CHAPTER 3

LOGIC CIRCUITS

Unit Objectives:

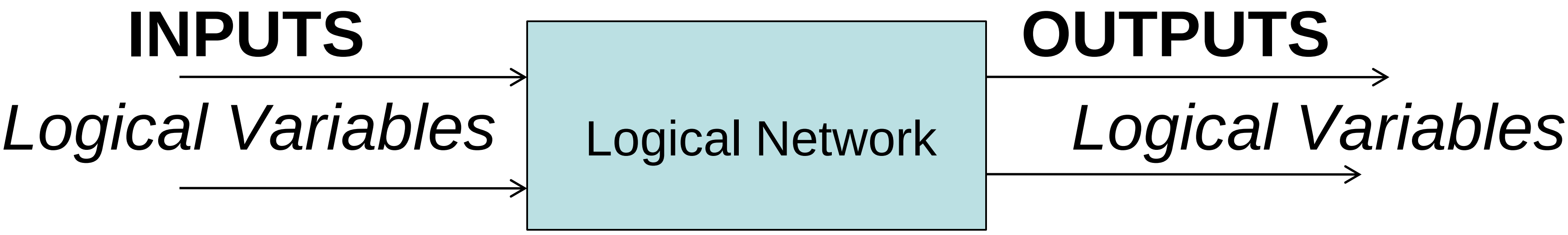
- Understand the significance of logic gates.
- Explain the rules and usage of boolean Algebra.
- Discuss the process of implementation of Boolean Algebra.
- Explain the significance of function simplification and Karnaugh maps.

LOGIC CIRCUITS

- Are electronic devices used in computers to perform logic operations on 1 or more input signals.
- Inputs and outputs to digital computers are in one of the 2 possible states / Voltage levels which are represented by 0’s or 1’s
- If the higher voltage is associated with 1 and lower voltage with a 0, the circuit is said to be based upon **positive logic.**
- If the lower voltage is associated with a 1 and higher voltage with 0, the circuit is said to be based on **negative logic**.

LOGIC CIRCUITS

- Any variable that can take on two states e.g. (0/1, True/false; on/off) is called **a logical variable**.
- A circuit whose inputs and outputs are described by logical variables is called a **logical network**.



Logical Networks

There are two types of logical networks:

- **Combinatorial networks:** Their outputs depend on the current inputs.

- **Sequential Networks:** Their outputs depend on both the current state of the network as well as the inputs

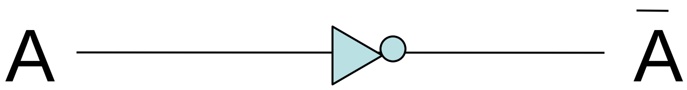
Logic Gates

- ***A combinatorial circuit with only one output*** is called a **logic gate**. They accept logical values at their inputs and they produce corresponding logical values at their outputs.
- It performs a logical operation on one or more binary inputs and produces a single binary output.
- **A table listing all the outputs for the various inputs** is called a **truth table.**
- All combinatorial circuits can be constructed from the 7 most common logic gates.

Logic Gates

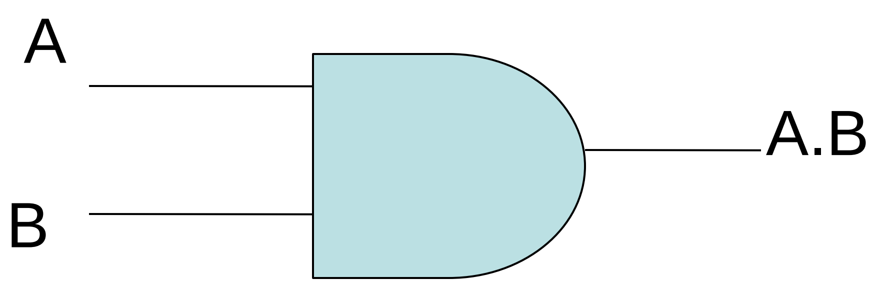
1. The inverter(NOT) Gate:

When the input is 1 the output is 0 and vice versa.



