Discrete Mathematics (MATH 151)

Dr. Borhen Halouani

King Saud University

February 9, 2020

1. Boolean Algebra
   - Introduction
   - Boolean Functions
   - Representing Boolean Functions
   - Logic Gates

Introduction

- Boolean algebra traces its origins to an 1854 book by mathematician George Boole.
- Boolean algebra is a division of mathematics which deals with operations on logical values and incorporates binary variables
- We study Boolean algebra as a foundation for designing and analyzing digital systems!
- Boolean algebra provides the operations and the rules for working with the set $\{0, 1\}$.

The operations and the rules

The three operations in Boolean algebra:

1. the Boolean sum, denoted by $+$ or by OR.
2. the Boolean product, denoted by $\cdot$ or by AND.
3. The complement of an element, denoted with a bar, is defined by $\bar{0} = 1$ and $\bar{1} = 0$.

$1 + 1 = 1$	$1 \cdot 1 = 1$
$1 + 0 = 1$	$1 \cdot 0 = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$
$0 + 0 = 0$	$0 \cdot 0 = 0$

Example 2.1

Using the definitions of complementation, the Boolean sum, and the Boolean product to find the value of: $1 \cdot 0 + \overline{(0+1)}$

Solution

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

Remark

The complement, Boolean sum, and Boolean product correspond to the logical operators, $\neg$, $\vee$, and $\wedge$, respectively, where 0 corresponds to F (false) and 1 corresponds toT (true).
Equalities in Boolean algebra can be directly translated into equivalences of compound propositions.

Example 2.2

Translate $1 \cdot 0 + \overline{(0+1)} = 0$ *into a logical equivalence.*

Solution

We obtain a logical equivalence when we translate each 1 into a T, each 0 into an F, each Boolean sum into a disjunction, each Boolean product into a conjunction, and each complementation into a negation.We obtain

$$(T \wedge F) \vee \neg(T \vee F) \equiv F.$$

Example 2.3

Translate the logical equivalence $(T \wedge T) \vee \neg F \equiv T$ into an identity in Boolean algebra.

Solution

We obtain an identity in Boolean algebra when we translate each T into a 1, each F into a 0, each disjunction into a Boolean sum, each conjunction into a Boolean product, and each negation into a complementation.We obtain

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

Boolean Expressions and Boolean Functions

- Let $B = \{0, 1\}$.
- Then $B^n = \{(x_1, x_2, ..., x_n) | x_i \in B \; for \; 1 \leq i \leq n\}$ is the set of all possible n-tuples of 0s and 1s.
- The variable x is called a **Boolean variable** if it assumes values only from B, that is, if its only possible values are 0 and 1.
- A function from B^n to B is called a **Boolean function of degree n**.

Boolean Expressions and Boolean Functions

Example 2.4

The function $F(x, y) = x \cdot \bar{y}$ from the set of ordered pairs of Boolean variables to the set $\{0, 1\}$ is a Boolean function of degree 2 with $F(1, 1) = 0, F(1, 0) = 1, F(0, 1) = 0,$ and $F(0, 0) = 0$. We display these values of F in Table 1.

TABLE 1

x	y	$F(x, y)$
1	1	0
1	0	1
0	1	0
0	0	0

Boolean Expressions and Boolean Functions

Example 2.5

Find the values of the Boolean function represented by $F(x, y, z) = x \cdot y + \overline{z}$

TABLE 2

x	y	z	xy	$\overline{z}$	$F(x, y, z) = xy + \overline{z}$
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

Boolean Expressions and Boolean Functions

- Boolean functions can be represented using expressions made up from variables and Boolean operations.
- The Boolean expressions in the variables $x_1, x_2, \ldots, x_n$ are defined recursively as $0, 1, x_1, x_2, \ldots, x_n$ are Boolean expressions.
- If E_1 and E_2 are Boolean expressions, then $\overline{E}_1$, $(E_1 E_2)$, and $(E_1 + E_2)$ are Boolean expressions.
- Each Boolean expression represents a Boolean function.
- The values of this function are obtained by substituting 0 and 1 for the variables in the expression.

