3: Logic Circuits, Boolean Algebra, and Truth Tables - NOTES

TOPIC 1: Logic Representation

There are three common ways in which to represent logic.
1. Truth Tables
2. Logic Circuit Diagram
3. Boolean Expression

We will discuss each herein and demonstrate ways to convert between them.

TOPIC 2: Truth Tables

A truth table is a chart of 1s and 0s arranged to indicate the results (or outputs) of all possible inputs. The list of all possible inputs are arranged in columns on the left and the resulting outputs are listed in columns on the right. There are 2 to the power n possible states (or combination of inputs). For example with three inputs there are 2^3=8 possible combination of inputs.

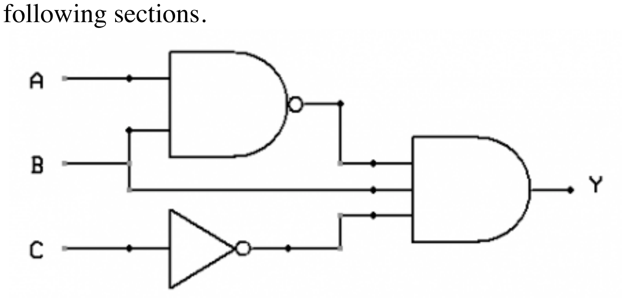
(inputs) (outputs)

all possible states

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

TOPIC 3: Logic Diagram

A logic diagram uses the pictoral description of logic gates in combination to represent a logic expression. An example below shows a logic diagram with three inputs (A, B, and C) and one output (Y). The interpretation of this will become clear in the following sections.



TOPIC 4: Boolean Expression

Boolean Algebra can be used to write a logic expression in equation form. There are a few symbols that you'll recognize but need to redefine.

+ means OR, so A+B is the same as A OR B.

* means AND, so AB is the same as A AND B (the * symbol isn't normal written but understood)

$\overline{A}$ means NOT so $\overline{A}$ means NOT A (sometimes !A will be used for convenience)

⊕ means XOR so A⊕B means A XOR B

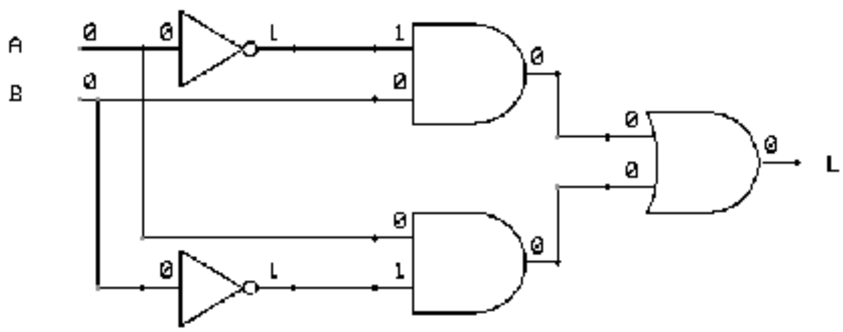
Note: Sometimes when the ! is used to represent the NOT it is used before the letter and sometimes it is used after the letter. Care should be used so that you understand which method is being used!

Below is an example boolean expression. In fact, it represents the same logic as the example logic circuit diagram above. This concept will also become clearer when we cover converting from and to the boolean expression below.

$$Y = \left(\overline{AB}\right)B\overline{C}$$

TOPIC 5: Converting from a Logic Circuit Diagram to a Truth Table

This conversion is accomplished by selecting each state (or combination of inputs) one at a time, replacing the inputs with their respective values and figuring the value of each point through the circuit until the output is reached. The final output value for each state is then listed in the truth table next to the value of each input. Below is a logic circuit diagram with the input values. Study it carefully for an extended period of time, it is an animated image and the inputs and output will change every few seconds.



Below are the results of the conversion in truth table form.

Inputs		Output
A	B	L
0	0	0
0	1	1
1	0	1
1	1	0

TOPIC 6: Converting Logic Circuit Diagrams to Boolean Expressions

To convert from a logic circuit diagram to a boolean expression we start by listing our inputs at the correct place and process the inputs through the gates, one gate at a time, writing the result at each gate's output. The following is the resulting Boolean expression of each of the gates.

AND $Y = AB$

OR $Y = A + B$

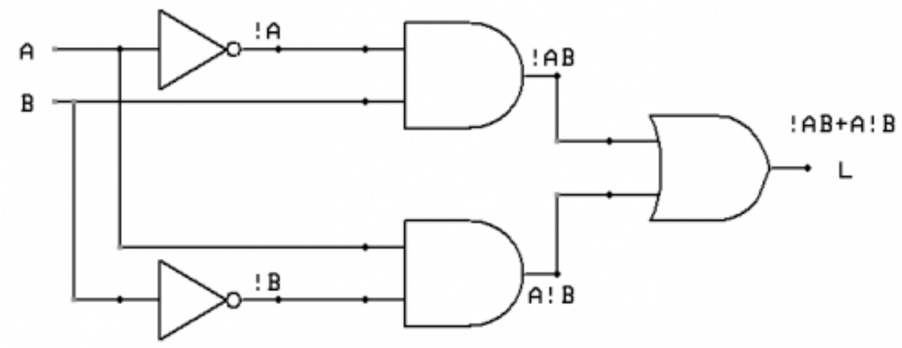
NOT $Y = \overline{A}$

NOR $Y = \overline{A + B}$

NAND $Y = \overline{AB}$

XOR $Y = A \oplus B$

And here is an example of the process being carried out. The fact that the result simplifies to the XOR is merely coincidental.



$$L = \overline{A}B + A\overline{B} = A \oplus B$$

TOPIC 7: Converting Truth Tables to Boolean Expressions

There are two methods for converting truth tables to boolean expressions.