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

2. The AND Gate:

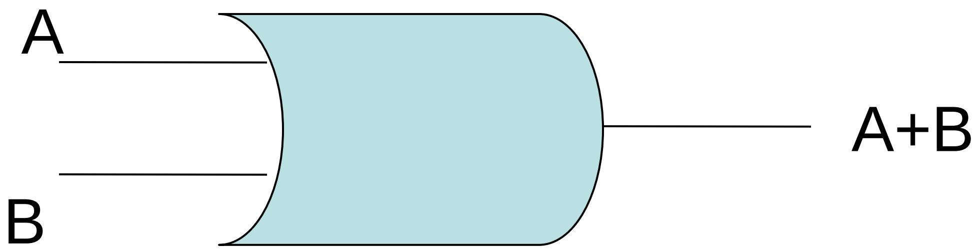


Input		Output
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

The output is 1 if all the inputs are 1's

Logic Gates

3. The OR Gate:



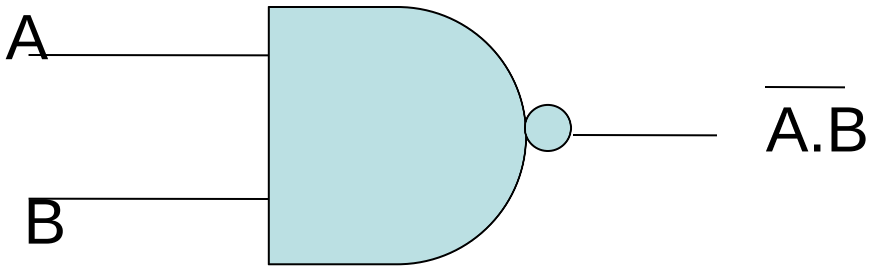
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

- **The output is 0 if all the inputs are 0's.**

- If an inverter is combined with another logic gate, the presence of the inverter is indicated by placing a small circle at the affected input or output.

Logic Gates

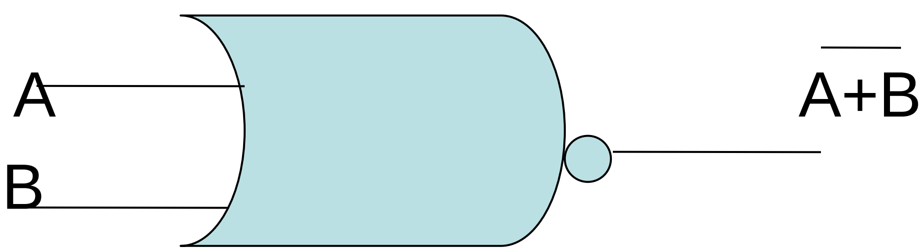
4. The NAND (NOT AND) Gate



A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

The output is 0 if all the inputs are 1’s.

5. The NOR (NOT OR) Gate

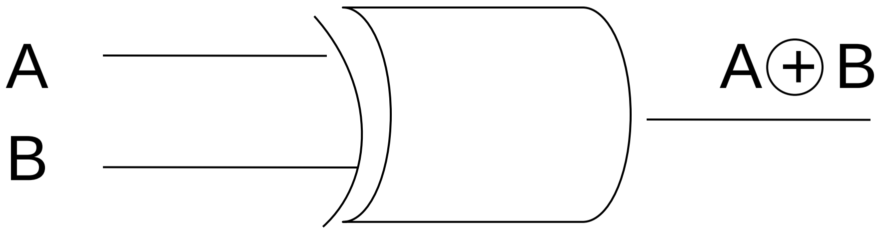


A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

The output is 1 if all the inputs are 0’s.