Boolean Expressions and Boolean Functions

- Boolean functions F and G of n variables are equal if and only if $F(b_1, b_2, ..., b_n) = G(b_1, b_2, ..., b_n)$ whenever $b_1, b_2, ..., b_n$ belong to B.
- Two different Boolean expressions that represent the same function are called equivalent.
- The complement of the Boolean function F is the function $\overline{F}$, where $\overline{F}(x_1, ..., x_n) = \overline{F(x_1, ..., x_n)}$.
- Let F and G be Boolean functions of degree n. The Boolean sum F + G and the Boolean product FG are defined by $(F + G)(x_1, ..., x_n) = F(x_1, ..., x_n) + G(x_1, ..., x_n)$, $(FG)(x_1, ..., x_n) = F(x_1, ..., x_n)G(x_1, ..., x_n)$.

Boolean Expressions and Boolean Functions

Theorem 2.1

There are 2^{2^n} different Boolean functions on n Boolean variables.

Example 2.6

How many different Boolean functions of degree two are there?

A Boolean function of degree two is a function from a set with four elements, namely, pairs of elements from $B = \{0, 1\}$, to B, a set with two elements. Hence, there are 16 different Boolean functions of degree two. In Table 3 we display the values of the 16 different Boolean functions of degree two, labeled $F_1, F_2, \ldots, F_{16}$.

Boolean Expressions and Boolean Functions

TABLE 3 The 16 Boolean Functions of Degree Two.

x	y	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Table 4 displays the number of different Boolean functions of degrees one through six. The number of such functions grows extremely rapidly.

TABLE 4 The Number of Boolean Functions of Degree n.

Degree	*Number*
1	4
2	16
3	256
4	65,536
5	4,294,967,296
6	18,446,744,073,709,551,616

Identities of Boolean Algebra

- There are many identities in Boolean algebra. The most important of these are displayed in Table 5.
- Each of the identities in Table 5 can be proved using a table.
- We will prove one of the distributive laws in this way in next example

TABLE 5 Boolean Identities.

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

Identities of Boolean Algebra(Example)

Example 2.7

Show that the distributive law $x(y+z) = xy + xz$ is valid.

TABLE 6 Verifying One of the Distributive Laws.

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

Identities of Boolean Algebra(Example)

Example 2.8

Prove the absorption law $x(x + y) = x$ using the other identities of Boolean algebra shown in Table 5.

Solution

1. $x(x + y) = (x + 0)(x + y)$ Identity law for the Boolean sum
2. $= x + (y \cdot 0)$ Distributive law of the Boolean sum over the Boolean product
3. $= x + (0 \cdot y)$ Commutative law for the Boolean product
4. $= x + 0$ Domination law for the Boolean product
5. $= x$ Identity law for the Boolean sum.

Duality

Definition 2.1

The dual of a Boolean expression is obtained by interchanging Boolean sums and Boolean products and interchanging 0s and 1s.

Example 2.9

Find the duals of $x(y+0)$ *and* $\overline{x} \cdot 1 + (\overline{y} + z)$.
The duals are $x + (y \cdot 1)$ *and* $(\overline{x} + 0)(\overline{y}z)$, *respectively.*

Remark 2.1

The dual of a Boolean function F *represented by a Boolean expression is the function represented by the dual of this expression. This dual function, denoted by* F^d.

