Unit 04: Introduction to Digital Circuits

1. **Logic Gates [AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR]**
2. **Standard Logic Expression Forms:**
   - **Sum of Product (SOP) and**
   - **Product Of Sum (POS)**
3. **Logic Expression Realization & Minimization**
   - **using K-map (up to 4 variables only)**
4. **Adder Circuits:**
   - **Half Adder**
   - **Full Adder**
5. **Subtractor Circuits:**
   - **Half subtractor**
   - **Full subtractor**

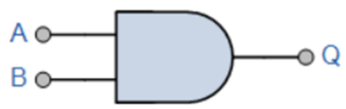
Logic Gates:

Logic gates are the basic building blocks of any digital system. It is a physical electronic device implementing a Boolean function. A logical operation is performed on one or more binary inputs which produces a single binary output. The relationship between the input and the output is based on a certain logic. Inputs & outputs of logic gate can occur only in two levels. These two levels are termed as HIGH and LOW or ON and OFF or simply 1 and 0. A table which lists all the possible combinations of input variables and the corresponding outputs is called as TRUTH TABLE. It shows how logic gate responds to various combinations of logic levels at the inputs.

The Logic AND Gate

The output of AND gate will be at logic 1 state, only when all the inputs are at logic 1 state. Otherwise, output will be at logic 0 state. Hence, AND gate is also called as *all or nothing gate.*

Symbol



Boolean Expression **Q = A.B**

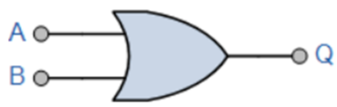
Truth Table

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

The Logic OR Gate

The output of OR gate will be at logic 1 state, even if one of its input is at logic 1 state. Its output will be at logic 0 state, only when all inputs are at logic 0 state. Hence, OR gate is also called as *any or all gate.*

Symbol



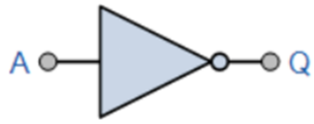
Boolean Expression Q = A + B

Truth Table

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

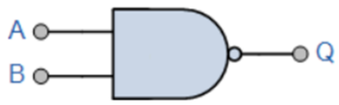
The Logic NOT Gate

A NOT gate, also called as an inverter, has only one input and one output. It is a device whose output is always the complement of its input.

Symbol	Truth Table	
	A	Q
	0	1
	1	0
Boolean Expression Q = not A or $\overline{A}$	Read as inverse of A gives Q	

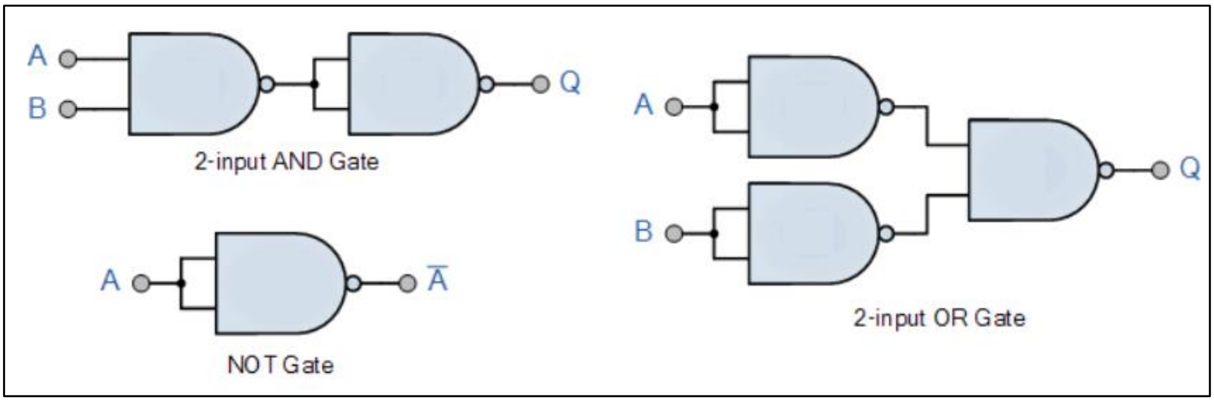
The Logic NAND Gate

NAND means NOT AND. It is combination of an AND gate and NOT gate. Its output will be at logic 0 state, only when all the inputs are at logic 1 state. Otherwise, output will be at logic 1 state.