Sum of Products Method

A	B	C	Y	
0	0	0	0	
0	0	1	0	
0	1	0	1	→ A!BC!
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	→ ABC

ABC + A!BC! = Y

Product of Sums Method

A	B	C	Y	
0	0	0	1	
0	0	1	1	
0	1	0	1	
0	1	1	0	→ A + B! + C!
1	0	0	0	→ A! + B + C
1	0	1	1	
1	1	0	1	
1	1	1	1	

(A + B! + C!)(A! + B + C) = Y

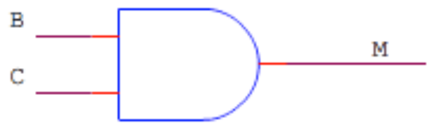
TOPIC 8: Converting Boolean Expressions to Logic Diagrams

Converting boolean expressions to logic diagrams is the most challenging conversion on this page because it requires a very good understanding of order of operation. Below is the order of operations used in this conversion.

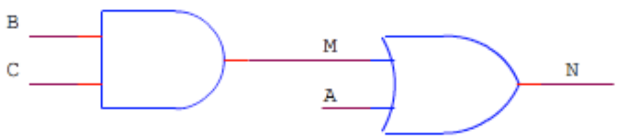
1. Bracketed quantities first
2. NOTs next
3. ANDs next
4. ORs last

In order to complete this conversion we will progress through the order of operations. We will first look for bracketed quantities or something in parentheses. Inside any parentheses we will look for more parentheses and then NOTs, then ANDs, then ORs. It's best to begin with an example.

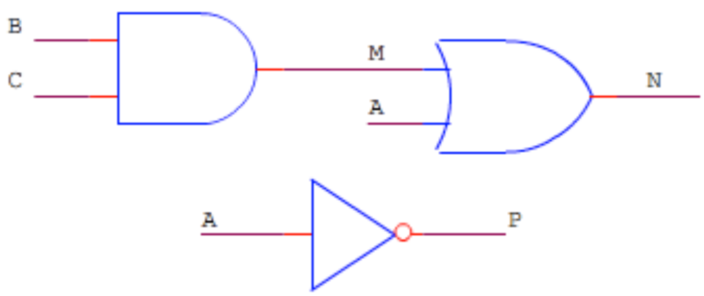
Example: (BC + A)(A! + C) = Y



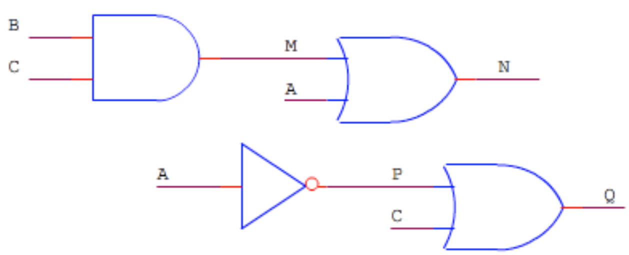
Example: (M+ A)(A! + C) = Y



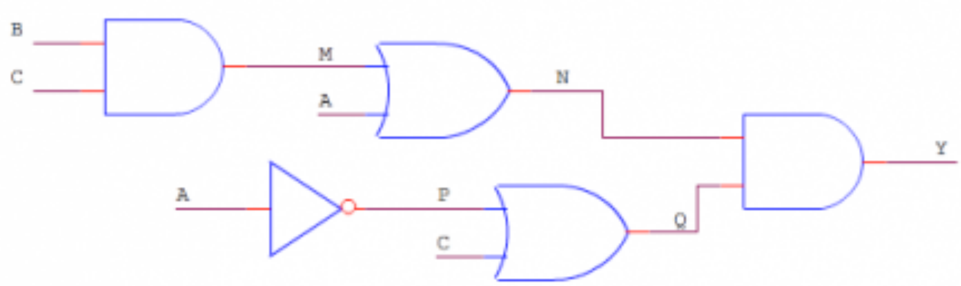
Example: N(A! + C) = Y



Example: N(P + C) = Y



Example: NQ = Y



TOPIC 9: Converting a Truth Table to a Logic Diagram

The easiest way to accomplish this is to first convert the truth table to a boolean expression and then to a logic diagram.

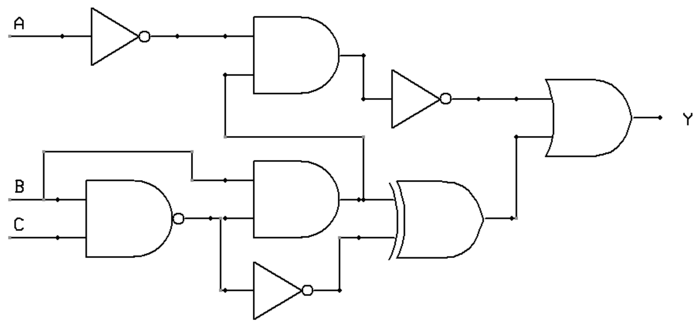
You should now be prepared to answer the following questions.

1. A logic system has 5 inputs. How many possible states exist in this system?

2. What symbol is used to represent the NOT gate when the line over the letter is not convenient to use?

3. A logic system has 3 inputs and therefore 8 possible states. The logic diagram representation is shown below. Complete the truth table and convert the output column to hexadecimal if the state 0 is the least significant bit and the state 7 is the most significant bit.

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



4. Give the boolean expression from the above circuit diagram.

5. A truth table has the same states as in number 3 above. However, the output column from top to bottom reads 00110101. Give the result of the sum of products method.

6. Give the result of the product of sums method in number 5 above.

7. A boolean expression is given Y = (A+B)C+!BA+!C(A+B)+!(AC). Just as in 3 above, produce a truth table and convert the output column to hexadecimal.