LOGIC GATES

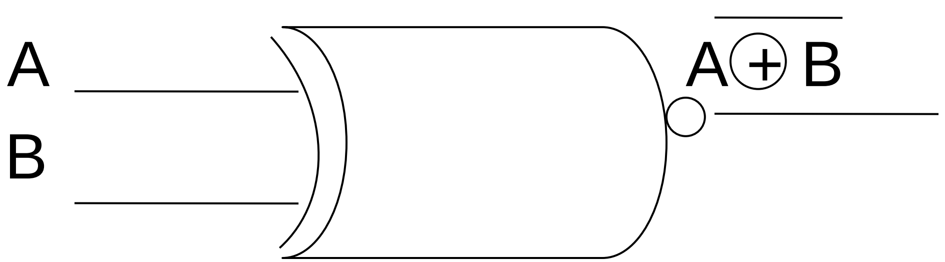
6. The EXCLUSIVE OR Gate



A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

The output is 0 if all the inputs are the same

7. The EXCLUSIVE NOR Gate

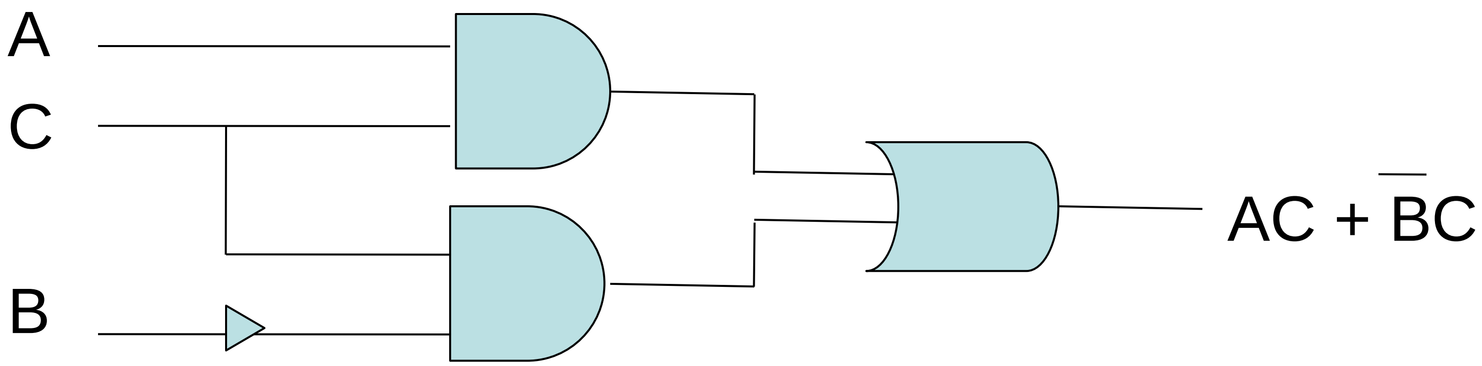


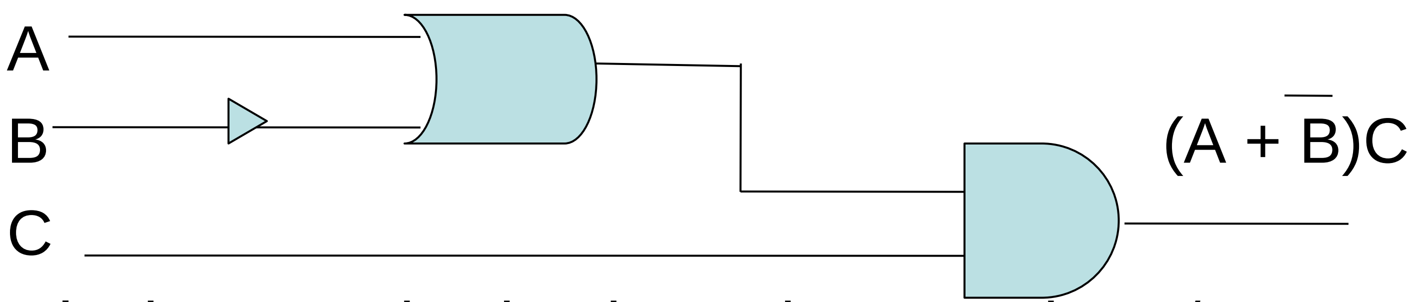
A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

The output is 1 if all the inputs are the same.

Complex Logic gates

- Logic circuits can be built by combining several of the elementary logic gates.
- **A graphical illustration of a logic circuit** is called a **logical diagram**





- Two logic networks that have the same input/output characteristics are said to be equivalent.

EXAMPLE

Prove that $AC + \overline{B}C = (A + \overline{B})C$

NOTE:

- The best way to prove equivalence is to use the **truth tables.**

A	B	C	$\overline{B}$	AC	$\overline{B}C$	$AC + \overline{B}C$	$A + \overline{B}$	$(A + \overline{B})C$
0	0	0	1	0	0	0	1	0
0	0	1	1	0	1	1	1	1
0	1	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0
1	0	1	1	1	1	1	1	1
1	1	0	0	0	0	0	1	0
1	1	1	0	1	0	1	1	1

BOOLEAN ALGEBRA

It is a mathematical structure that consists of a set containing only a 0 and 1, the unary operator (complementation) and the binary operation of addition and multiplication. **Subtraction and Division** are not defined in Boolean algebra

