

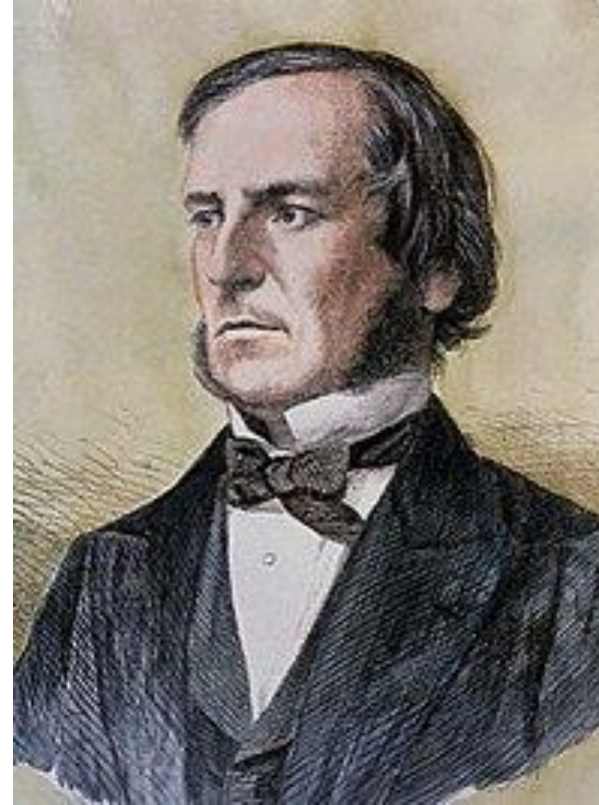
# Lecture 6 – Boolean algebra

Dr. Aftab M. Hussain,  
Associate Professor, PATRIOT Lab, CVEST

## Chapter 2

# Boolean algebra

- Boolean algebra was introduced by George Boole in his first book *The Mathematical Analysis of Logic* (1847)
  - Analyzed statements for their truth or falsehood
  - Created a mathematical framework to do so
- In the 1930s, while studying switching circuits, Claude Shannon observed that one could apply the rules of Boole's algebra in this setting, and he introduced switching algebra as a way to analyze and design circuits by algebraic means in terms of logic gates



George Boole



Claude Shannon

# Basic operations

- **NOT:** This operation is represented by a prime (sometimes by an overbar). For example,  $x' = z$  (or  $\bar{x} = z$ ); meaning that  $z$  is what  $x$  is not
- In other words, if  $x = 1$ , then  $z = 0$ , but if  $x = 0$ , then  $z = 1$
- The NOT operation is also referred to as the complement operation, since it changes a 1 to 0 and a 0 to 1, i.e., the result of complementing 1 is 0, and vice versa
- **AND:** This operation is represented by a dot or by the absence of an operator
- For example,  $x \cdot y = z$  or  $xy = z$
- The logical operation AND is interpreted to mean that  $z = 1$  if and only if  $x = 1$  and  $y = 1$ ; otherwise  $z = 0$
- **OR:** This operation is represented by a plus sign. For example,  $x + y = z$ , meaning that  $z = 1$  if  $x = 1$  or if  $y = 1$  or if both  $x = 1$  and  $y = 1$ . If both  $x = 0$  and  $y = 0$ , then  $z = 0$

# Basic operations

- An easy way to remember this is to make a table of all possible values of the variables and the results of these operations

AND

$x$	$y$	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

OR

$x$	$y$	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT

$x$	$x'$
0	1
1	0

# Binary logic

- Binary logic is different from binary numbers although it uses some of the same symbols
- In binary logic, we assume that variables can have ONLY two values – no other values are possible
- In binary numbers variables can have higher values or fraction or negative values, however, that is not the case in binary logic
- For example: in binary numbers,  $(1+1 = 10)_2$ , however, in binary logic,  $1+1 = 1$  because two trues make a true
- There are formal rules and proofs for many of the statements we make in binary logic
- In modern circuits, logic gates are used to perform binary logic using a variety of complex architectures