Definition 2.2

A Boolean algebra is a set B with two binary operations $\vee$ and $\wedge$, elements 0 and 1, and a unary operation $\bar{}$ such that these properties hold for all x, y, and z in B:

$$\left.\begin{aligned} x \vee 0 &= x \\ x \wedge 1 &= x \end{aligned}\right\} \quad \text{Identity laws}$$

$$\left.\begin{aligned} x \vee \overline{x} &= 1 \\ x \wedge \overline{x} &= 0 \end{aligned}\right\} \quad \text{Complement laws}$$

$$\left.\begin{aligned} (x \vee y) \vee z &= x \vee (y \vee z) \\ (x \wedge y) \wedge z &= x \wedge (y \wedge z) \end{aligned}\right\} \quad \text{Associative laws}$$

$$\left.\begin{aligned} x \vee y &= y \vee x \\ x \wedge y &= y \wedge x \end{aligned}\right\} \quad \text{Commutative laws}$$

$$\left.\begin{aligned} x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \\ x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) \end{aligned}\right\} \quad \text{Distributive laws}$$

Sum-of-Products Expansions

Two important problems of Boolean algebra will be studied in this section.

- The first problem is: Given the values of a Boolean function, how can a Boolean expression that represents this function be found?
This problem will be solved by showing that any Boolean function can be represented by a Boolean sum of Boolean products of the variables and their complements.
The solution of this problem shows that every Boolean function can be represented using the three Boolean operators $\cdot, +$, and $\bar{}$
- The second problem is: Is there a smaller set of operators that can be used to represent all Boolean functions?
We will answer this question by showing that all Boolean functions can be represented using only one operator.

Sum-of-Products Expansions

Example 2.10

Find Boolean expressions that represent the functions F(x, y, z) and G(x, y, z), which are given in Table 1.

TABLE 1

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

- An expression that has the value 1 when x = z = 1 and y = 0, and the value 0 otherwise, is needed to represent F.
- Such an expression can be formed by taking the Boolean product of x, $\overline{y}$, and z.
- This product, $x\overline{y}z$, has the value 1 if and only if $x = \overline{y} = z = 1$, which holds if and only if x = z = 1 and y = 0.

- To represent G, we need an expression that equals 1 when $x = y = 1$ and $z = 0$, or $x = z = 0$ and $y = 1$.
- We can form an expression with these values by taking the Boolean sum of two different Boolean products.
- The Boolean product $xy\overline{z}$ has the value 1 if and only if $x = y = 1$ and $z = 0$. Similarly, the product $\overline{x}y\overline{z}$ has the value 1 if and only if $x = z = 0$ and $y = 1$.
- The Boolean sum of these two products, $xy\overline{z} + \overline{x}y\overline{z}$, represents G,

Definition 2.3

A literal is a Boolean variable or its complement.
A minterm of the Boolean variables $x_1, x_2, \ldots, x_n$ *is a Boolean product* $y_1 y_2 \ldots y_n$, *where* $y_i = x_i$ *or* $y_i = \overline{x_i}$.
Hence, a minterm is a product of n literals, with one literal for each variable.

Example 2.11

Find a minterm that equals 1 if $x_1 = x_3 = 0$ *and* $x_2 = x_4 = x_5 = 1$, *and equals 0 otherwise.*
- The minterm $\overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4 \cdot x_5$ *has the correct set of values.*

Definition 2.4

The sum of minterms that represents the function is called the complete sum-of-products expansion ***(CSP)*** *or the disjunctive normal form of the Boolean function.*

Example 2.12

Find the complete sum-of-products expansion (CSP) for the function $F(x, y, z) = (x + y)\overline{z}$.

Solution We will find the sum-of-products expansion of $F(x, y, z)$ in two ways.

- First, we will use Boolean identities to expand the product and simplify.We find that:

$$
\begin{aligned}
F(x, y, z) &= (x + y)\overline{z} && \\
&= x\overline{z} + y\overline{z} && \text{Distributive law} \\
&= x1\overline{z} + 1y\overline{z} && \text{Identity law} \\
&= x(y + \overline{y})\overline{z} + (x + \overline{x})y\overline{z} && \text{Unit property} \\
&= xy\overline{z} + x\overline{y}\,\overline{z} + xy\overline{z} + \overline{x}y\overline{z} && \text{Distributive law} \\
&= xy\overline{z} + x\overline{y}\,\overline{z} + \overline{x}y\,\overline{z}. && \text{Idempotent law}
\end{aligned}
$$

- Second, we can construct the sum-of-products expansion by determining the values of F for all possible values of the variables x, y, and z. These values are found in Table 2. The sum-of products expansion of F is the Boolean sum of three minterms corresponding to the three rows of this table that give the value 1 for the function. This gives: $F(x, y, z) = xy\overline{z} + x\overline{y}\,\overline{z} + \overline{x}y\overline{z}$.

