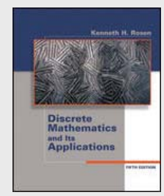
Chapter 10 (Part 1): Boolean Algebra

- Boolean Functions (10.1)
- Representing Boolean Functions (10.2)
- Logic Gates (10.3)



- It has started from the book titled "The laws of thought" written by George Boole in 1854
- Claude Shannon showed how the basic rules of logic could be used to design circuits
- The circuit in electronic devices have inputs, each of which is either 0 or 1 and produce outputs that are also 0 or 1

- The operation of a circuit is defined by a Boolean function that specifies the value of an output for each set of inputs
- One of the goal is to describe methods for finding a simplified expression (min number of sums and products) that represents a Boolean function (Karnaugh maps).

Boolean Functions (10.1)

- Introduction
  - Electronic and optimal switches are studied using the set {0, 1} and the rules of Boolean algebra
  - The 3 operations in Boolean algebra that are used are Complementation, Boolean Sum and the Boolean Product

Boolean Functions (10.1) (cont.)

– Complement of an element is defined by:

$$\bar{0} = 1; \; \bar{1} = 0$$

– Boolean sum denoted by + or by OR obeys:
1 + 1 = 1, 1+ 0 = 1, 0 + 1 = 1, 0 + 0 = 0

– Boolean product denoted by AND obeys:
1 . 1 = 1, 1 . 0 = 0, 0 . 1 = 0, 0 . 0 = 0.

Boolean Functions (10.1) (cont.)

– Example: Find the value $1 \cdot 0 + \overline{(0+1)}$

Solution: $1 \cdot 0 + \overline{(0+1)} = 0 + \bar{1} = 0 + 0 = 0$

- Boolean expressions & Boolean Functions

– Definition

Let B = {0, 1} then $B^n = \{(x_1, x_2, \ldots, x_n)$ such that $x_i \in B$ for $1 \le i \le n\}$. The variable x is called a Boolean variable if it assumes values only from B (0 or 1). A function B^n to B is called a Boolean function of degree n.

Boolean Functions (10.1) (cont.)

– Example: The function $F(x, y) = x\bar{y}$ from B^2 ={(x, y) such that (x, y) ∈ {0, 1} * {0,1)} is a Boolean function with:

F(1, 1) = 0, F(1, 0) = 1, F(0, 1) = 0, F(0, 0) = 0

– Example: Find the values of the Boolean function represented by $F(x, y, z) = xy + \bar{z}$

Solution: The values of this function are displayed on the following table:

Boolean Functions (10.1) (cont.)

x	y	z	xy	$\bar{z}$	$F(x, y, z) = xy + \bar{z}$
1	1	1	1	0	1
1	1	0	1	1	1
1	0	1	0	0	0
1	0	0	0	1	1
0	1	1	0	0	0
0	1	0	0	1	1
0	0	1	0	0	0
0	0	0	0	1	1

Identities of Boolean algebra

Identity	Name
$\overline{\overline{x}} = x$	Law of the double component
$x + x = x$ $\quad$ $x \cdot x = x$	Idempotent laws
$x + 0 = x$ $\quad$ $x \cdot 1 = x$	Identity laws
$x + 1 = 1$ $\quad$ $x \cdot 0 = 0$	Domination laws
$x + y = y + x$ $\quad$ $xy = yx$	Commutative laws
$x + (y + z) = (x + y) + z$ $x(yz) = (xy)z$	Associative laws
$x + yz = (x + y)(x + z)$	Distributive laws
$\overline{(xy)} = \overline{x} + \overline{y}$ $\quad$ $\overline{(x+y)} = \overline{x}\,\overline{y}$	De Morgan's laws
$x + xy = x$ $\quad$ $x(x + y) = x$	Absorption laws
$x + \overline{x} = 1$	Unit property
$x\overline{x} = 0$	Zero property

Boolean Functions (10.1) (cont.)

– Example: Show that the distributive law $x(y + z) = xy + xz$

Solution:

x	y	z	y + z	xy	xz	x(y + z)	xy + xz
1	1	1	1	1	1	1	1
1	1	0	1	1	0	1	1
1	0	1	1	0	1	1	1
1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0

Boolean Functions (10.1) (cont.)