# Formalization of Boolean algebra

- Boolean algebra, like any other deductive mathematical system, may be defined with a set of elements, a set of operators, and a number of unproved axioms or postulates
- A *operator* defined on a set  $S$  of elements is a rule that assigns, to each pair of elements from  $S$ , a unique element from  $S$
- As an example, consider the relation  $a * b = c$ . We say that  $*$  is an operator if it specifies a rule for finding  $c$  from the pair  $(a, b)$  and also if  $a, b, c \in S$

# Postulates of Boolean algebra – Closure

- A set  $S$  is closed with respect to an operator if, for every pair of elements of  $S$ , the operator specifies a rule for obtaining an element of  $S$
- For example, the set of natural numbers  $N = \{1, 2, 3, 4, c\}$  is closed with respect to the operator  $+$  by the rules of arithmetic addition, since, for any  $a, b \in N$ , there is a unique  $c \in N$  such that  $a + b = c$
- The set of natural numbers is *not* closed with respect to the operator  $-$  by the rules of arithmetic subtraction, because  $2 - 3 = -1$  and  $2, 3 \in N$ , but  $(-1) \notin N$
- The Boolean logic structure is closed with respect to NOT, AND and OR logic operations

# Postulates of Boolean algebra – Associative law

- The operator  $*$  on a set  $S$  is said to be associative whenever  $(x * y) * z = x * (y * z)$  for all  $x, y, z, \in S$
- In case of real numbers, the multiplication and addition operations are associative while subtraction and division are not
- Similarly in binary logic, the operators AND and OR are associative
- Thus,  $x \text{ AND } (y \text{ AND } z)$  is the same as  $(x \text{ AND } y) \text{ AND } z$
- Also,  $x \text{ OR } (y \text{ OR } z)$  is the same as  $(x \text{ OR } y) \text{ OR } z$



# Postulates of Boolean algebra – Commutative law

- The operator  $*$  on a set  $S$  is said to be commutative whenever  $x * y = y * x$  for all  $x, y \in S$
- In case of real numbers, the multiplication and addition operations are commutative, while subtraction and division are not
- Similarly in binary logic, the operators AND and OR are commutative
- Thus,  $x \text{ AND } y$  is the same as  $y \text{ AND } x$
- Also,  $x \text{ OR } y$  is the same as  $y \text{ OR } x$

# Postulates of Boolean algebra – Identity

- A set  $S$  is said to have an identity element with respect to an operation  $*$  on  $S$  if there exists an element  $e \in S$  with the property that  $e * x = x * e = x$  for every  $x \in S$
- *Example:* The element 0 is an identity element with respect to the operator  $+$  on the set of integers  $I = \{..., -3, -2, -1, 0, 1, 2, 3, ...\}$ , since  $x + 0 = 0 + x = x$  for any  $x \in I$
- The set of natural numbers,  $N$ , has no identity element, since 0 is excluded from the set
- In Boolean logic, 0 is the identity element for OR operation and 1 is the identity element for AND operation

# Postulates of Boolean algebra – Distributive

- If  $*$  and  $\#$  are two operators on a set  $S$ ,  $*$  is said to be distributive over  $\#$  whenever  $x * (y \# z) = (x * y) \# (x * z)$
- In normal algebra, multiplication is distributive over addition:  $x(y+z) = xy + xz$
- In Boolean logic, The operator AND ( $\cdot$ ) is distributive over OR ( $+$ ); that is,  $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
- Also, the operator OR ( $+$ ) is distributive over AND ( $\cdot$ ); that is,  $x + (y \cdot z) = (x + y) \cdot (x + z)$
- This is counter intuitive!
- An easy way to prove the any statement in Boolean algebra is the make a table of all possible values of the variables and their results (*not possible in ordinary arithmetic*)

# Postulates of Boolean algebra – Distributive

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

<b><i>x</i></b>	<b><i>y</i></b>	<b><i>z</i></b>
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1