Digital Electronics

Boolean Algebra & Logic Gates

Logic Gates

Logic gates are electronic circuits that can be used to implement the most elementary logic expressions, which are also known as Boolean expressions. The logic gate is the most basic building block of combinational logic. There are three basic logic gates, namely the OR gate, the AND gate and the NOT gate. Other logic gates that are derived from these basic gates are the NAND gate, the NOR gate, the EXCLUSIVE-OR gate and the EXCLUSIVE-NOR gate. This chapter deals with logic gates and implementations using NAND and NOR gates followed by simplification of Boolean functions using Boolean Laws and theorems and using K-maps.

Positive and Negative Logic

The binary variables can have either of the two states, i.e., the logic '0' state or the logic '1' state. These logic states in digital systems such as computers, for instance, are represented by two different voltage levels or two different current levels. If the more positive of the two voltage or current levels represents a logic '1' and the less positive of the two levels represents a logic '0', then the logic system is referred to as a *positive logic system*. If the more positive of the two voltage or current levels represents a logic '0' and the less positive of the two levels represents a logic '1', then the logic system is referred to as a *negative logic system*.

If the two voltage levels are 0V and +5V, then in the positive logic system the 0V represents logic '0' and the +5V represents logic '1'. In the negative logic system, 0V represents logic '1' and 5V represents logic '0'. If the two voltage levels are 0V and −5V, then in the positive logic system the 0V represents a logic '1' and the

−5V represents a logic '0'. In the negative logic system, 0V represents logic '0' and −5V represents logic '1'.

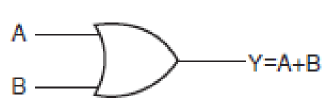
Logic Gates

The logic gate is the most basic building block of any digital system, including computers. Each one of the basic logic gates is a piece of hardware or an electronic circuit that can be used to implement some basic logic expression. While laws of Boolean algebra could be used to do manipulation with binary variables and simplify logic expressions, these are actually implemented in a digital system with the help of electronic circuits called logic gates. The three basic logic gates are the OR gate, the AND gate and the NOT gate.

OR Gate

A logic gate used to perform the operation of ***logical addition*** is called an OR gate. An OR gate performs an *OR* 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. An OR gate is a logic circuit with two or more inputs and one output. The output of an OR gate is LOW only when all of its inputs are LOW. For all other possible input combinations, the output is HIGH. A truth table lists all possible combinations of input binary variables and the corresponding outputs of a logic system. Figure shows the circuit symbol and the truth table of a two-input OR gate. The operation of a two-input OR gate is explained by the logic expression

$$Y = A+B$$



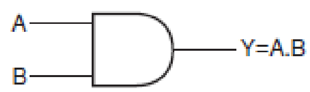
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Two input OR Gate

AND Gate

A logic gate used to perform ***logical multiplication*** is known as 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. In all other cases, the output is LOW. The logic symbol and truth table of a two-input AND gate is shown in figure. The AND operation on two independent logic variables A and B is written as Y = A.B and reads as Y equals A AND B. The operation of a two-input AND gate is explained by the logic expression

$$Y = A.B$$



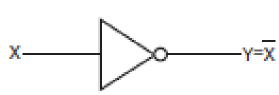
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Two input AND Gate

NOT Gate

A logic gate used to perform ***logical inversion*** is known as a NOT gate. A NOT gate is a one-input, one-output logic circuit whose output is always the complement of the input. That is, a LOW input produces a HIGH output, and vice versa. If X is the input to a NOT circuit, then its output Y is given by Y = $\overline{X}$ or X' and reads as Y equals NOT X. The logic symbol and truth table of a NOT gate is shown in figure. The operation of a NOT gate is explained by the logic expression

$$Y = \overline{X}$$



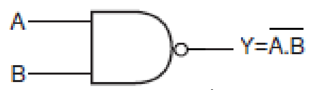
X	Y
0	1
1	0

NOT Gate

NAND Gate

NAND stands for NOT AND. An AND gate followed by a NOT circuit makes it a NAND gate. The output of a NAND gate is logic '0' when all its inputs are logic '1'. For all other input combinations, the output is logic '1'. The symbol and truth table of a NAND gate is as shown. NAND gate operation is logically expressed as

$$Y = \overline{AB}$$



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

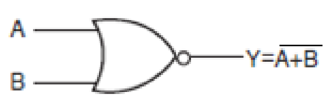
Two input NAND Gate

NAND Gate is known as Universal gate as it can be used alone to implement any gate operation. Hence it is said to be functionally complete.

NOR Gate

NOR stands for NOT OR. An OR gate followed by a NOT circuit makes it a NOR gate. The output of a NOR gate is logic '1' when all its inputs are logic '0'. For all other input combinations, the output is logic '0'. The symbol and truth table of a NOR gate is as shown. The output of a two-input NOR gate is logically expressed as

$$Y = (\overline{A + B})$$



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

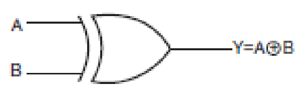
Two input NOR Gate

NOR gate is also known as Universal gate as it is used alone to implement any gate operation and hence it is also functionally complete.

EXCLUSIVE-OR Gate

The EXCLUSIVE-OR gate, commonly written as EX-OR gate, is a two-input, one-output gate. The output of an EX-OR gate is logic ‘1’ when the inputs are unlike and logic ‘0’ when the inputs are like. Although EX-OR gates are available in integrated circuit form only as two-input gates, unlike other gates which are available in multiple inputs also, multiple-input EX-OR logic functions can be implemented using more than one two-input gates. The output of a multiple-input EX-OR logic function is logic ‘1’ when the number of 1s in the input sequence is odd and logic ‘0’ when the number of 1s in the input sequence is even, including zero. The symbol and truth table of an EX-OR gate is shown in figure. The output of a two-input EX-OR gate is logically expressed as

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



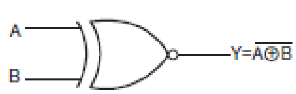
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Two input EX-OR Gate

EXCLUSIVE-NOR Gate

EXCLUSIVE-NOR, commonly written as EX-NOR, means NOT of EX-OR, i.e., the logic gate that we get by complementing the output of an EX-OR gate. The truth table of an EX-NOR gate is obtained from the truth table of an EX-OR gate by complementing the output entries as shown in figure. Logically,

$$Y = \overline{A \oplus B} = A'B' + AB$$



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

Two input EX-NOR Gate

The output of a two-input EX-NOR gate is logic ‘1’ when the inputs are like and logic ‘0’ when they are unlike. In general, the output of a multiple-input EX-NOR logic function is logic ‘0’ when the number of 1s in the input sequence is odd and a logic ‘1’ when the number of 1s in the input sequence is even including zero.