CIS11032 Logic Designing & Computer Organization

Lesson 03 Boolean Logic

AR Fathima Shafana
Department of ICT
Faculty of Technology
South Eastern University of Sri Lanka

Learning Outcomes

At the completion of this lesson students should be able to,

- Understand the Boolean logic
- Apply Boolean algebra to simplify Boolean equations
- Use K-Map to simplify logic Equations

COURSE OUTLINE

- Boolean Algebra
- Fundamental Laws and Theorems of Boolean Algebra
- Minterm and Maxterm
- Karnaugh Maps
- Don’t Care States

INTRODUCTION

- George Boole, a nineteenth-century English Mathematician, developed a system of logical algebra by which reasoning can be expressed mathematically

- Boole's system of logical algebra was investigated as a tool for analyzing and designing relay switching circuits

- Boolean algebra has turned out to be a convenient tool in two areas:
    I. **Analysis:** It is an economical way of describing the function of digital circuitry.
    II. **Design**: Given a desired function, Boolean algebra can be applied to develop a simplified implementation of that function.

- Today Boolean algebra is the backbone of computer circuit analysis

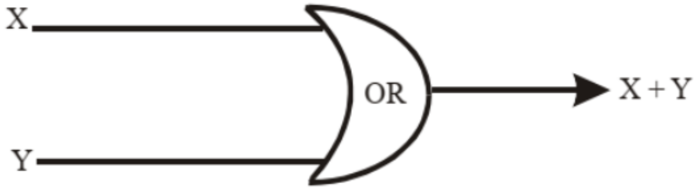
INTRODUCTION contd.

- Boolean algebra works mainly with the following mathematical operations: addition, multiplication and negation.
- These operations are associated with the OR gate, the AND gate and NOT gate respectively.

Logical Addition

- Logical Addition is symbolized with '+'
- However, besides the general addition, it performs the OR operation

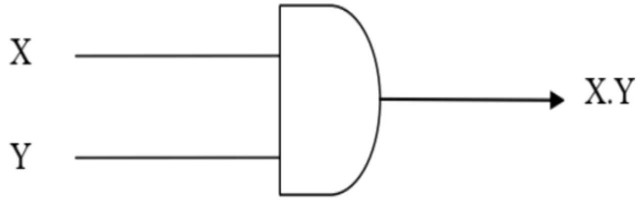
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1



Logical Multiplication

- Logical Multiplication is symbolized with a '.'
- It performs the AND operation

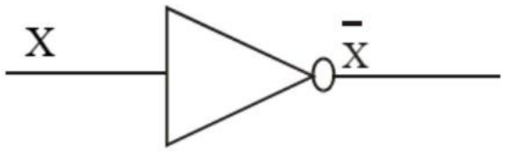
A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1



Negation

- The symbol used to represent complementation of a variable is a bar (-) above the variable
- The output signal is always the opposite or complement of the input signal
- It takes only one input, hence known as a **Unary Operator**

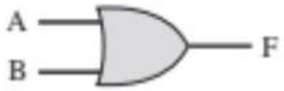
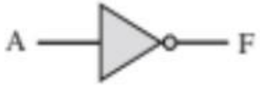
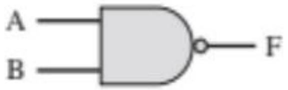
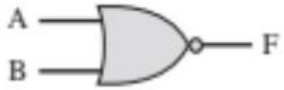
A	$\bar{A}$
0	1
1	0



- Beside the three operators stated above certain other operators are also much prevalently used in practice.