TABLE 2

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

Definition 2.5

Maxterm is a sum of all the literals (with or without complement).

- *A maxterm is a sum of input values where every input value occurs exactly once.*
- *When expressing a truth table, each maxterm represents a row where the output is 0, such that the maxterm will give a true value in all cases except for that row.*

Definition 2.6

the complete Product of Sums ***(CPS)****: A boolean expression consisting purely of Maxterms (sum terms) is said to be in canonical product of sums form.*

x	y	z	minterm	maxterm
1	1	1	xyz	$\overline{x}+\overline{y}+\overline{z}$
1	1	0	$xy\,\overline{z}$	$\overline{x}+\overline{y}+z$
1	0	1	$x\overline{y}z$	$\overline{x}+y+\overline{z}$
1	0	0	$x\overline{y}\,\overline{z}$	$\overline{x}+y+z$
0	1	1	$\overline{x}yz$	$x+\overline{y}+\overline{z}$
0	1	0	$\overline{x}y\overline{z}$	$x+\overline{y}+z$
0	0	1	$\overline{x}\,\overline{y}z$	$x+y+\overline{z}$
0	0	0	$\overline{x}\,\overline{y}\,\overline{z}$	$x+y+z$

Example 2.13

Find the complete product of sum expansion (CPS) for the function $F(x, y, z) = (x + y)\overline{z}$.

Solution We will find the complete product of sum expansion of $F(x, y, z)$ in two ways.

- First, we will use Boolean identities to expand the product and simplify.We find that:

$$\begin{aligned}
F(x, y, z) &= (x + y)\overline{z} \\
&= (x + y + 0)(0 + \overline{z}) \\
&= ((x + y) + (z\overline{z}))((x\overline{x}) + \overline{z}) \\
&= (x + y + z)(x + y + \overline{z})(x + \overline{z})(\overline{x} + \overline{z}) \\
&= (x + y + z)(x + y + \overline{z})(x + 0 + \overline{z})(\overline{x} + 0 + \overline{z}) \\
&= (x + y + z)(x + y + \overline{z})(x + (y\overline{y}) + \overline{z})(\overline{x} + (y\overline{y}) + \overline{z})
\end{aligned}$$

$$= (x + y + z)(x + y + \overline{z})(x + y + \overline{z})(x + \overline{y} + \overline{z})(\overline{x} + y + \overline{z})(\overline{x} + \overline{y} + \overline{z})$$

$$= (x + y + z)(x + y + \overline{z})(x + \overline{y} + \overline{z})(\overline{x} + y + \overline{z})(\overline{x} + \overline{y} + \overline{z})$$

- Second, we can construct the complete product of sums expansion by determining the values of F for all possible values of the variables x, y, and z. These values are found in Table 2. The complete product of sums expansion of F is the Boolean sum of five maxterms corresponding to the five rows of this table that give the value 0 for the function. This gives:

$F(x, y, z) = (\overline{x}+\overline{y}+\overline{z})(\overline{x}+y+\overline{z})(x+\overline{y}+\overline{z})(x+y+\overline{z})(x+y+z)$

Remark

we can obtained **CPS** by giving the CSP for the complement of the function, and we take the complement of the CSP give the CPS.

$$CPS(F) = \overline{CSP(\overline{F})}$$

Functional Completeness

- Every Boolean function can be expressed as a Boolean sum of minterms.
- Each minterm is the Boolean product of Boolean variables or their complements.
- This shows that every Boolean function can be represented using the Boolean operators $\cdot, +,$ and $\bar{}$.
- Because every Boolean function can be represented using these operators we say that the set $\{\cdot, +, \bar{}\}$ is functionally complete.

