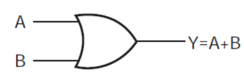
PHY 307: Electronics-II

Tutorial-3

Logic gates, Boolean algebra

Logic gates: The logic gate is the most basic building block of any digital system, e.g., mobile phones, computers, etc.. Each one of the basic logic gates is a piece of an electronic circuit (hardware), which implements basic logic expression. The laws of Boolean algebra could be used to do manipulation with binary variables and simplify logic expressions. The three basic logic gates are the OR gate, the AND gate and the NOT gate.

OR gate: An OR gate performs an ORing operation on two or more than two logic variables. The OR operation on two independent logic variables A and B is written as $Y = A+B$ and reads as Y equals A OR B and not as A plus B. The output of an OR gate is HIGH when any of its inputs is HIGH.

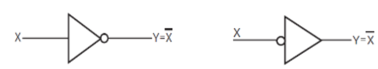


An OR gate.

AND gate: An AND gate is a logic circuit having two or more inputs and one output. The output of an AND gate is HIGH only when all of its inputs are in the HIGH state.

An AND gate.

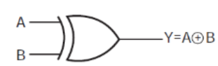
NOT gate: A NOT gate is a one-input, one-output logic circuit whose output is always the complement of the input, i.e., a LOW input produces a HIGH output, and vice versa.



A NOT gate

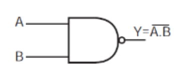
EXCLUSIVE-OR Gate: The EXCLUSIVE-OR (EX-OR) gate, is a two-input, one-output gate. The output of an EX-OR gate is a logic '1' when the inputs are unlike and a logic '0' when the inputs are like.

$$Y = A'\oplus B' = A'B+AB'$$



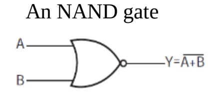
An EX-OR gate

NAND Gate: NAND stands for NOT AND. An AND gate followed by a NOT circuit makes it a NAND gate. $Y= (A.B)'$



An NAND gate

NOR Gate: NOR stands for NOT OR. An OR gate followed by a NO T circuit makes it a NOR gate. $Y =(A+B)'$



A NOR gate

Boolean algebra: In 1849, a British mathematician, George Boole presented an algebraic formulation of the processes of logic thoughts and reasoning, called Boolean Algebra. The basic description of the Boolean algebra is based on concepts from set theory. A Boolean algebra is a closed algebraic system containing a set P of two or more elements and two operations, AND (.) and OR (+). For every, a and b in set P, $(a.b)$ and $(a+b)$ belong to P.

The symbolic Boolean algebra can be practiced through the relay and switching circuits following that there are only two possible states for switches, open and closed, a situation reminiscent of Boole's special algebra on two symbols, 0 and 1.

Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

PHY 307: Electronics-II

Tutorial-3

Logic gates, Boolean algebra

1. **Boolean algebra:** Simplify the following expression:

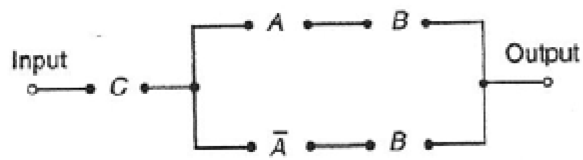
   (i). A.B.C'.D + A.B'.C'.D + A'.B.C.D + A.B'.C.D + A'.B.C.D'

   (ii). (A'.B+B.C')'.(A.B')'

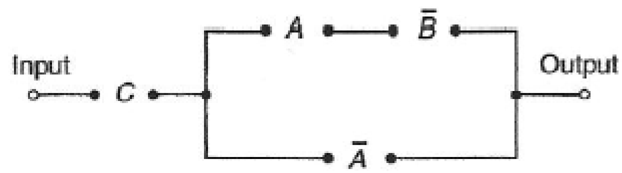
   (iii). [(A.B')'+(B.C')'+(A'.B)'

2. **Switching circuits to Boolean expression:** Determine the Boolean expression and construct a truth table for the switching circuits shown below,

(i).



(ii).



3. **Boolean expression to switching circuits:**

   (i). *A*.(*A*.B'.*C* + *B*.(*A* + C'))

   (ii). *A*.*B*.*C*.(*A* + *B* + *C*)

4. **EX-OR to NOT Gate:** How can you implement a NOT circuit using a two-input EX-OR gate?