Add a new Topic: Boolean Algebra

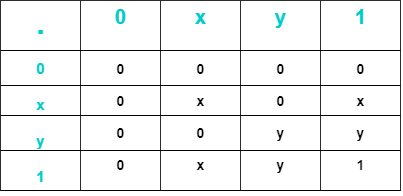
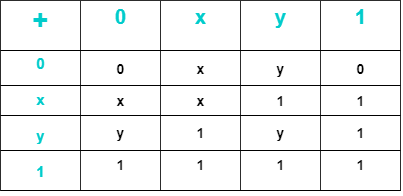
**After** page : **Digital Computers**

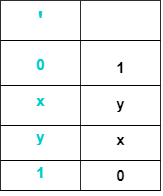
**Boolean Algebra**

A **Boolean Algebra** is an **algebra (set, operations, elements)** consisting of a set B with >=2 elements, together with **three operations**- the AND operation (Boolean Product), the OR operation (Boolean Sum), and the NOT operation (Complement) - defined on the set, such that for any element a, b, c…. of set B, a\*b, a+b and a’ are in B.

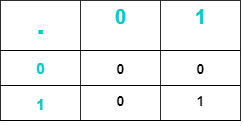
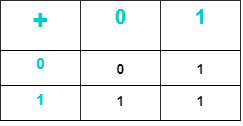
**For Example:**

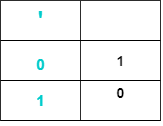
* Consider the four-element Boolean algebra B4 = ( {0, x, y, 1}; \*, +, ‘; 0, 1). The AND, OR and NOT operations are described by the following tables:



* Consider the two-element Boolean algebra B2 = ({0,1}; \*, +, ‘; 0, 1). The three operations **\* (AND), + (OR) and ‘ (NOT)** are defined as follows:



**Key Terms:**

* **Boolean Function:** Boolean algebra is a switching algebra that deals with binary variables and logic operations. The variables are designated by letters such as A, B, x, and y. The three basic logic operations are AND, OR and NOT. A Boolean function can be expressed algebraically with binary variables, the logic operation symbols, parentheses and equal sign. For a given combination of values of the variables, the Boolean function can be either 1 or 0. Consider for example, the Boolean Function:

**F = x + y’z**

The Function F is equal to 1 if x is 1 or if both y’ and z are equal to 1; F is equal to 0 otherwise.

* **Truth Table**: The relationship between a function and its binary variables can be represented in a truth table. To represent a function in a truth table we need a list of the **2n** combinations of the n binary variables.
* **Logic Diagram**: A Boolean function can be transformed from an algebraic expression into a logic diagram composed of AND, OR and NOT gates.

The purpose of Boolean algebra is to facilitate the analysis and design of digital circuits. It provides a convenient tool to:

* Express in algebraic form a truth table relationship between binary variables.
* Express in algebraic for the input-output relationship of logic diagrams.
* Find simpler circuits for the same function.

A Boolean function specified by a truth table can be expressed algebraically in many different ways. **Two ways of forming Boolean expressions are canonical and non-canonical forms.**

* **Canonical Form** express all binary variables in every product (AND) or sum (OR) term of the Boolean function. To determine the canonical sum-of-products form for a Boolean function F (A, B, C) = A’B + C’ + ABC, which is in **non-canonical form**, the following steps are used:

F = A’B + C’ +ABC

= A’B(C + C’) + (A + A’)(B + B’)C’ + ABC,

where **x + x’ = 1** is a basic identity of Boolean algebra

= A’BC + A’BC’ + ABC’ + AB’C’ + A’BC’ + A’B’C’ + ABC

= A’BC + A’BC’ + ABC’ + AB’C’ + A’B’C’ + ABC

By manipulating a Boolean expression according to Boolean algebra rules, one may obtain a simpler expression that will require fewer gates.

The below table lists the most **basic identities** of Boolean algebra. All the identities in the table can be proven by means of **truth tables**.