Question

Can we find a smaller set of functionally complete operators?

Question

Can we find a smaller set of functionally complete operators?

- We can do so if one of the three operators of this set can be expressed in terms of the other two.
- This can be done using one of De Morgan's laws.
- We can eliminate all Boolean sums using the identity $x + y = \overline{\overline{x}\ \overline{y}}$ which is obtained by taking complements of both sides in the second De Morgan law, given in Table 5, and then applying the double complementation law. This means that the set $\{\cdot, \bar{}\}$ is functionally complete.

Remark

- Similarly, we could eliminate all Boolean products using the identity $xy = \overline{\overline{x} + \overline{y}}$
- which is obtained by taking complements of both sides in the first De Morgan law, given in Table 5, and then applying the double complementation law. Consequently $\{+, ^-\}$ is functionally complete.
- Note that the set $\{+, \cdot\}$ is not functionally complete, because it is impossible to express the Boolean function $F(x) = \overline{x}$ using these operators

Karnaugh Maps

MSP form (Minimal sum-of-product), is obtained using K-maps method.

- To reduce the number of terms in a Boolean expression representing a circuit, it is necessary to find terms to combine.
- There is a graphical method, called a Karnaugh map or K-map, for finding terms to combine for Boolean functions involving a relatively small number of variables.
- K-maps give us a visual method for simplifying sum of products expansions.

We will first illustrate how K-maps are used to simplify expansions of Boolean functions in n variables.

- Presented as a two-dimensional grid of 2^n squares.
- Each axis represents the different input values to the circuit, and the contents of each square represents the output value for the intersecting input values.
    - ∗ Horizontally and vertically adjacent squares only differ by a single input variable.
    - ∗ Number of rows and columns must be a power of 2.
    - ∗ **Note:** row and column labels must also differ by a single digit.
- Simplified circuit is found by circling groups of adjacent 1's on the grid.
- **Result:** The minimal expression for the circuit.

Examles

- K-maps in Two Variables.

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

Find the K-maps for: (a) $xy + \bar{x}y$, (b) $x\bar{y} + \bar{x}y$, (c) $x\bar{y} + \bar{x}y + \bar{x}\,\bar{y}$

	y	$\bar{y}$
x	1	
$\bar{x}$	1	

(a)

	y	$\bar{y}$
x		1
$\bar{x}$	1	

(b)

	y	$\bar{y}$
x		1
$\bar{x}$	1	1

(c)

Simplify the sum-of-products expansions given,

	y	
x	1	
$\bar{x}$	1	

	y	$\bar{y}$
x		1
$\bar{x}$	1	

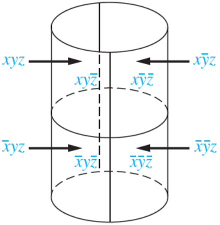
	y	$\bar{y}$
x		1
$\bar{x}$	1	1

Examles

A K-map in three variables is a rectangle divided into eight cells. One of the ways to form a K-map in three variables is shown in Figure (a). This K-map can be thought of as lying on a cylinder, as shown in Figure (b).

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x	xyz	$xy\bar{z}$	$x\bar{y}\bar{z}$	$x\bar{y}z$
$\bar{x}$	$\bar{x}yz$	$\bar{x}y\bar{z}$	$\bar{x}\bar{y}\bar{z}$	$\bar{x}\bar{y}z$

(a)



(b)

Example:

$$f = \overline{x}_1\overline{x}_2x_3 + \overline{x}_1x_2x_3 + x_1\overline{x}_2x_3 + x_1x_2\overline{x}_3 + x_1x_2x_3$$

$x_1 \backslash x_2x_3$	00	01	11	10
0	0	1	1	0
1	0	1	1	1

Next step: circle blocks of 1's

- Cannot contain any 0's in the block.
- Height and width of block must be a power of 2.
- Blocks are allowed to overlap.