$A = \overline{\overline{A}}$

$AA = A$

$A + A = A$

$A . 0 = 0$

$A + 0 = A$

$A . 1 = A$

$A + 1 = 1$

$A . \overline{A} = 0$

$A + \overline{A} = 1$

$AB = BA$

$A + B = B + A$

$(AB)C = A(BC)$

$A + (B + C) = (A + B) + C$

$A(B + C) = AB + AC$

$(B + C)A = BA + CA$

$\overline{A + B} = \overline{A}\ \overline{B}$

$\overline{AB} = \overline{A} + \overline{B}$

$AB + A\overline{B} = A$

$A + AB = A$

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

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

$(A + B)(A + C) = A + BC$

$A(A + B) = A$

$A\overline{B} + B = A + B$

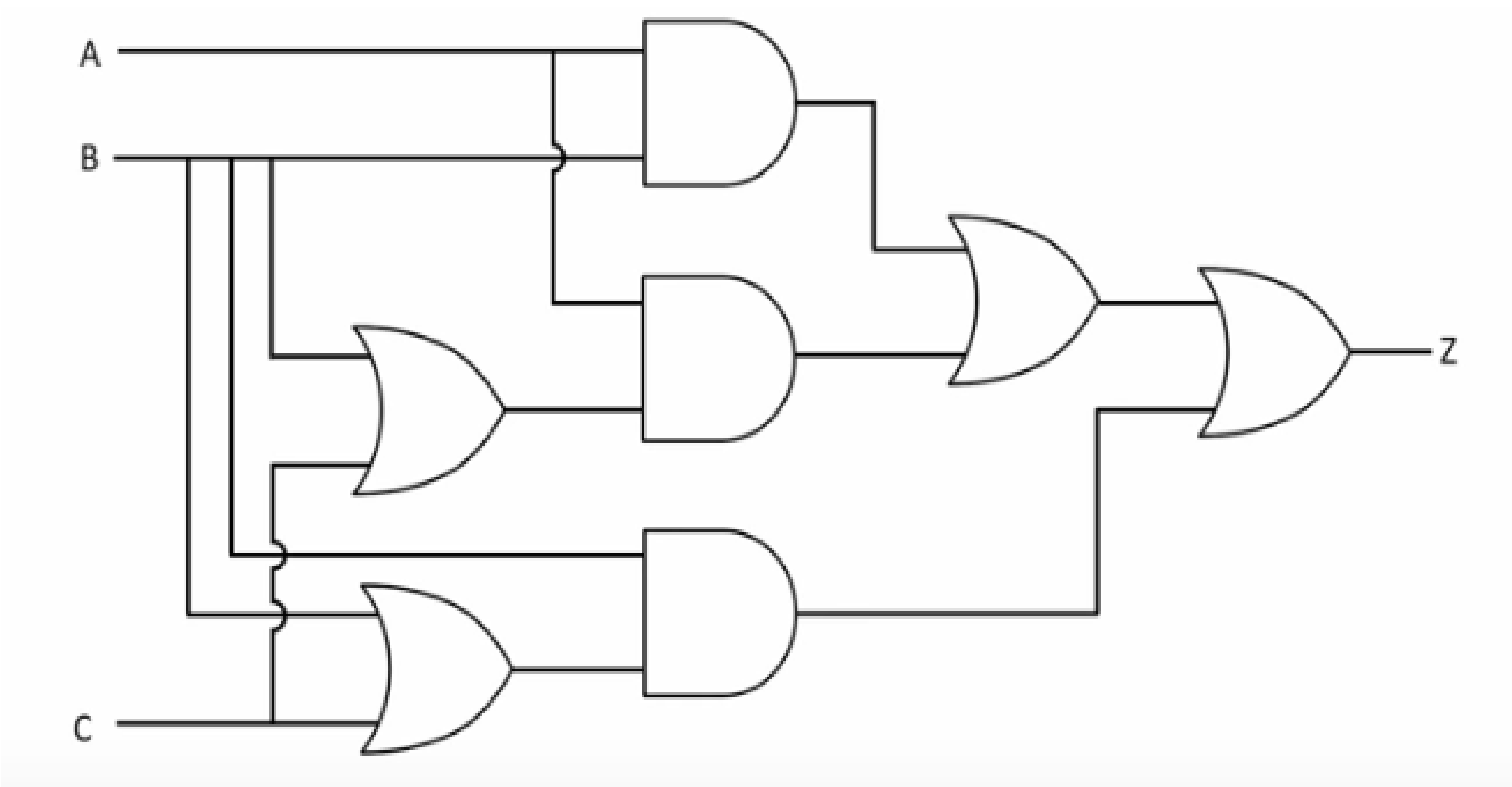
$\overline{A}B + A\overline{B} = A \oplus B$

$\overline{AB}(A + B) = A \oplus B$

Why Boolean Algebra?