- Duality
  - Goal: the dual of a Boolean expression is obtained by interchanging Boolean sums and Boolean products and interchanging 0s and 1s
  - Example: Find the duals of x (y + 0) and

$$\bar{x}.1 + (\bar{y} + z)$$

Solution: the duals are x + (y.1) and $(\bar{x} + 0)(\bar{y}z)$

Boolean Functions (10.1) (cont.)

  - The duality principle enables to obtain more identities. Since an identity between functions represented by Boolean expressions remains valid when the duals of both sides of the identity are taken

- The abstract definition of a Boolean Algebra
  - Goal: Recognize that a particular structure (set, propositions,…) is a Boolean algebra ⇒ all results of Boolean algebra will apply

Boolean Functions (10.1) (cont.)

– Definition 1:

A Boolean algebra is a set B with two binary operations ∨ and ∧, elements 0 and 1, and a unary operation ⁻ such that these properties hold for all x, y and z in B:

$$\left.\begin{matrix} x \vee 0 = x \\ x \wedge 1 = x \end{matrix}\right\} \textit{Identity laws}$$

$$\left.\begin{matrix} x \vee \bar{x} = 1 \\ x \wedge \bar{x} = 0 \end{matrix}\right\} \textit{Complement laws}$$

$$\left.\begin{matrix} (x \vee y) \vee z = x \vee (y \vee z) \\ (x \wedge y) \wedge z = x \wedge (y \wedge z) \end{matrix}\right\} \textit{Associative laws}$$

$$\left.\begin{matrix} x \vee y = y \vee x \\ x \wedge y = y \wedge x \end{matrix}\right\} \textit{Commutative laws}$$

$$\left.\begin{matrix} x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \\ x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \end{matrix}\right\} \textit{Distributive laws}$$

Boolean Functions (10.1) (cont.)

– The set of propositions in n variables, with the ∨ and ∧ operators, **F** and **T**, and the negation operator, also satisfies all the properties of a Boolean algebra

– Similarly, the set of subsets of a universal set Ω with the union and intersection operators, the empty set and the universal set, and the set complementation operator is a Boolean algebra

Representing Boolean Functions (10.2)

- Two problems of Boolean algebra are emphasized in this section

1. Given the values of a Boolean function, how can a Boolean expression that represents this function be found?
2. Is there a smaller set of operators that can be used to represent all Boolean functions?

Representing Boolean Functions (10.2) (cont.)

- Sum-of-products expansions
  - Goal: Find a Boolean expression that represents a Boolean function
  - Example: Find Boolean expressions that represent the functions F(x, y, z) and G(x, y, z) given by the following table:

x	y	z	F	G
1	1	1	0	0
1	1	0	0	1
1	0	1	1	0
1	0	0	0	0
0	1	1	0	0
0	1	0	0	1
0	0	1	0	0
0	0	0	0	0

Representing Boolean Functions (10.2) (cont.)

x	y	z	F	G
1	1	1	0	0
1	1	0	0	1
1	0	1	1	0
1	0	0	0	0
0	1	1	0	0
0	1	0	0	1
0	0	1	0	0
0	0	0	0	0

Solution:

- To form the Boolean expressions for the function F(x,y,z), we need to look at the table and notice that F has the value 1 only if x = 1 and y = 0 and z =1 otherwise F has value 0
  $\Rightarrow$ F(x,y,z) = $x\bar{y}z$

- To form the Boolean expression for the function G(x,y,z), we have to notice that G(x,y,z) = 1 $\Leftrightarrow$ x = y = 1 and z = 0 or x = z = 0 and y = 1; otherwise G(x,y,z) = 0
  $\Rightarrow$ G(x,y,z) = $(xy\bar{z}) + (\bar{x}y\bar{z})$

Representing Boolean Functions (10.2) (cont.)

– Definition 1:

A literal is a Boolean variable or its complement. A minterm of the Boolean variables x1=, x_2, …, x_n is a Boolean product $y_1y_2\ldots y_n$, where $y_i = x_i$ or $y_i = \bar{x}_i$. Hence, a minterm is a product of n literals, with one literal of each variable.

The minterm $y_1y_2\ldots y_n$, is 1 if and only if each y_i is 1, and therefore $x_i = 1$ when $y_i = x_i$ and $x_i = 0$ when $y_i = \bar{x}_i$.

Representing Boolean Functions (10.2) (cont.)

