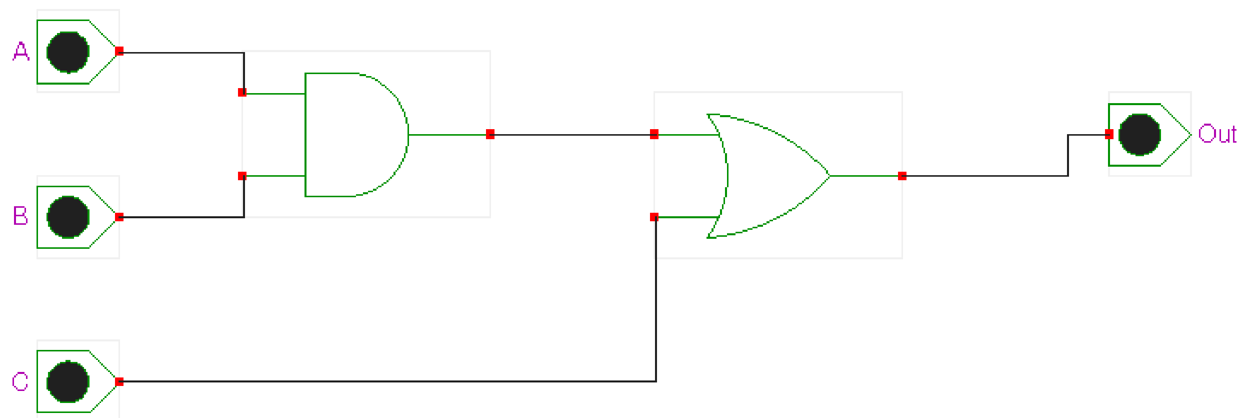
TECH 3232
Lecture Notes
9/10/2018

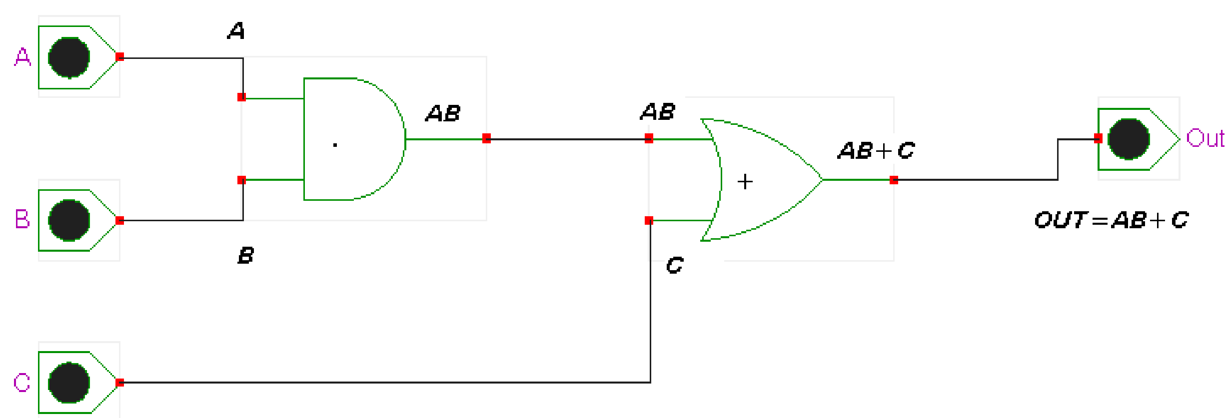
We saw in the last lecture how all the basic gates (AND, OR, NOT, XOR, NAND, NOR, etc) worked and also saw how the Truth Table, Boolean Algebra Expression and Timing Diagrams for each gate. But what if we want to combine multiple gates into a more complex circuit?

Below is an example of a more complex circuit:



To determine the Boolean equation:

Derive the equation for each gates output, starting with its inputs (so for the AND gate, the inputs are A and B so the output is AB. For the OR gate, the output is the first input (which is the equation for the AND gate's OUTPUT) OR C. So the Equation is OUT = AB + C.

