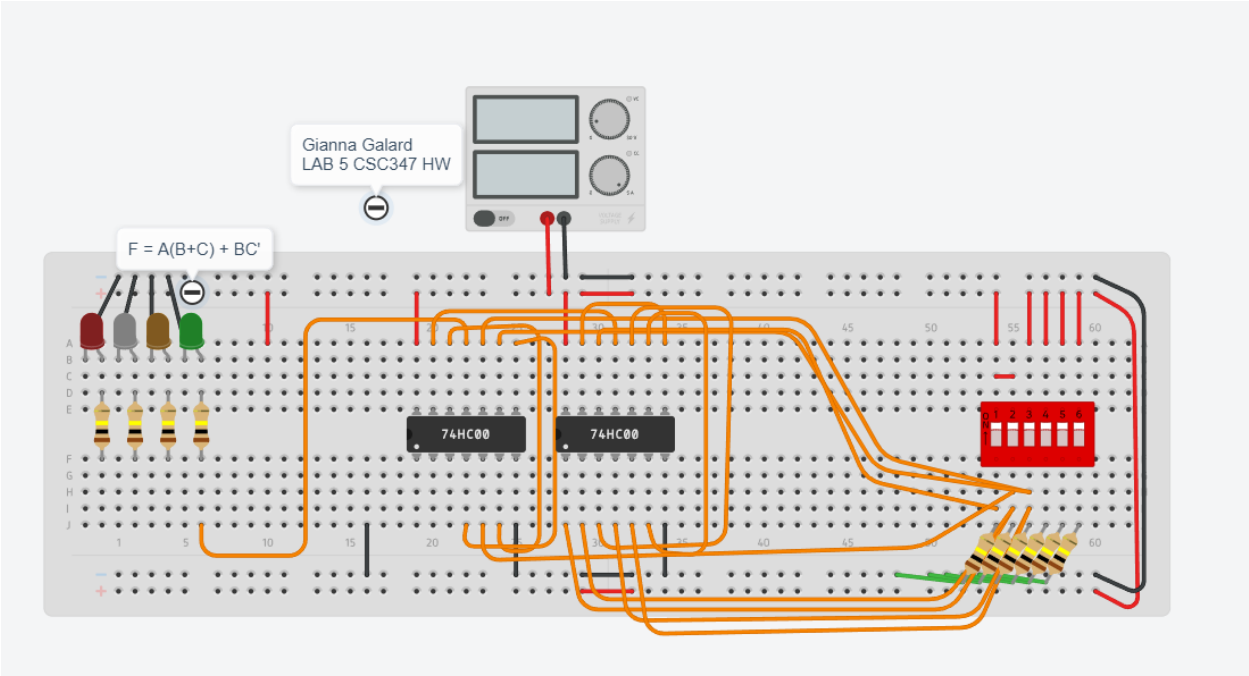
Conclusion:

After the student tested each input for the circuit, it shows that the expected outputs match the results, meaning that the conversion to only NAND and XOR gates is valid. By minimizing the number of gates used, we conserve more power throughout the board and the overall neatness of the circuit, making it a lot easier to read because of the number of gates used.

Homework:

Draw the equivalent purely NAND/NOR gate representation of the function below and build the circuits on Tinkercad using NAND and NORchips, respectively. DO NOT simplify the function or change the function to sum-of-product or product-of-sum format! In the lab report, include the logic diagram with NAND implementation, the diagram with NOR implementation, the Tinkercad screenshot and link.





LINKS

https://www.tinkercad.com/things/eq6w7fXgx1T-csc347-lab-5/editel?sharecode=0KrUvgmgMevdrX0fTiMiq0WrxkH_k2UlB9ccBMhAl-k

https://www.tinkercad.com/things/eq6w7fXgx1T-csc347-lab-5/editel?sharecode=0KrUvgmgMevdrX0fTiMiq0WrxkH_k2UlB9ccBMhAl-k