Symbol	Truth Table		
	A	B	Q
	0	0	1
	0	1	1
	1	0	1
Boolean Expression $Q = \overline{A.B}$	1	1	0

The Universal NAND Gate

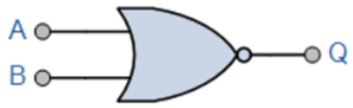
The Logic NAND Gate is generally classed as a "Universal" gate because it is one of the most commonly used logic gate types. NAND gates can also be used to produce any other type of logic gate function, and in practice the NAND gate forms the basis of most practical logic circuits. By connecting them together in various combinations the three basic gate types of AND, OR and NOT function can be formed using only NAND gates,



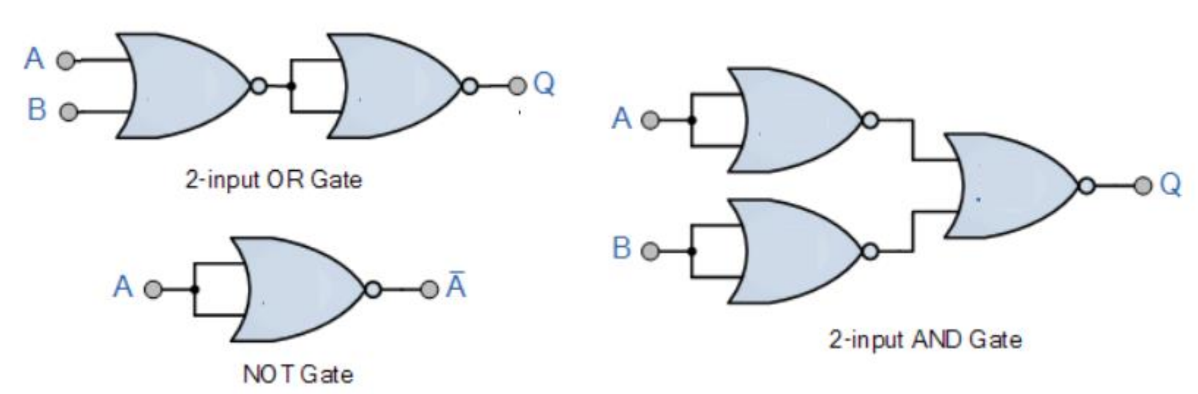
The Logic NOR Gate

NOR means NOT OR. It is combination of an OR gate and NOT gate. Its output will be at logic 1 state, only when all the inputs are at logic 0 state. Otherwise, output will be at logic 0 state.

The Universal NOR Gate

Symbol	Truth Table		
	A	B	Q
	0	0	1
	0	1	0
	1	0	0
Boolean Expression $Q = \overline{A + B}$	1	1	0

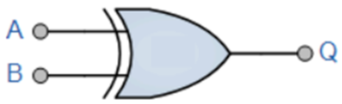
Like the NAND gate seen in the last section, the NOR gate can also be classed as a "Universal" type gate. NOR gates can be used to produce any other type of logic gate function just like the NAND gate and by connecting them together in various combinations the three basic gate types of AND, OR and NOT function can be formed using only NOR gates, for example.



The Logic Exclusive-OR [Ex-OR] Gate

An Ex-OR gate will have output at logic 1 state, only when one and only one of its two inputs is at logic 1 state. When both inputs are at same logic state then output will be at logic 0 state.

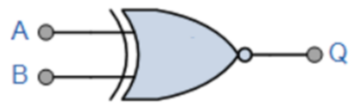
$$Q = A.\overline{B} + \overline{A}.B$$

Symbol	Truth Table		
	A	B	Q
	0	0	0
	0	1	1
	1	0	1
Boolean Expression $Q = A \oplus B$	1	1	0

The Logic Exclusive-NOR [Ex-NOR] Gate

An Ex-NOR gate is combination of an Ex-OR gate and a NOT gate. Its output will be at logic 0 state, only when one and only one of its two inputs is at logic 1 state. When both inputs are at same logic state then output will be at logic 1 state.

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

Symbol	Truth Table		
 Boolean Expression $Q = \overline{A \oplus B}$	A	B	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	1

Summary of all logic gates

The operation of the Digital Logic Gates can be summarized into a single truth table as shown below. This truth table shows the relationship between each output of the main digital logic gates for each possible input combination.

Inputs		Truth Table Outputs For Each Gate					
A	B	AND	NAND	OR	NOR	EX-OR	EX-NOR
0	0	0	**1**	0	**1**	0	**1**
0	1	0	**1**	**1**	0	**1**	0
1	0	0	**1**	**1**	0	**1**	0
1	1	**1**	0	**1**	0	0	**1**

Boolean Algebra

- Boolean algebra is a special branch of algebra which is mostly used in digital electronics. Boolean algebra was invented in the year of 1854, by an English mathematician George Boole.
- Boolean algebra is a method of simplifying the logic circuits in digital electronics. It is also called as Switching algebra
- In general mathematics, we represent the mathematical operations between algebraic variables by using mathematical operators like +, -, *, /. Similarly, in

Boolean algebra, we represent the Boolean operations by using logical operators like AND, OR, NOT operations.

- Complement means The reversal or inverse or opposite value. For example, if the variable is 1, then its complement will be 0. Similarly, if the variable is 0, then its complement will be 1. The complement variable is represented by a 'bar' on the variable.