P	Q	NOT P ($\overline{P}$)	P AND Q ($P \cdot Q$)	P OR Q ($P + Q$)	P NAND Q ($\overline{P \cdot Q}$)	P NOR Q ($\overline{P + Q}$)	P XOR Q ($P \oplus Q$)
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	1	0	0	0

Logic Gates for Boolean Operators

Name	Graphical Symbol	Algebraic Function	Truth Table
AND		$F = A \bullet B$ or $F = AB$	A B | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = A + B$	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
NOT		$F = \overline{A}$ or $F = A'$	A | F 0 | 1 1 | 0
NAND		$F = \overline{AB}$	A B | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = \overline{A + B}$	A B | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
XOR		$F = A \oplus B$	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0

Fundamental Laws and Theorems of Boolean Algebra

OR Operations

1. $X + 0 = X$
2. $X + 1 = 1$
3. $X + X = X$
4. $X + \bar{x} = 1$

AND Operations

5. $X . 0 = 0$
6. $X . 1 = X$
7. $X . X = X$
8. $X . \bar{x} = 0$

3rd and 7th are called "Idempotent"

Fundamental Laws and Theorems of BooleanAlgebra

Double Complement

9. $\bar{\bar{x}} = X$

Commutative Laws

10. X + Y = Y + X
11. XY = YX

Associative Laws

12. (X + Y) +Z = X +(Y + Z)
13. (X . Y). Z =X. (Y. Z)

Distribution Law

14. X (Y + Z) = XY + XZ

Dual of Distributive Law

15. X + Y .Z = (X + Y) . (X + Z)

Fundamental Laws and Theorems of BooleanAlgebra

Laws of Absorption

16. X + XZ = X

17. X (X + Z) = X

Identity Theorem

18.X+ $\bar{x}$ Y =X+Y

19. X ($\bar{x}$ +Y) = XY

De' Morgan's Theorem

20. $\overline{X+Y} = \bar{X}\bar{Y}$

21 $\overline{XY} = \bar{X} + \bar{Y}$

EXAMPLE 1

$$\mathbf{XY + YZ} + \overline{Y}\ \mathbf{Z = XY + Z}$$

$$\begin{aligned} \text{L.H.S.} &= XY + YZ + \overline{Y}\,Z \\ &= XY + Z(Y + \overline{Y}) \\ &= XY + Z.1 \\ &= XY + Z \\ \text{L.H.S} &= \text{R.H.S.} \end{aligned}$$

Exercises: Simplify using Boolean Algebraic Laws

1. $AB + AB' + A'B$
2. $AB + A'B + ABC' + AB'C$
3. $(A+B).(A'+B)$
4. $AB + AB + ABC$

Show that $A+AB+ABC = A$

Question 01

$AB + AB' + A'B$

$A(B+B') + A'B$ $\quad$ $B+B' = 1$

$A.1 + A'B$ $\quad$ $A+A'B = A+B$

$A+B$

Question 02 $AB + A'B + ABC' + AB'C$

$B(A+A') + ABC' + AB'C$ $\qquad$ $A+A' = 1$

$B.1 + ABC' + AB'C$

$B + ABC' + AB'C$

$B + BAC' + AB'C$ $\qquad$ $X+XY=X$

$B + AB'C$

$B + B'AC$ $\qquad$ $X+X'Y=X+Y$

$B + AC$

Question 03 (A+B).(A'+B)

A.A' + AB + A'B + B.B

0 + AB + A'B + B

B + AB + A'B

B + A'B

B

A.A' = 0; B.B=B

B + AB = B

X + XY = X

Question 04 AB + AB + ABC

AB + AB + ABC

AB + ABC

AB

X+X =X

X + XY = X

Show that $A+AB+ABC = A$

LHS : $A + AB + ABC$ $\qquad$ $X+XY=X$

$A + ABC$ $\qquad$ $X+XY=X$

A

RHS

Hence, Proved.

Example 2

Find the complement of the expression

I. $X + YZ$

II. $\bar{A}\,B + C\,\overline{D}$

$(X + YZ)'$

$X' . (YZ)'$

$X' . (Y'+Z')$

$(A+B)' = A'.B'$

$(AB)' = A'+B'$

Example 2

Find the complement of the expression

I. $X + YZ$

II. $\bar{A}B + C\bar{D}$

(A’B+CD’)’

(A+B’).(C’+D)

Minterm and Maxterm

- **Minterm (Sum of Products)**
  - The products of the variables are separated by a plus sign.
  - The variables can be complemented or un-complemented.

$$AB + A\overline{B} + \overline{A}B + \overline{A}\,\overline{B} + AB\overline{C} + A\overline{B}C + A\overline{B}C$$

- **Maxterm (Product of Sums)**
  - The sum of variables are separated by a multiplication sign.

