VIBEKA NAND JHA

ASSOCIATE PROFESSOR

DEPARTMENT OF PHYSICS

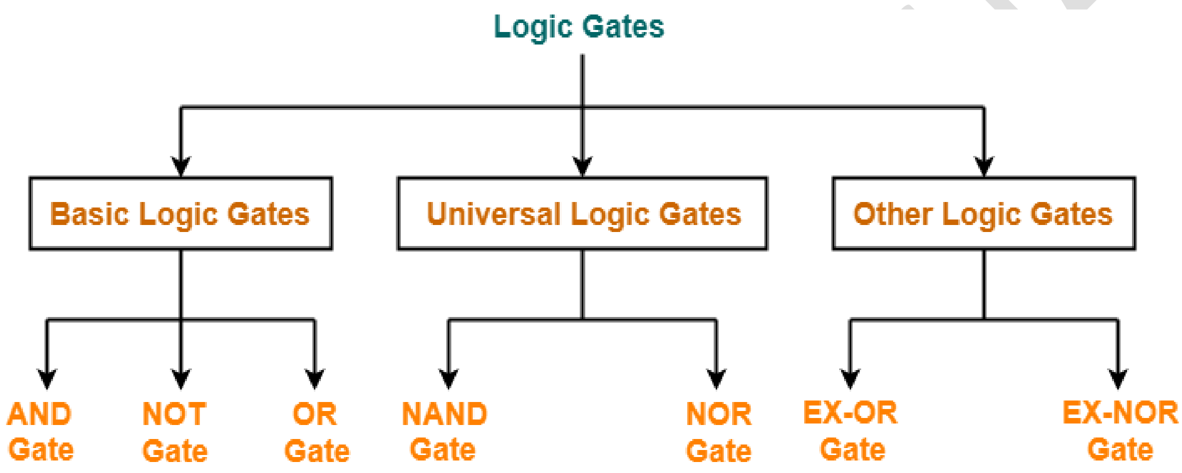
SNSRKS COLLEGE SAHARSA

PHYSICS (H)-PAPER VI

LOGIC GATES (CIRCUITS)

In 1854, **George Boole** performed an investigation into the “laws of thought” which were based around a simplified version of the “group” or “set theory”, and from this Boolean algebra was developed. **Boolean algebra** deals mainly with the theory that both logic and set operations are either “TRUE” or “FALSE” but not both at the same time.

A logic gate is a physical device which performs logic operation on one or more inputs and produces a single logical output. The relationship between the input and the output is based on certain logic. Based on this, logic gates are named as AND gate, OR gate, NOT gate etc. We can classify these Logic gates into the following three categories:



TRUTH TABLE:

The table used to represent the boolean expression of a logic gate function is commonly called a **Truth Table**. A logic gate truth table shows each possible input combination to the gate or circuit with the resultant output depending upon the combination of these input(s).

For example, consider a single **2-input** logic circuit with input variables labelled as A and B. There are “four” possible input combinations or 2^2 of “OFF” and “ON” for the two inputs. However, when dealing with Boolean expressions and especially logic gate truth tables, we do not general use “ON” or “OFF” but instead give them bit values which represent a logic level “1” or a logic level “0” respectively.

Then the four possible combinations of A and B for a 2-input logic gate is given as:

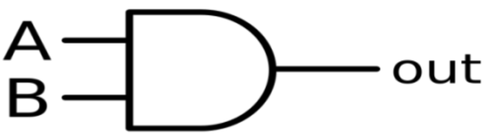
- Input Combination 1. – “OFF” – “OFF” or (0, 0)
- Input Combination 2. – “OFF” – “ON” or (0, 1)
- Input Combination 3. – “ON” – “OFF” or (1, 0)
- Input Combination 4. – “ON” – “ON” or (1, 1)

STANDARD/ BASIC LOGIC GATES

AND, NOT and OR gates are the basic gates; we can create any logic gate or any Boolean expression by combining a mixture of these gates.

1. AND GATE: An AND gate has n inputs and one output. The output of AND gate is high (1) only when input A and input B are both high (1). The logic symbol and truth table for this gate are:

Logic Symbol



If A and B are inputs of AND gate then the output

Y=A.B

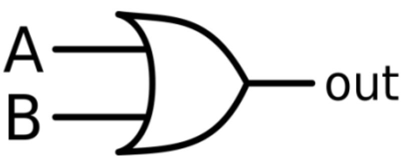
Truth Table

Input		Output
A	B	F = A.B
0	0	0
0	1	0
1	0	0
1	1	1

- It follows the associative law.
- It follows the commutative law.