- Its used to simplify logic expressions without changing its functionality.
- Used to prove equivalence of logic networks.

Example 1: Simplify the circuit below using Boolean Algebra.



Example 2:

Prove that $\mathbf{\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC}$ **is** equivalent to $\mathbf{BC + AC + AB}$**.**

$\mathbf{\bar{A}\bar{B}C + BC + AB}$

$= C(\bar{A}\bar{B} + B) + AB$

$= C(\bar{A} + B) + AB$

$= \bar{A}C + BC + AB$

$= \bar{A}C + (\bar{A} + A)BC + AB$

$= \bar{A}C + \bar{A}BC + ABC + AB$

$= \bar{A}C(1 + B) + AB(C + 1)$

$= \mathbf{\bar{A}C + AB}$

Digital Design Process

- ➢ Determine all the input/output relationships that must be true for the network being designed and put them in convenient tabular form.
- ➢ Use the drawn up table to find Boolean expressions for each output.
- ➢ Simplify the expressions from 2 above
- ➢ Use the expressions resulting from step 3 to develop the desired logical diagram.

Three design tools

- Truth Table (*To define a logical network*)
- Boolean Expression *(for minimization)*
- Logic diagram *(For the actual design)*

Example

Design a three input network that will output a 1 if the majority of the inputs are 1's otherwise the output is zero

Step 1 (Draw a truth Table)

A	B	C	X	
0	0	0	0	X_0
0	0	1	0	X_1
0	1	0	0	X_2
0	1	1	1	X_3
1	0	0	0	X_4
1	0	1	1	X_5
1	1	0	1	X_6
1	1	1	1	X_7

Boolean expression

Step 2 (Find the Boolean expression.)

There are two types of Boolean Expressions

1. SUM OF PRODUCTS (SOP's)

We form a product for each row for which the output is 1 and then add the products.

- A product is 1 if and only if all of its factors are 1
- A sum is 1 if at least one of its terms is a 1.

$X_3 = \overline{A}BC$; $X_5 = A\overline{B}C$; $X_6 = AB\overline{C}$; $X_7 = ABC$

$X = X_3 + X_5 + X_6 + X_7$

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

Boolean expression

2. PRODUCT OF SUMS (POS)

A sum is formed for each row for which the output is zero then the sums are multiplied.

- A sum is 0 if all its terms are 0's.
- A product is 0 if at least one of its factors is 0.

$X_0 = A + B + C$; $X_1 = A + B + \overline{C}$; $X_2 = A + \overline{B} + C$; $X_4 = \overline{A} + B + C$