- Example: Find a minterm that equals 1 if $x_1 = x_3 = 0$ and $x_2 = x_4 = x_5 = 1$, and equals 0 otherwise

  Solution: The minterm is: $\bar{x}_1 x_2 \bar{x}_3 x_4 x_5$

- Minterms are used to determine the Boolean expression of a Boolean function in a table

Representing Boolean Functions (10.2) (cont.)

- Example: Find the sum-of-products expansion for the function $F(x, y, z) = (x + y)\bar{z}$

  Solution (1):

$$
\begin{aligned}
(x + y)\bar{z} &= x\bar{z} + y\bar{z} = x1\bar{z} + 1y\bar{z} \\
&= x(y + \bar{y})\bar{z} + (x + \bar{x})y\bar{z} \\
&= xy\bar{z} + x\bar{y}\bar{z} + xy\bar{z} + \bar{x}y\bar{z} \\
&= xy\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z} \quad (\text{since } u + u = u)
\end{aligned}
$$

Representing Boolean Functions (10.2) (cont.)

Solution (2): We use the minterm technique from a table

x	y	z	x + y	$\bar{z}$	(x + y) $\bar{z}$
1	1	1	1	0	0
1	1	0	1	1	1
1	0	1	1	0	0
1	0	0	1	1	1
0	1	1	1	0	0
0	1	0	1	1	1
0	0	1	0	0	0
0	0	0	0	1	0

Representing Boolean Functions (10.2) (cont.)

We need only to spot the set of values (x,y,z) for which $F(x,y,z) = (x+y)\bar{z} = 1.$ This set is:

{(1,1,0);(1,0,0);(0,1,0)}

We need to find the minterm that equals 1 if

$$\underbrace{(x = y = 1 \text{ and } z = 0)}_{xy\bar{z}} \text{ OR } \underbrace{(x = 1 \text{ and } y = z = 0)}_{x\bar{y}\bar{z}} \text{ OR } \underbrace{(x = z = 0 \text{ and } y = 1)}_{\bar{x}y\bar{z}}$$

$$xy\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z}$$

– Theorem 1: Every Boolean function can be represented using the three Boolean operators **.**, +, and ⁻ (complement)

Representing Boolean Functions (10.2) (cont.)

Functional completeness

- The set {., +, ⁻} is functionally complete because of theorem 1
- Question: Can we find a smaller set of functionally complete operators?
- We answer "yes" to this question if one of the three operators can be expressed in terms of the other two
- One of the De Morgan's laws provide the solution: $x + y = \overline{\overline{x}\,\overline{y}}$ since $\overline{(x + y)} = \overline{x}\,\overline{y}$

Representing Boolean Functions (10.2) (cont.)

- Similarly, we have: $xy = \overline{\overline{x} + \overline{y}}$
- Finally, we have found a smaller set functionally complete; it contains only 2 operators!
- Question: Can we still reduce the set to only one operator? The answer is "yes".
- Indeed, there are 2 sets that contain only one operator, they are: the **NAND** and the **NOR**.

Representing Boolean Functions (10.2) (cont.)

- NAND (|) is defined as

  1|1 = 0 and 1|0 = 0|1= 0| =1

- NOR (↓) is defined as

  1↓1 = 1↓0 = 0↓1 = 0 and 0↓0 =1

- Since the set {., ‾} is functionally complete and $\overline{x}$ = x|x, xy = (x|y) | (x|y) then the set {|} is functionally complete.

- Home-exercise: Prove that the set {↓} is functionally complete.

Logic Gates (10.3)

✓ Introduction

- A computer, or electronic devices, is made up of a number of circuits

- Each circuit can be designed using the rules of Boolean algebra

- The basic element of circuits are called gates

- Each type of gates implements a Boolean Operation

Logic Gates (10.3) (cont.)

– All circuits studied in this chapter provides output that depends only on the input, and not on the current state of the circuit

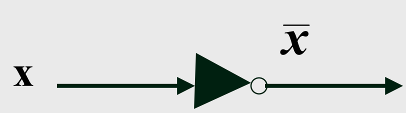
– These circuits that have no memory capabilities are called combinational circuits or gating networks

– Combinatorial circuits are built using 3 types of elements: (1) inverter, (2) OR gate, and (3) AND gate

Logic Gates (10.3) (cont.)

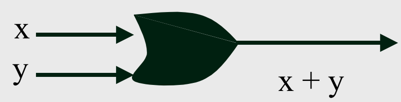
– Role of each type of elements:

1. Inverter: it accepts the value of one Boolean variable as input and produces the complement of this value as its output.