2. OR GATE: An OR gate has n inputs and one output. The output of OR gate is high(1) when either input A or input B or both are 1s, that is, if any of the input is high, the output is high. . The logic symbol and truth table for this gate are:

Logic Symbol



If A and B are inputs of OR gate then the output

Y=A+B

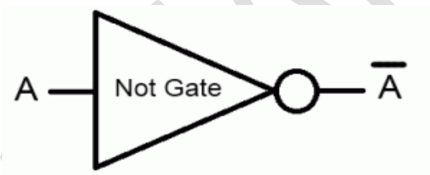
Truth Table

Input		Output
A	B	F = A+B
0	0	0
0	1	1
1	0	1
1	1	1

- It follows the associative law.
- It follows the commutative law.

3. NOT GATE: This is the most basic gate, with one input and one output. It produces a 'high (1)' output if the input is 'low (0)' and vice-versa. It produces an inverted version of the input at its output. This is why it is also known as an inverter. The logic symbol and truth table for this gate are:

Symbol



If A is the input of NOT gate then the output

Y = A'

Truth Table

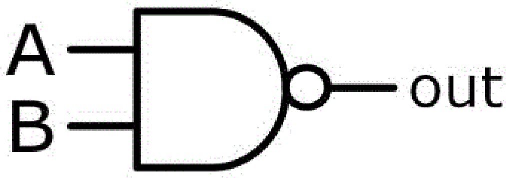
A	NOT A
0	1
1	0

UNIVERSAL LOGIC GATES

A universal logic gate is a logic gate that can be used to construct all other logic gates; we can create any logical Boolean expression using ONLY NOR gates or ONLY NAND gates.

1. NAND GATE: This is an AND gate followed by a NOT gate. The gate gets its name from this NOT AND behaviour. If inputs A and B are both 'high(1)', the output Y is not 'high(1)'. It is also called as bubbled OR. The logic symbol and truth table for this gate are:

Symbol



If A and B are inputs of NAND gate then the output

Y= (A.B)'

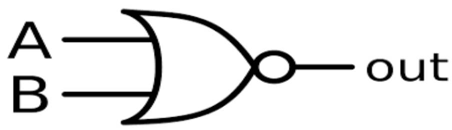
Truth Table

A	B	A **NAND** B
0	0	1
0	1	1
1	0	1
1	1	0

- It doesn't follow the associative law.
- It follows the commutative law.

2. NOR GATE: A NOT- operation applied after OR gate gives a NOT-OR gate (or simply NOR gate). Its output is 'high (1)' only when both inputs A and B are 'low(0)'. It is also called as bubbled AND. The logic symbol and truth table for this gate are:

Symbol



If A and B are inputs of NOR gate then the output

Y= (A+B)'

Truth Table

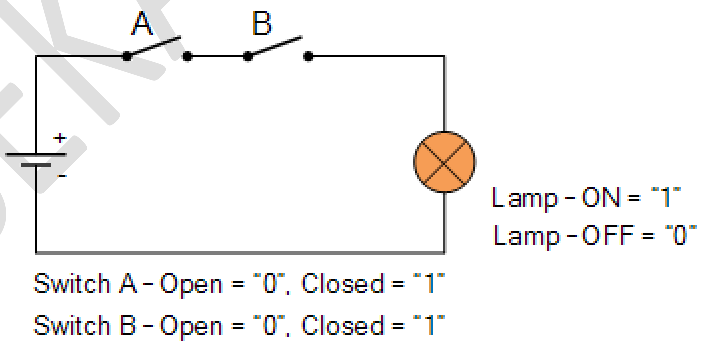
A	B	A **NOR** B
0	0	1
0	1	0
1	0	0
1	1	0

- It doesn't follow the associative law.
- It follows the commutative law.

SWITCHING CIRCUIT

Boolean Algebra is a simple and effective way of representing the switching action of standard Logic Gates and the basic logic statements which concern us here are given by the logic gate operations of the AND, the OR and the NOT gate functions.

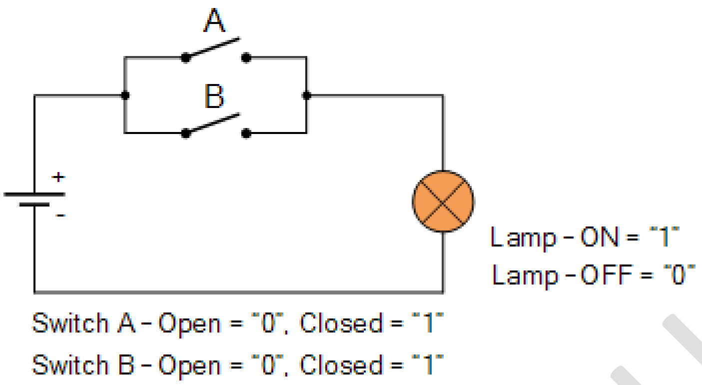
1. Switch Representation of the AND Function



Here the two switches, A and B are connected together to form a series circuit. As there are only two Switches, each with two possible states "open" or "closed". Defining Logic "0" as being when the switch is open and Logic "1" when the switch is closed.