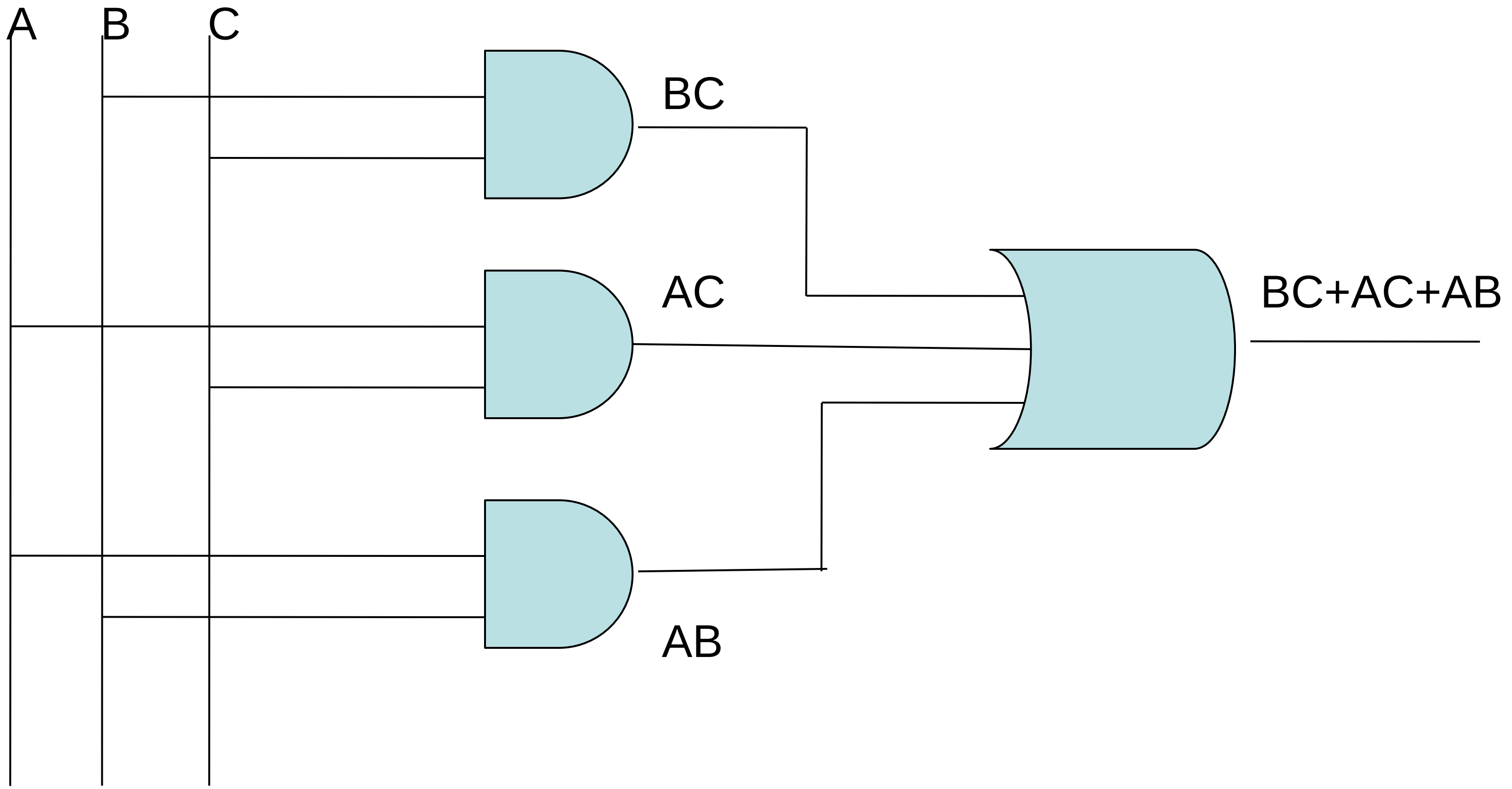
$$X = X_0X_1X_2X_4 = (A + B + C)(A + B + \overline{C})(A + \overline{B} + C)(\overline{A} + B + C)$$

Step 3 Simplify the expression

$$\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC = BC + AC + AB$$

Digital Design Process

- **Step 4: Draw the logic diagram**



MAXITERMS & MINITERMS

- The occurrence of a variable or its complement in an expression is called a **literal**.

- A term in the SUM OF PRODUCTS that includes a literal for every input is called a **miniterm**.

- A term in the PRODUCT OF SUMS that includes a literal for every input is called a **maxiterm**.

- e.g. in $A\overline{B}C + \overline{A}BC + A\overline{C}$; $\overline{A}BC$ and $\overline{A}BC$ are miniterms, $A\overline{C}$ is not a miniterm.

- Similarly in $(\overline{A}+B+\overline{C})(A+\overline{B}+\overline{C})(\overline{A}+C)$, $(\overline{A}+B+\overline{C})$ and $(A+\overline{B}+\overline{C})$ are

- maxiterms, $(\overline{A}+C)$ is not.

KARNAUGH MAPS

- A Karnaugh map is a truth table for a single output consisting of arrays of squares where each square corresponds to a row of a truth table. It is used in simplifying Boolean algebra expressions.
- The symbols at the top represent the variables associated with the columns and the symbols on the left represent the variables associated with the rows.
- The value of each output for each input is put in the corresponding square.
- For each 1 in the Karnaugh map there is a corresponding miniterm in the output’s Sum of product expression and each 0 represents a maxiterm in the Product of Sums expression.

EXAMPLES

Two inputs

A

B	0	1
0	0	2
1	1	3

Three Inputs

AB

C	00	01	11	10
0	0	2	6	4
1	1	3	7	5

Four Inputs

AB

CD	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Rules of Grouping in K-maps

1) A group must only contain 1s no 0s.
2) A group can only be horizontal or vertical not diagonal.
3) A group must contain 2^n 1s i.e 1,2,4,8.
4) Each group should be as large as possible.
5) Groups may overlap
6) Groups may wrap around a table.
7) Every 1 must be in at least one group.
8) There should be as few groups as possible.