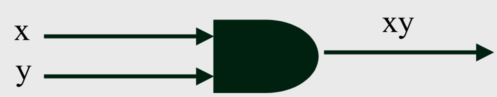
Logic Gates (10.3) (cont.)

2. OR gate: it accepts as input values of 2 or more Boolean variables. The output is the Boolean sum of their values



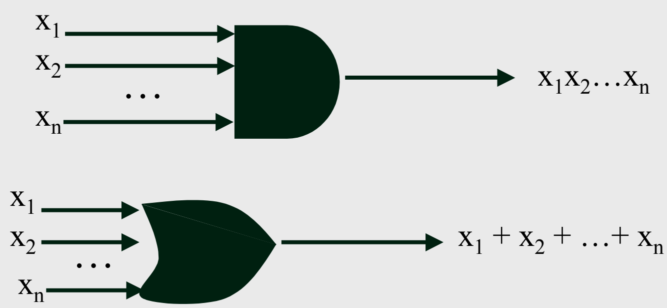
Logic Gates (10.3) (cont.)

3. AND gate: The inputs to this gate are the values of two or more Boolean variables. The output is the Boolean product of their values.



Logic Gates (10.3) (cont.)

– We can have multiple inputs to AND and OR:



Gates with n inputs

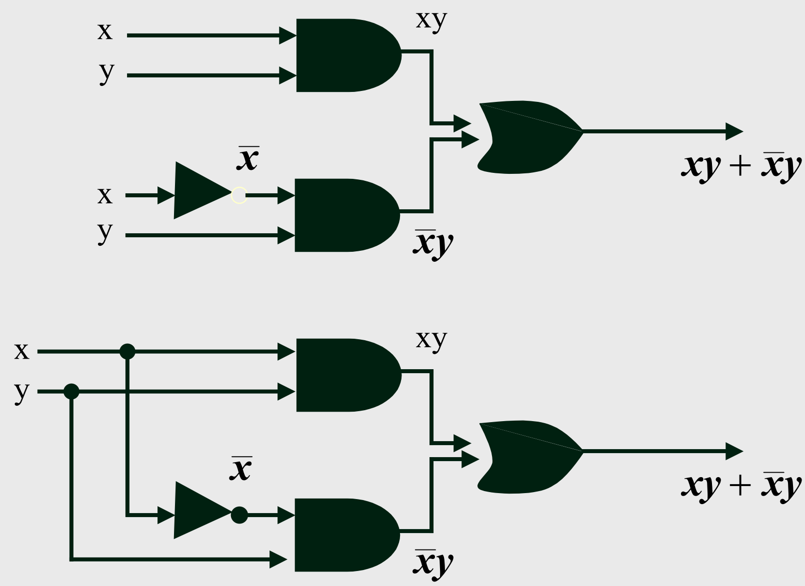
Logic Gates (10.3) (cont.)

- Combinations of Gates

  – Combinational circuits can be constructed using a combination of inverters, OR gates, and AND gates

  – When combinations of circuits are formed, some gates may share input using branching

  – The following figure shows 2 ways of drawing the same circuit



Two ways to draw the same circuit

Logic Gates (10.3) (cont.)

– Example: Construct circuits that produce the following outputs:

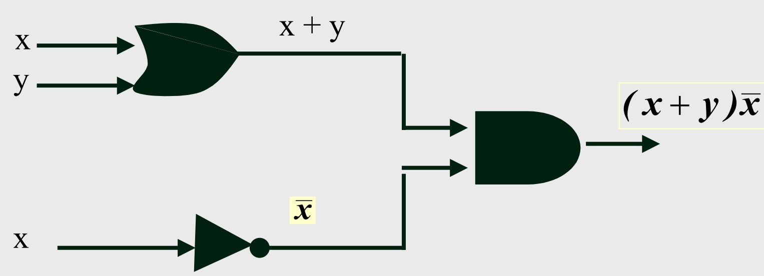
a) $(x + y)\bar{x}$

b) $\overline{x\overline{(y + \bar{z})}}$

c) $(x + y + z)(\bar{x}\,\bar{y}\,\bar{z})$

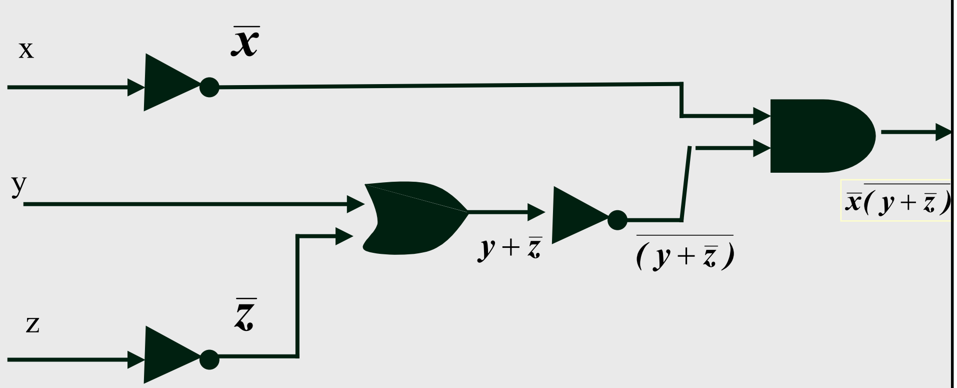
Logic Gates (10.3) (cont.)

– *Solution* for (a):



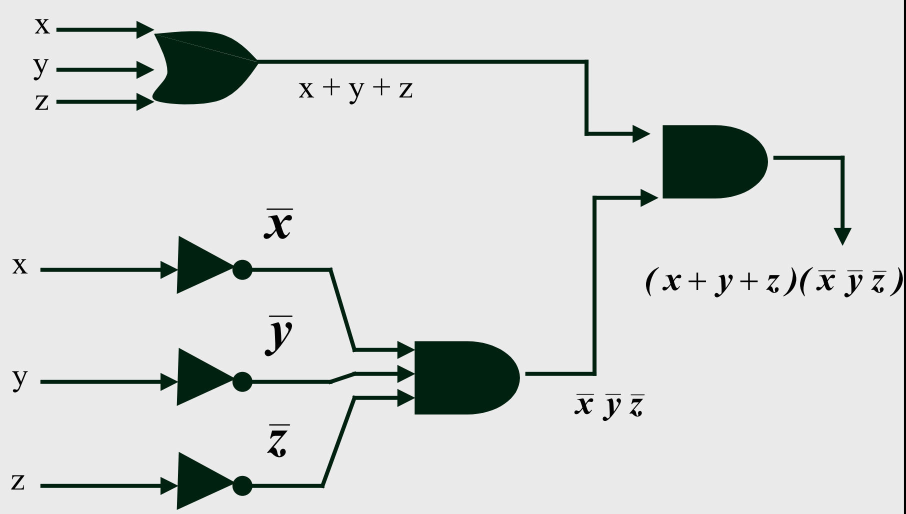
Logic Gates (10.3) (cont.)

– *Solution* for (b):



Logic Gates (10.3) (cont.)

– *Solution* for (c):



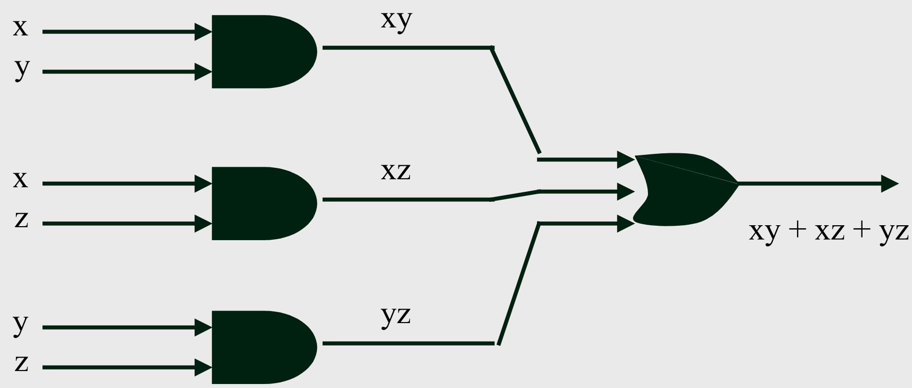
Logic Gates (10.3) (cont.)

- Example of circuits

  – Example: A committee of 3 individuals decides issues for an organization. Each individual votes either yes or no for each proposal that arises. A proposal is passed if it receives at least 2 yes votes. Design a circuit that determines whether a proposal passes.

  Solution: We want to have the following Boolean function: xy + xz + yz represented by the following circuit:

Logic Gates (10.3) (cont.)



A circuit for majority voting