ITM 207: Circuit Diagram Conversions

EXPRESSION TO DIAGRAM, DIAGRAM TO EXPRESSION

> **Note:** Please ensure you are comfortable with the boolean operators and gates prior to studying this tip sheet. You can review those topics in the Boolean Basics Tip Sheet.

Expression to Diagram

Before we even begin drawing a circuit diagram from an expression, check the question to see if it requires you to simplify it first. Simplified expressions will produce the same outputs as the original if given the same inputs, but are shorter and will require less gates in a diagram. If you need help simplifying an expression, take a look at our Simplifying Expressions Tip Sheet.

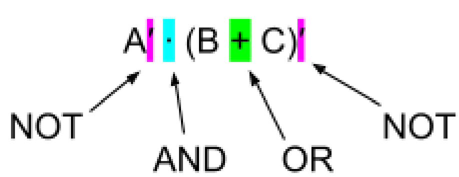
Step 1: Identify gates within the given expression

Wherever variables interact with another to change the output, there is a gate. Look for multiplication, addition, and the prime symbol since they represent **AND**, **OR**, and **NOT** respectively. Brackets with prime symbols may also be a **NAND** or **NOR** gate depending on the structure of the elements inside.

Let's analyze the following expression:

$$X = A'(B + C)'$$

We can see there are two prime symbols, multiplication between A′ and (B + C)′, and addition between B and C. Therefore, we could identify the gates like this:



> **Tip:** When working on circuit diagram practice questions, referring to a document with all the gates and their expressions can help you quickly identify and better memorize them. You can use the lecture slides, the textbook, our Boolean Basics Tip Sheet, or even your own notes!

Toronto Metropolitan University

Program Advising and Student Success

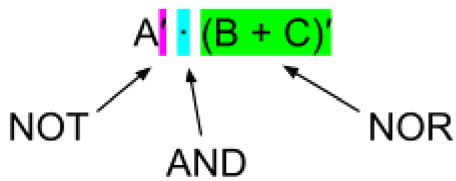
Ted Rogers School of Management, TRS 1-004
https://www.torontomu.ca/trsm-pass/

Ted Rogers School of Management | Program Advising and Student Success

However, if we compare this expression with the gates closely, we can actually see a similarity between the structure of the expression's latter half and the expression for the NOR gate:

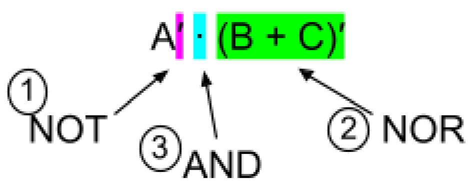
- NOR: (A + B)′
- Expression: (B + C)′

Given this new observation, we can revise our identified gates:



Step 2: Assign order to the gates based on the order of operations

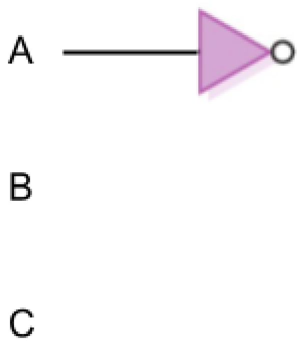
We have our gates, but we must figure out which ones to draw first. We can assign the drawing order using the order of operations. You can imagine the variables as numbers, and ask yourself which part of the expression to solve first. Since the prime symbol directly modifies the term it is attached to, it can be considered to have the same priority as exponents. Following the order of operations, we can order the gates like this:



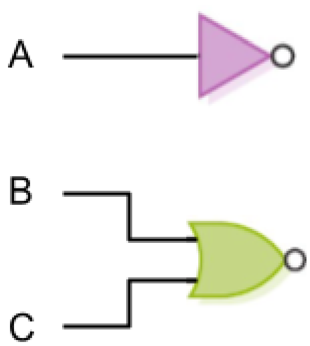
The order between the NOT and NOR gate in this case actually does not matter as long as they come before the AND. If you were to solve this expression algebraically, you could start by solving either section.

Step 3: List input variables and draw the diagram based on decided order

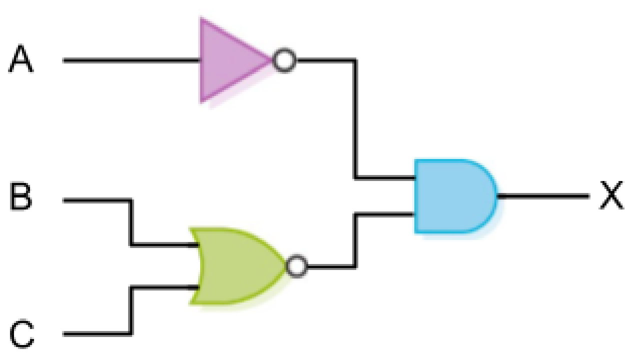
We are ready to draw our diagram! Let's list the input variables on the left and draw the NOT gate since it is first in the decided order. It is applied to the variable A:



Next is the NOR gate. It connects B and C:



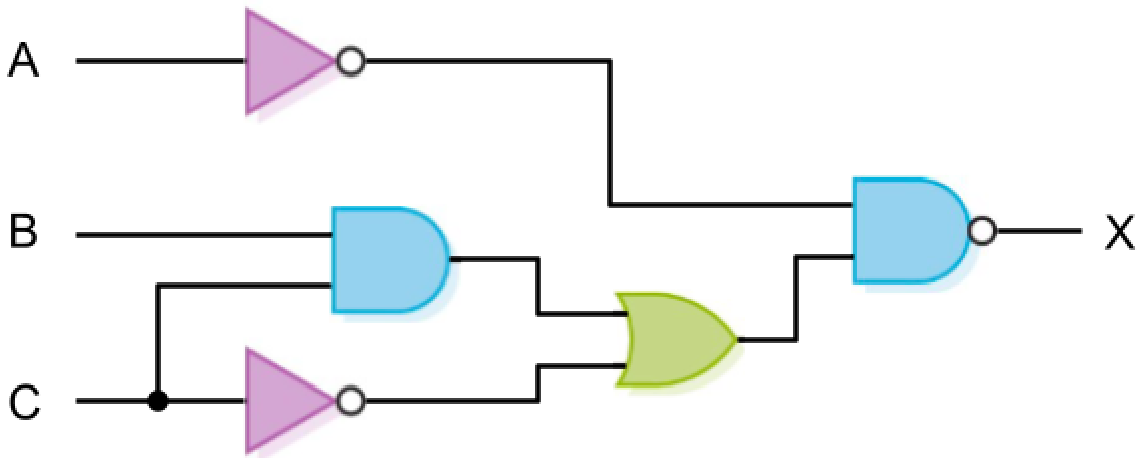
Lastly, the AND gate connects the NOR and NOR gate, leading to the output:



Notice how the NOT and NOR gate do not depend on inputs from one another. As mentioned earlier, you can start with drawing either due to the structure of the given expression.

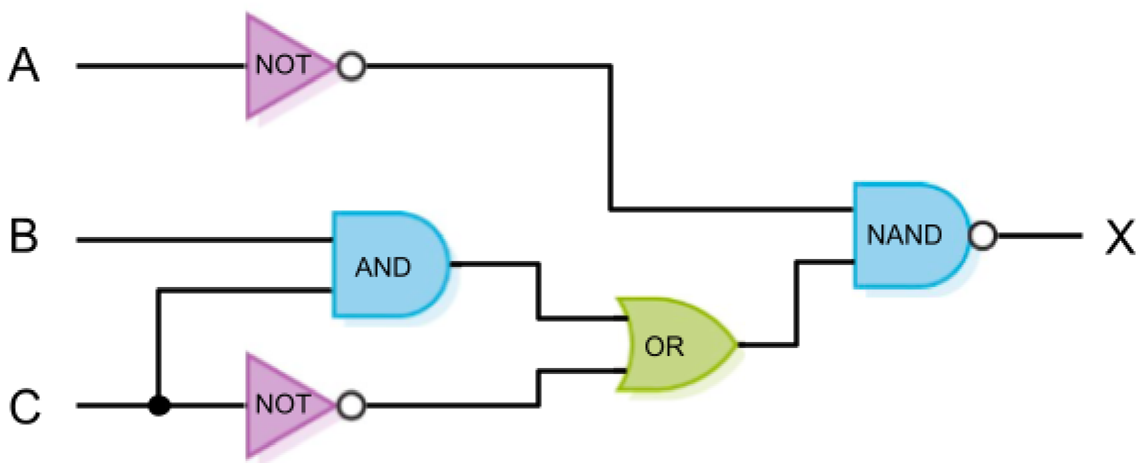
Diagram to Expression

Now let's try writing an expression for the following diagram:



Step 1: Identify the gate symbols

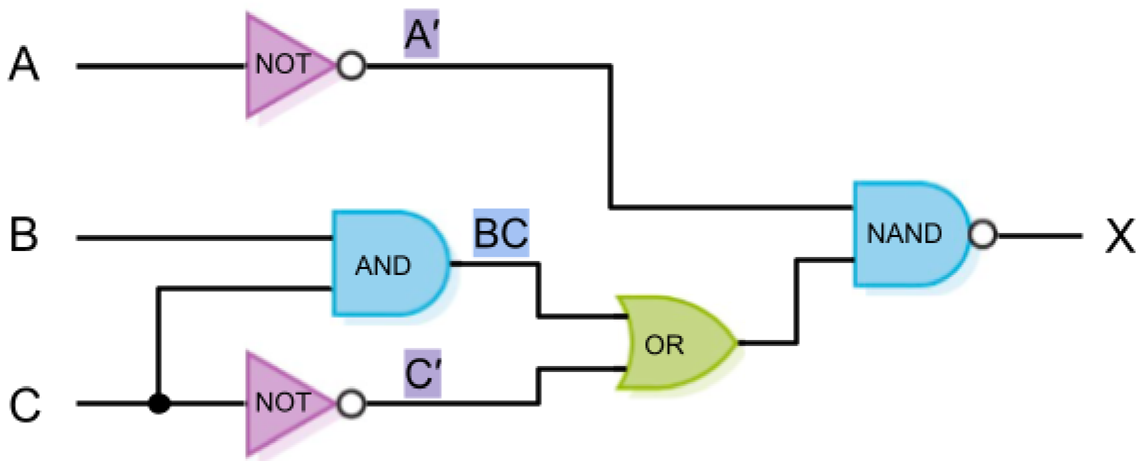
With enough practice and memorization, you will be able to tell what each gate is and does just by looking at the diagram. But for now, let's label them to make it clear:



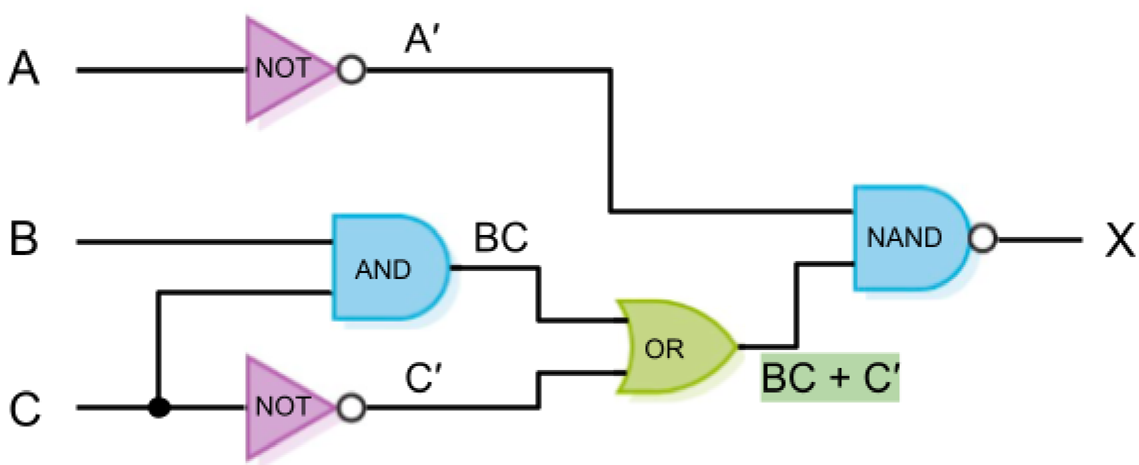
Step 2: Record the cumulative changes to the inputs made by the gates

Starting from the input variables, follow the lines and write down the altered input after each gate. Refer to the boolean operations and gate definitions to help complete this process.

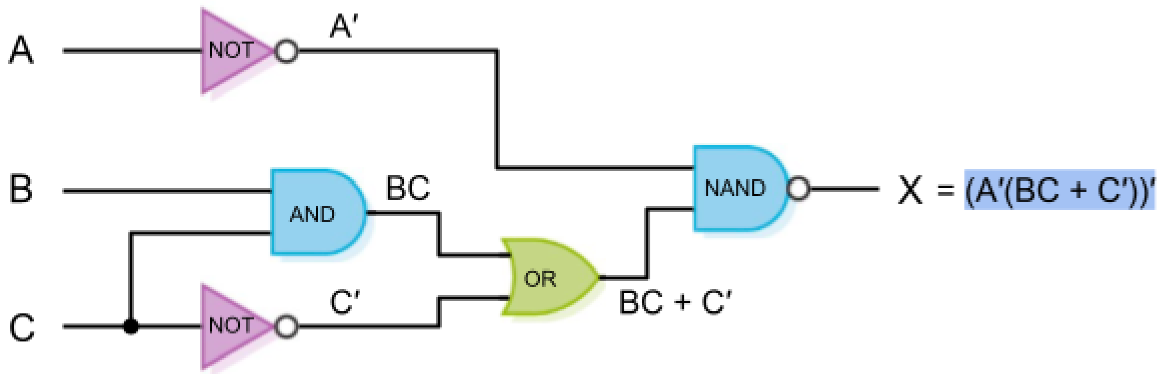
Firstly, we can see the two NOT gates operating on variables A and C. The AND gate combines signals from B and a split-off line from C. We can record the changed inputs like this:



Next is the OR gate. We can't do the NAND gate yet because we don't have all of its inputs. For the OR gate, look at the input signals leading to it and operate on them using the its function:



Lastly, combine the inputs for the NAND gate using its function to get the final expression:



After attaining the expression for a diagram, check the question to see if it requires you to simplify it. In our Simplifying Expressions Tip Sheet, we will look at how to simplify this expression along with some other tips!

Note: For conversions between circuit diagrams and truth tables, it is much easier to first convert to an expression then to the desired format. Refer to our Truth Table Conversions Tip Sheet to review how to perform conversions between truth tables and expressions!