EXAMPLE 1

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

Using Boolean Algebra

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

$$= \bar{A}BC + ABC + AB\bar{C} + ABC$$

$$\mathbf{= BC + AB}$$

Look for adjacent groups that include 2^n miniterms where n is an integer. The larger the group the greater is the reduction.

AB

C	00	01	11	10
0	0	0	1	0
1	0	1	1	0

AB

BC

EXAMPLE 2

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

Using Boolean simplification:

$\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$

$\overline{A}BC + ABC + A\overline{B}C + ABC \quad AB\overline{C} + ABC$

$=$ **BC + AC + AB**

		AB			
		00	01	11	10
C	0	0	0	1	0
	1	0	1	1	1

AB

AC

BC

A function may be used to state where the output is 1 instead of drawing a Karnaugh map. e.g. for the above example
$F(A,B,C) = \Sigma(3,5,6,7)$

EXAMPLE 3

$$F(A,B,C) = \Sigma(0,1,2,3,7)$$

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

$\bar{A}$

BC

More Examples

AB

C	00	01	11	10
0	1		1	1
1	1			1

$A\overline{C}$

$\overline{B}$

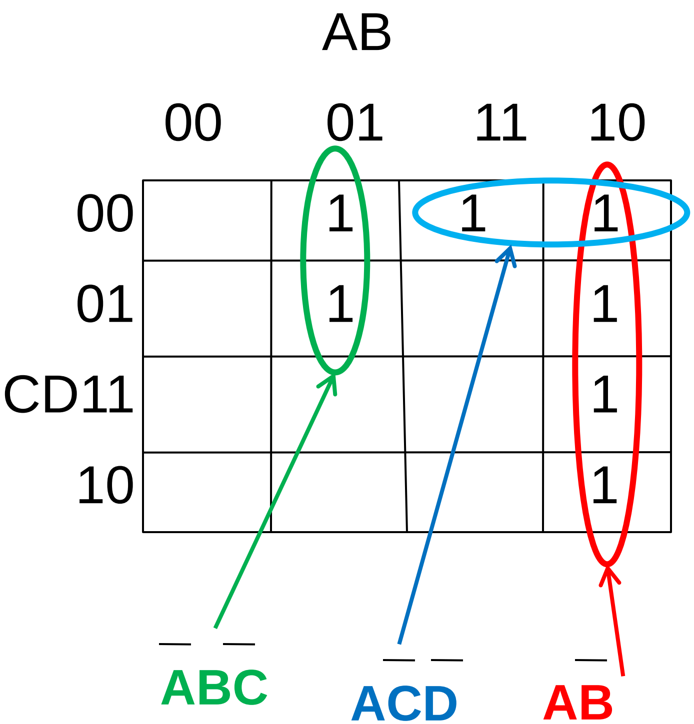
AB

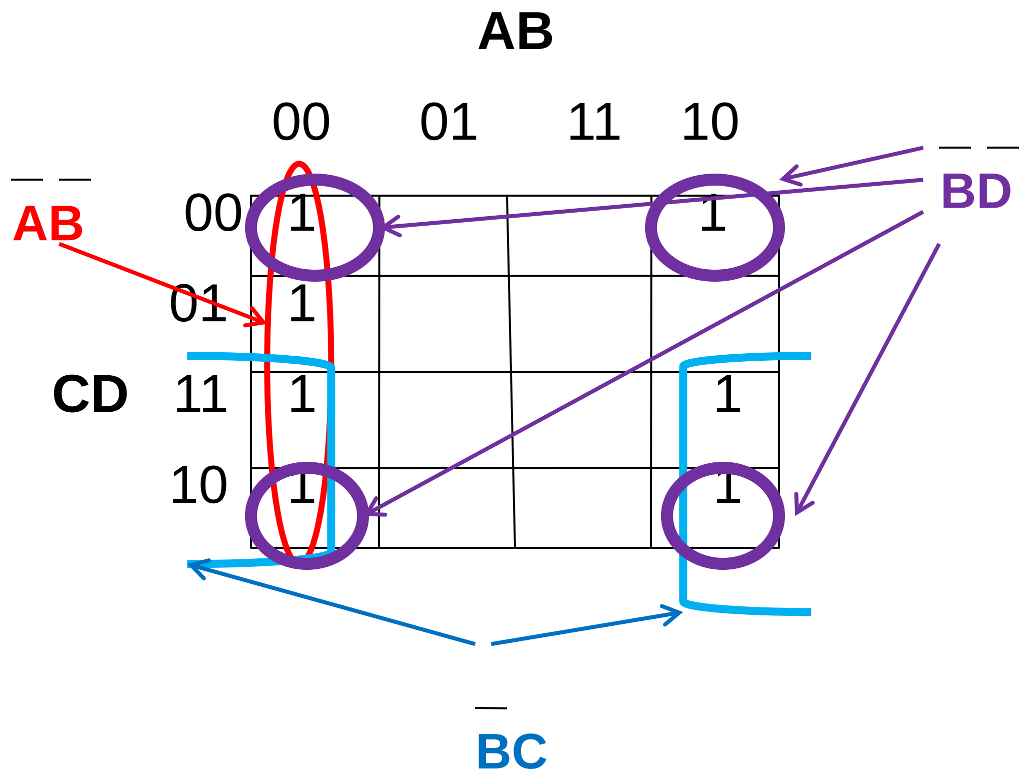
C	00	01	11	10
0	1			1
1		1		

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

$\overline{A}BC$

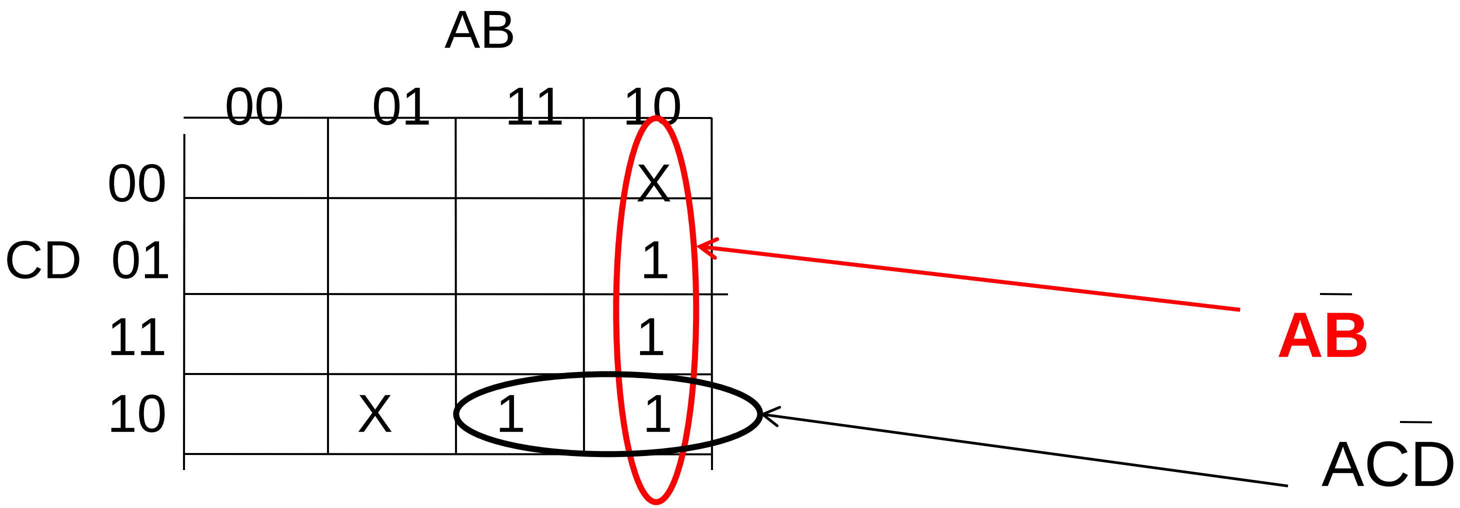
More Examples





THE DON'T CARE CASES

- For some designs some input combinations cannot occur. Their outputs are represented by X's in the Karnaugh Map and they may or may not be included in the prime implicants.
- They are called **Don't Care Case**s denoted by the function **d(A,B,C) = Σ(….)**



In the example above it is important to include the miniterm $A\bar{B}\bar{C}\bar{D}$ in but not $\bar{A}BC\bar{D}$

Reading Assignment 4:

- Read more about The Don’t care cases as used in Karnaugh maps.