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

$$(A + \overline{B} + C)(A + \overline{B} + \overline{C})(\overline{A} + \overline{B} + C)$$

Fundamental Products

- The products that produce a high (1) output are called Fundamental products.

- For the two input variables A and B,

A	B	Fundamental Product	Output Z
0	0	$\overline{A}\,\overline{B}$	1
0	1	$\overline{A}B$	1
1	0	$A\overline{B}$	1
1	1	AB	1

For three variables A, B and C

A	B	C	Output Z	Fundamental Product	Output for product	Sum terms	Output for Sum
0	0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$	1	$A+B+C$	0
0	0	1	0	$\overline{A}\,\overline{B}\,C$	1	$A+B+\overline{C}$	0
0	1	0	1	$\overline{A}\,B\,\overline{C}$	1	$A+\overline{B}+C$	0
0	1	1	1	$\overline{A}\,BC$	1	$A+\overline{B}+\overline{C}$	0
1	0	0	0	$A\overline{B}\,\overline{C}$	1	$\overline{A}+B+C$	0
1	0	1	0	$A\overline{B}\,C$	1	$\overline{A}+B+\overline{C}$	0
1	1	0	1	$AB\overline{C}$	1	$\overline{A}+\overline{B}+C$	0
1	1	1	0	ABC	1	$\overline{A}+\overline{B}+\overline{C}$	0

How to obtain Minterm and Maxterm from truth table?

Sum of Products (Minterm)

For each row of the truth table for which the output is 1, the Boolean term is the product of variables that are equal to 1 and the complement of variable that are equal to 0. The sum of these products is the desired Boolean equation.

Product of Sums (Maxterm)

For each row of the truth table for which the output is 0, the Boolean term is the sum of the variables that are equal 0 plus the complement of the variables that are equal to 1. The product of these sum is the desired Boolean equation

A	B	C	Output Functional Values
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

From the truth table,

Sum-of-Product Equation

$$X = \overline{A}\,\overline{B}\,C + \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + AB\,\overline{C}$$

Product-of-Sums Equation:

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

Boolean Expression and Logic Diagrams

- Boolean expressions are written to mathematically describe the behavior of a logic circuit.
- Using a truth table and the Boolean expression, one can determine which combinations of input signals cause the output signal.
- By the use of Boolean Algebra, the reduced expression can be derived.

EXAMPLE 01

$X = AB + ABC + A\overline{B}\,\overline{C} + A\overline{C}$

(a) Draw the logic diagram for the expression.

(b) Minimize the expression.

(c) Draw the logic diagram for the reduced expression.

X = AB + ABC+AB'C'+AC'

= AB + AB'C' + AC'

=AB + AC'B' + AC'   X+XY= X

=AB+AC'

=A(B+C')

EXAMPLE 02

Show that,

I. $ABC + \bar{A}B + AB\bar{C} = B$

II. $A\bar{C}D + \bar{A}B\bar{C}D + \bar{A}\bar{B}D + A\bar{B}CD = D(\bar{C} + \bar{B})$

Karnaugh Maps

- Karnaugh Maps are much preferred to minimize the Boolean expressions instead of Boolean Algebra.
- A Karnaugh map is a graphical form of a truth table and consists of a square or rectangular array of adjacent cells or blocks.
- The number of cells in a particular map depends on the number of variables in the Boolean expression to be minimized.
- The number of cells for a particular map is determined from expression.

$$\mathbf{N = 2^n}$$

where N = number of cells required for the Karnaugh map

n = number of variables in the Boolean expression

	A	$\overline{A}$
B		
$\overline{B}$		

	$\overline{B}$	A
$\overline{B}\,\overline{C}$		
$\overline{B}\,C$		
$B\,C$		
$B\,\overline{C}$		

	$\overline{A}\,\overline{B}$	$\overline{A}\,B$	AB	$A\,\overline{B}$
$\overline{C}\,\overline{D}$				
$\overline{C}\,D$				
CD				
$C\,\overline{D}$				