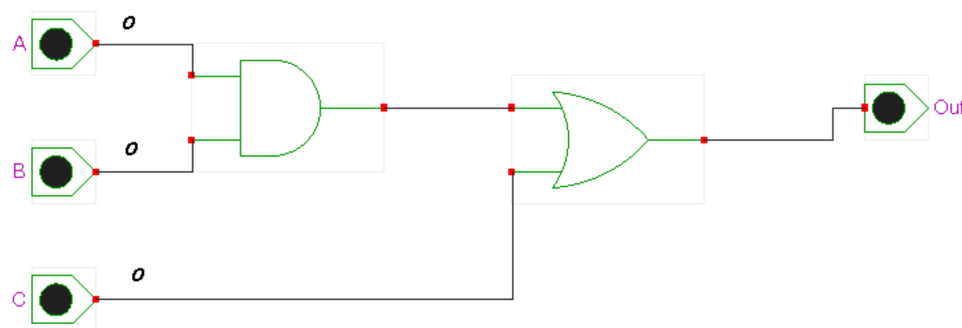


For the Truth Table, First fill in all the possible inputs (using a binary count for the order of inputs AND to insure you cover all possible input combinations). Since there are 3 inputs (A,B,C) you will need three columns. Start the count at 0 decimal or 000 Binary and count all the way to 7 decimal or 111 binary. The table will look like:

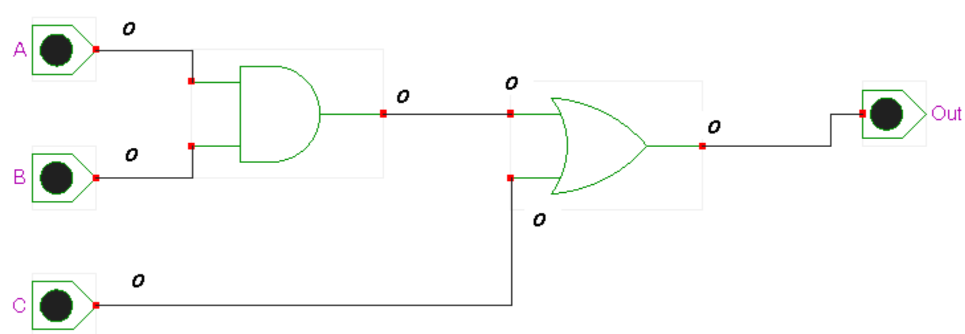
C	B	A	Out
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Note: the inputs can be placed in any order, but no matter what, the binary count should start at 0 and go to the highest value in a binary count.

You can fill in the truth table using various methods. The **<u>1st method</u>** is to take one row of inputs at a time, and place them on the schematic (for example row 1):



And start working though the circuit to derive the next output. So here, A=0 and B=0, since they are going through an AND gate, the output of the AND is 0, now that 0 and C (which is also zero) go into the OR gate. Since 0 OR 0 is 0 the Output is 0. So for row one, the output is 0.



We do this method for each row of the Truth Table until it is all filled in:

C	B	A	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

The 2nd method is faster, but takes a little bit of thinking. You would fill in the truth table's inputs as before, but now look at the circuit. Notice that input C is connected to one of the inputs of the OR gate. Since we know that anytime one of the inputs of an OR gate is one, the output will be one. So in the example circuit, anytime C=1 the Output will be 1. So we can fill in the output for every row where C=1 like this:

C	B	A	Out
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

So we were able to fill in ½ of the truth table by looking at the circuit and deducing the output of the OR gate. Now the other leg of the OR is connected to the AND gate's OUTPUT. We know that both inputs have to be 1 for the output to be 1. So we look for A and B both being a one. This occurs 2 times on the truth table:

C	B	A	Out
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

We have already figured out one of those outputs (since C was 1, the output was 1), for the combination of C=0, B=1 and A=1….since A and B are both ones, the output of the AND gate is 1, this is fed into the top input of the OR gate, making the output of the OR gate a 1 (and thus the circuits output is a 1. No other combination can create a 1 output of the AND gate and C=0 so the OR gate cannot turn on, so the rest of the T.T. is filled with zeros:

C	B	A	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Either method works, but I would suggest you use the 1st method until you get really good at it, then you can start looking for patterns and use method two.