$x_1 \backslash x_2x_3$	00	01	11	10
0	0	1	1	0
1	0	1	1	1

Circled blocks correspond to x_3 and x_1x_2, so
$F(x_1, x_2, x_3) = x_1 \cdot x_2 + x_3$.
We denote by **MSP** form (Minimal sum-of-product), is obtained using K-maps method.

Example 2.14

Given the truth table on Inputs Output the right, determine the simplest equivalent gate arrangement.

AB \ CD	00	01	11	10
00	0	1	1	0
01	0	1	1	1
11	1	1	1	0
10	1	1	1	0

$$X = A\overline{C} + D + \overline{A}BC$$

Inputs				Output
A	B	C	D	X
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

MPS form (Minimal product-of-sums), is obtained using K-maps method.
Instead of creating a disjunction of cases where the output of the circuit is 1, the product-of-sums technique creates a conjunction of the cases where the output is 0.
- These equations can be reduced using the same techniques used on minterms.

- boolean logic rules
- Karnaugh maps

Example:

$$F(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(\overline{x_1} + x_2 + x_3)(\overline{x_1} + x_2 + \overline{x_3})(\overline{x_1} + \overline{x_2} + \overline{x_3})$$

$x_1 \backslash x_2x_3$	00	01	11	10
0	0	1	1	1
1	0	0	0	1

So we can minimize F as:

$$F(x_1, x_2, x_3) = (x_2 + x_3)(\overline{x_1} + \overline{x_3})$$

Introduction

- Boolean algebra is used to model the circuitry of electronic devices.
- Each input and each output of such a device can be thought of as a member of the set $\{0, 1\}$.
- A computer, or other electronic device, is made up of a number of circuits.
- Each circuit can be designed using the rules of Boolean algebra.
- The basic elements of circuits are called **gates**.

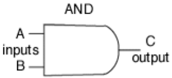
The three main ways of specifying the function of a combinational logic circuit are:

1. Boolean Algebra. This forms the algebraic expression showing the operation of the logic circuit for each input variable either True or False that results in a logic "1" output.
2. Truth Table. A truth table defines the function of a logic gate by providing a concise list that shows all the output states in tabular form for each possible combination of input variable that the gate could encounter.
3. Logic Diagram. This is a graphical representation of a logic circuit that shows the wiring and connections of each individual logic gate, represented by a specific graphical symbol, that implements the logic circuit.

1. The **inverter**, which accepts the value of one Boolean variable as input and produces the complement of this value as its output.
2. The **OR gate**. The inputs to this gate are the values of two or more Boolean variables. The output is the Boolean sum of their values.
3. The **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.
4. The **NAND gate** function is a combination of the two separate logical functions, the AND function and the NOT function in series.
5. The **NOR gate** is also a combination of two separate logic functions, Not and OR connected together to form a single logic function which is the same as the OR function except that the output is inverted.

- AND gate

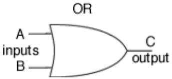
  → $C = A \cdot B$



A	B	C
0	0	0
1	0	0
0	1	0
1	1	1

- OR gate

  → $C = A + B$



A	B	C
0	0	0
1	0	1
0	1	1
1	1	1

- Buffer

Buffer

A	B
0	0
1	1

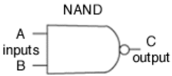
- NOT Gate

→ $B = \overline{A}$

NOT
(Inverter)