NOTE: Two cells of a map are considered to be adjacent as long as their respective addressed differed by no more than one variable

Simplification Steps

- The desired minimized Boolean expression is obtained from the Karnaugh map by applying the following two steps:

1. All 1's must be included in at least one group. It is permissible, and desirable, to enclosed a 1 more than once if it facilitates enlarging another enclosure
2. Each group represents a minterm. The sum of the minterms that represent these groups is the minimized Boolean expression in sum-of-products form corresponding to the given logic function

Rules for Simplification

1. Groups may not include any cell containing a zero.
2. Groups may be horizontal or vertical but not diagonal
3. Group must contain 1,2,4,8 or in general 2^n cells.
4. Each group should be as large as possible.
5. Each cell containing a one must be in **at least** one group.
6. Groups may overlap.
7. Groups may wrap around the table. The leftmost cell in a row may be grouped with the rightmost cell and the top cell in a column may be grouped with the bottom cell.
8. There should be as few groups as possible, as long as this does not contradict any of the previous rules.

EXAMPLE 01

1. Minimize the Boolean expression $X = AB + A\bar{B} + BC$ using K-map

X = AB+AB'+BC

X= ABC + ABC' + AB'C + AB'C' + ABC + A'BC

A \ BC	00	01	11	10
0			1	
1	1	1	1	1

A

BC

X = A+BC

EXAMPLE 01

2. Using K-map reduce the expression Y = A+ AB′C + AB

$Y = A + AB'C + AB$

$Y = AB'C+AB'C'+ABC'+ABC + AB'C + ABC+ABC'$

A \ BC	00	01	11	10
0				
1	1	1	1	1

A

EXAMPLE 02

$X = \bar{A}BC + B\bar{C} + \bar{B}\bar{C} + ABC$

I. Minimize the above expression by using **K-map**

II. Draw the circuit for the **minimized expression**.

Don’t care Conditions and K-map

Example 03

Minimize the following Boolean functions using **Karnaugh Maps**

1. $F(A,B,C,D) = \sum(0,1,2,4,5,8,9,10,11)$
2. $F(A,B,C) = \sum(0,1,2,5,6)$
3. $F(A,B,C,D) = \sum(0,3,4,5,7,9,11,12,13,14) + \sum d(3,15)$

	A	B	C	D	Output
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

$$\sum(0,1,2,4,5,8,9,10,11)$$

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

A'C'

AB'

B'D'

A'C'+AB'+B'D'

Example 04

A traffic light control system needs to be designed for a side road that allows cars to pass when:

There is no traffic on the main road ($M = 0$).

If there is traffic on the main road ($M = 1$), cars can still pass if a pedestrian is not present ($P = 0$).

Design a circuit that takes M (Main Road Traffic) and P (Pedestrian Request) as inputs and outputs S (Side Road Green Light), where:

$S = 1$ means the side road has a green light.

$S = 0$ means the side road must stop (red light).

Example 05

A security system in a building is designed to allow access (A = 1) only if certain conditions are met. The system has three inputs:

S = Security mode (1 if active, 0 if not)
K = Keycard detected (1 if valid keycard, 0 if not)
P = PIN entered correctly (1 if correct, 0 if incorrect)

The system should grant access (A = 1) under the following conditions:

If security mode is OFF (S = 0), access is granted regardless of K or P.
If security mode is ON (S = 1), access is granted only if both a valid keycard (K = 1) and a correct PIN (P = 1) are provided.

Design a circuit for the above instance.

Exercise

An automatic door control system operates based on three sensors:

M = Motion detected (1 if movement is detected, 0 otherwise)

L = Lock status (1 if the door is locked, 0 if unlocked)

T = Timer expired (1 if the door should be closed due to timeout, 0 otherwise)

The system should open the door (D = 1) under the following conditions:

- If motion is detected (M = 1) and the door is not locked (L = 0), the door opens.
- If motion is detected (M = 1) and the timer has not expired (T = 0), the door opens.
- If motion is not detected (M = 0), the door stays closed (D = 0).
- If the door is locked (L = 1), it stays closed regardless of the other inputs.

Thank you