

EE334
Binary Logic

- **NOT:** The not gate is simplest of the logic operations. The not function takes a signal and outputs its opposite. The truth table and circuit symbol for NOT is shown below. A **truth table** relates the inputs of a logic function to its outputs. NOT always has one input and one output.

NOT	
In	Out
0	1
1	0

- **AND:** The English rule for the AND function is “output a 1 if and only if all inputs are a 1, otherwise output a 0.” The AND function must have two or more inputs. Complete the AND truth table and draw the circuit symbol below.

AND		
A	B	Out
0	0	
0	1	
1	0	
1	1	

- **OR:** In English the rule for the OR function would be something like this, “output a 1 whenever any input is a 1, otherwise output a 0.” The OR function must have two or more inputs. Complete the OR truth table and draw the circuit symbol below.

OR		
A	B	Out
0	0	
0	1	
1	0	
1	1	

*** A circuit that implements a logic function, as well as the schematic symbol for that circuit is usually called a **“gate”**. Any idea why “gate” is an appropriate term?

Compound Functions (Non-basic Functions)

The functions NOR, NAND, XOR, and XNOR are combinations of the NOT, AND, and OR functions.

- **NOR:** NOR is short for NOT OR. You may think of a NOR function as an OR but with all of the outputs inverted (flipped). You can actually build a NOR circuit by connecting an OR circuit’s output and connecting to the input of a NOT. The NOR function outputs a 1 exactly

EE 334
Binary Arithmetic and Logic Expressions

when all the inputs are 0, otherwise it returns a 0. Complete the NOR truth table and draw the circuit symbol for NOR below.

NOR		
A	B	Out
0	0	
0	1	
1	0	
1	1	

- **NAND:** The NAND operation is an AND followed by a NOT (NOT AND or NAND). The NAND function outputs a 1 whenever all inputs are not 1s. Complete the NAND truth table and draw the circuit symbol for NAND below.

NAND		
A	B	Out
0	0	
0	1	
1	0	
1	1	

- **XOR:** The XOR operation is special case of OR. In fact, XOR stands for “exclusively OR.” In plain English, the XOR function for two or more inputs is “Output a 1 whenever there are an odd number of 1s in the input, otherwise output 0.” For the special case of two inputs, the XOR description is “Output a 1 when either input is a 1, but not both inputs are a 1.” Complete the XOR truth table and draw the circuit symbol for XOR below.

XOR		
A	B	Out
0	0	
0	1	
1	0	
1	1	

- **XNOR:** As you might imagine by now, XNOR is short for “NOT XOR”. XNOR is simply the XOR operation with the outputs subjected to a NOT. Complete the XNOR truth table and draw the circuit symbol for XNOR below.

XNOR		
A	B	Out
0	0	
0	1	
1	0	
1	1	

EE 334
Binary Arithmetic and Logic Expressions

14. Draw circuits for NAND and NOR using only NOT, AND, and OR gates.

15. Complete the following multi-function truth table. Note that there are three inputs.

A	B	C	NAND	NOR	XOR	XNOR
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

14. **Tricky!** Create an XOR circuit out of AND, OR, and NOT gates.

Boolean Logic Expressions

As we have seen, binary digits can be used in logic operations (AND, OR, NOT, etc.) Now we'll go a bit further and look at Boolean algebra and logic expressions.

Boolean algebra, developed by George Boole in the 1840s, is a way of writing logical expressions. There are three basic Boolean algebra operators: AND, OR, and NOT. Boolean algebra also uses parenthesis to group and order operations. Like regular algebra, Boolean algebra uses symbols (like x , y , z or A , B , C) and values (0 and 1 or True and False) in the expressions. Here are the common symbols for the Boolean operators along with some examples.

AND

- dot “.” Ex., $A \cdot B$
- hat: “ \wedge ” Ex., $A \wedge B$
- nothing Ex., AB

OR

- plus “+” Ex. $A+B$
- vee “ \vee ” Ex., $A \vee B$

EE 334
Binary Arithmetic and Logic Expressions

NOT

- negative “-“ Ex., $-A$
- hook “ \neg ” Ex. $\neg A$
- squiggle “ \sim ” Ex. $\sim A$
- bang “!” Ex. $!A$
- overbar Ex. \bar{A}

The order of precedence for Boolean algebra is (for this class)

- a. parentheses
- b. NOT
- c. AND
- d. OR

The important thing about Boolean expressions (for this class at least) is that they evaluate to a single logical value (0/1 or T/F) depending on the values of the expression’s variables and constants. Let’s check it out.

11. What is the logical value of $AB+C$ if $A = 1$, $B = 0$, and $C = 0$?

12. What is the logical value of $(W+Y)(X+\sim Z)$ if $W = 0$, $X = 0$, $Y = 1$, and $Z = 0$?

Truth Tables

Truth tables are very handy for visualizing all possibilities for a logic expression. A truth table shows all possible combinations of the variables and then shows the expression’s value for each combination. Sometimes the truth table will work up to the expression’s value by showing sub-expression. This is done so that humans can track the values more easily.

13. Complete the truth table below for $A\sim B$

A	B	$\sim B$	$A\sim B$

14. Create a truth table for $(AB+C) + \sim(AC)$