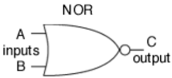
A	B
0	1
1	0

- NAND Gate



A	B	C
0	0	1
1	0	1
0	1	1
1	1	0

- NOR Gate



A	B	C
0	0	1
1	0	0
0	1	0
1	1	0

Combinations of Gates

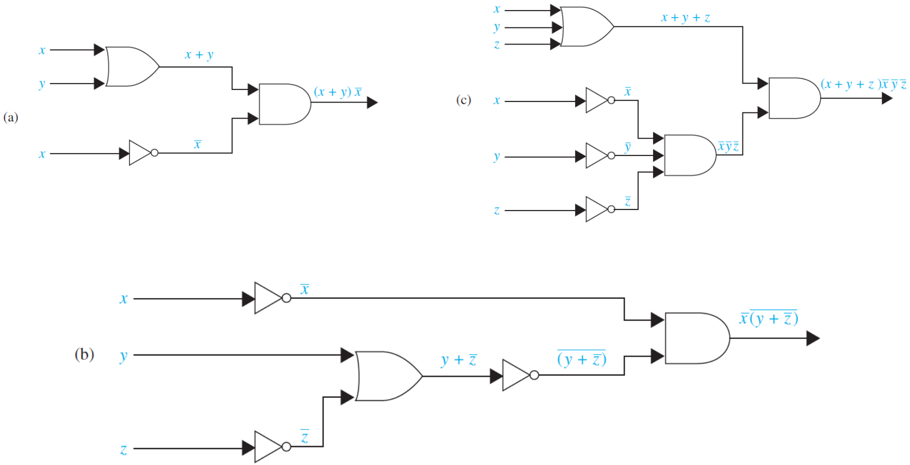
Combinational circuits can be constructed using a combination of inverters, OR gates, and AND gates.

Example 2.15

Construct circuits that produce the following outputs:

1. $(x + y)\overline{x}$
2. $\overline{x}\,\overline{(y + \overline{z})}$
3. $(x + y + z)(\overline{x}\;\overline{y}\;\overline{z})$

Combinations of Gates

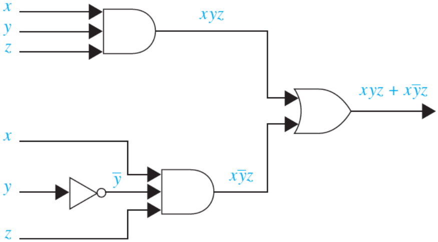


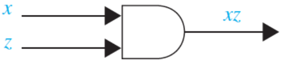
Minimization of Circuits

Example 2.16

represent the Boolean function by logic circuit
$f(x, y, z) = xyz + x\overline{y}z$

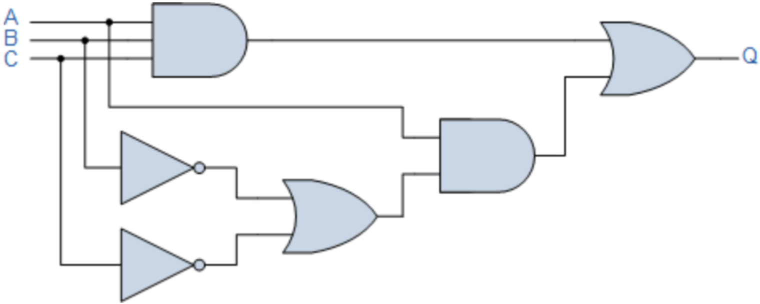
-Solution: $f(x, y, z) = xyz + x\overline{y}\ z = 1 \cdot (xz) = xz$



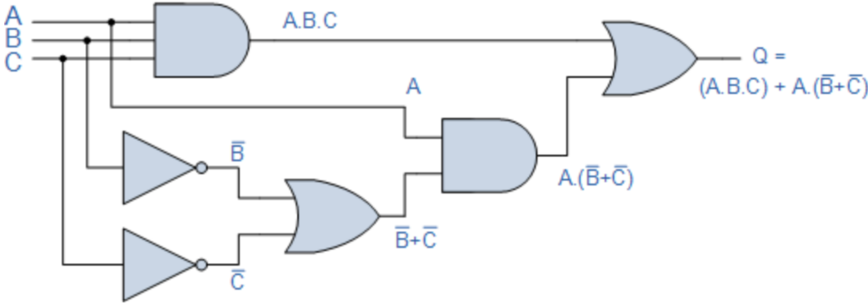


Example 2.17

Find the Boolean algebra expression for the following system.



Solution


A
B
C
A.B.C
A
$\overline{B}$
$\overline{C}$
$\overline{B}+\overline{C}$
$A.(\overline{B}+\overline{C})$
Q =
$(A.B.C) + A.(\overline{B}+\overline{C})$