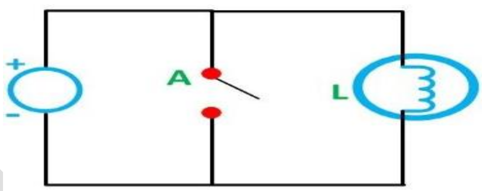
Therefore, in the circuit above, both switch A AND switch B must be closed (Logic "1") in order to put the lamp on. In other words, both switches must be closed, or at logic "1" for the lamp to be "ON". In Boolean algebra terms the output will be TRUE only when all of its inputs are TRUE.

2. Switch Representation of the OR Function



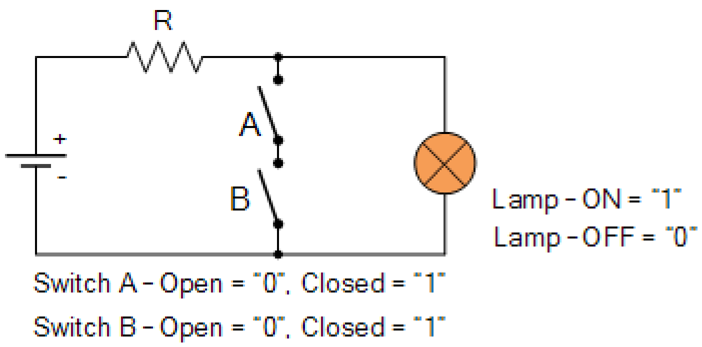
Here the two switches A and B are connected in parallel and either switch A **OR** switch B can be closed in order to put the lamp on. In other words, either switch can be closed, or at logic "1" for the lamp to be "ON". Defining Logic "0" as being when the switch is open and Logic "1" when the switch is closed. In Boolean algebra terms the output will be TRUE when any of its inputs are TRUE.

3. Switch Representation of the NOT Function



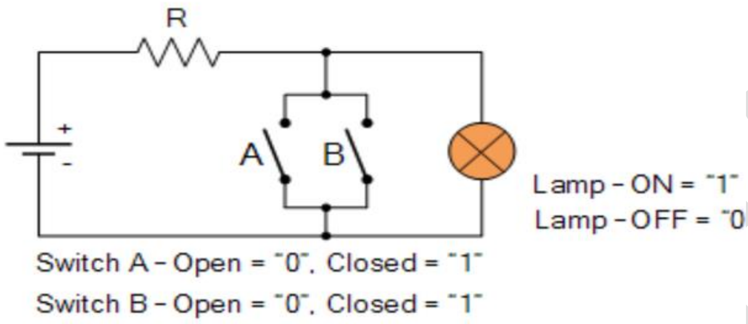
As NOT gates perform the logic INVERT or COMPLEMENTATION function they are more commonly known as Inverters because they invert the signal. In logic circuits this negation can be represented by a normally closed switch. If A means that the switch is closed, then NOT A or simply A' says that the switch is NOT closed or in other words, it is open. The logic NOT function has a single input and a single output as shown above.

4. Switch Representation of the NAND Function



The truth table for the NAND function is the opposite of that for the previous AND function because the NAND gate performs the reverse operation of the AND gate. In other words, the NAND gate is the complement of the basic AND gate.

5. Switch Representation of the NOR Function



The NOR function is the opposite of that for the previous OR function because the NOR gate performs the reverse operation of the OR gate. Here we can see that the NOR gate is the complement of the OR gate.

EXCERCISE

1. The output of an AND gate with three inputs, A, B, and C is HIGH when ________.

   a) A = 1, B = 1, C = 0

   b) A = 0, B = 0, C = 0

   c) A = 1, B = 1, C = 1

   d) A = 1, B = 0, C = 1

1. The output of a NOR gate is HIGH when________

   a) All inputs are HIGH

   b) Any input is HIGH

   c) All inputs are LOW

   d) Any input is LOW

2. The format used to present the logic output for the various combinations of logic inputs to a gate is called as-

   a) Boolean variable

   b) Input logic function

   c) Truth table

d) Boolean constant

3. The Boolean expression for a 3-input AND gate is ________

a) X = ABC

b) X = A + B + C

c) X = AB + C

d) X = AB

5. If the input to a NOT gate is A and the output is X, then _______

a) X=A’

b) X = 0

c) X = A

d) None of the above

REFERENCE

[1] www.electronics-tutorials.ws/boolean/bool_7.html

[2] en.wikipedia.org/wiki/Logic_gate