Simplification of Boolean functions:

By using the Boolean theorems and Boolean laws, we can simplify the Boolean expressions, by which we can reduce the required number of logic gates to be implemented. We can simplify the Boolean function by using two methods,

- The algebraic method – by using identities (Boolean laws)
- The graphical method – by using Karnaugh Map (K-map) method

The K-map method is very easy to simplify a function than using identities.

Representation of Boolean Expression

A Boolean expression is an expression which consists of variables, constants (0 and 1) and logical operators which gives output as either 1 or 0.

Generally, given Boolean expression can be represented in two ways as:

1. **Sum-of-products (SOP) Form:** It is also called *Disjunctive Normal Form.* A sum-of-products form can be formed by adding (or summing) two or more product terms using a Boolean addition operation. Here, the product terms are defined by using the AND operation and the sum term is defined by using OR operation.

Examples: a] $f(A, B, C) = AB + \overline{B}C$ b] $f(A, B, C) = A + \overline{B}C + ABC$

2. **Product-of-sums (POS) Form:** It is also called *Conjunctive Normal Form*. A product-of-sums form can be formed by multiplying two or more sum terms using a Boolean multiplication operation. Here, the sum terms are defined by using the OR operation and product term is defined by using AND operation.

Examples: a] $f(A, B, C) = (A + B).(\overline{B} + C)$
b] $f(A, B, C) = A.(\overline{B} + C).(A + B + C)$

1. Standard Sum-of-products (SOP) Form:

- This form is also called as Disjunctive Canonical Form (DCF) or Canonical Sum-of-products form.
- In this form, the expression is the sum of a number of product terms where each product term contains all the variables of function either in complemented or uncomplemented form.
- It can be obtained from SOP form algebraically as shown below:

$$f(A, B, C) = \bar{A}B + \bar{B}C$$
$$f(A, B, C) = \bar{A}B(C + \bar{C}) + (A + \bar{A})\bar{B}C$$
$$f(A, B, C) = \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}C + \bar{A}\,\bar{B}C$$

- A product term which contains all the variables of function either in complemented or uncomplemented form is called a Minterm.
- An 'n' variable function can have 2^n minterms.
- A minterm is defined as the product term of 'n' variables, in which each of the 'n' variables will appear once either in its complemented or un-complemented form.
- The minterm is denoted as m_i where i is in the range of $0 \le i < 2^n$.
- A variable is in complemented form, if its value is assigned to 0, and the variable is un-complemented form, if its value is assigned to 1.
- For a 2-variable (A, B) Boolean function, possible minterms are: $2^2 = 4$

A	**B**	**Designation**	**Minterm**
0	0	m_0	$\bar{A}\bar{B}$
0	1	m_1	$\bar{A}B$
1	0	m_2	$A\bar{B}$
1	1	m_3	AB

- For a 3-variable (A, B, C) Boolean function, the possible minterms are: $2^3 = 8$

A	**B**	**C**	**Designation**	**Minterm**
0	0	0	m_0	$\bar{A}\,\bar{B}\bar{C}$
0	0	1	m_1	$\bar{A}\,\bar{B}C$
0	1	0	m_2	$\bar{A}\,B\bar{C}$
0	1	1	m_3	$\bar{A}\,BC$
1	0	0	m_4	$A\bar{B}\bar{C}$
1	0	1	m_5	$A\bar{B}C$
1	1	0	m_6	$AB\bar{C}$
1	1	1	m_7	ABC

For a 4-variable (A,B,C,D) Boolean function, the possible minterms are: 2^4 =16

A	**B**	**C**	**D**	**Minterm**
0	0	0	0	$m_0 = \bar{A}\,\bar{B}\bar{C}\bar{D}$
0	0	0	1	$m_1 = \bar{A}\,\bar{B}\bar{C}D$
0	0	1	0	$m_2 = \bar{A}\,\bar{B}C\bar{D}$
0	0	1	1	$m_3 = \bar{A}\,\bar{B}CD$
0	1	0	0	$m_4 = \bar{A}B\bar{C}\bar{D}$
0	1	0	1	$m_5 = \bar{A}B\bar{C}D$
0	1	1	0	$m_6 = \bar{A}BC\bar{D}$
0	1	1	1	$m_7 = \bar{A}BCD$

A	**B**	**C**	**D**	**Minterm**
1	0	0	0	$m_8 = A\bar{B}\bar{C}\bar{D}$
1	0	0	1	$m_9 = A\bar{B}\bar{C}D$
1	0	1	0	$m_{10} = A\bar{B}C\bar{D}$
1	0	1	1	$m_{11} = A\bar{B}CD$
1	1	0	0	$m_{12} = AB\bar{C}\bar{D}$
1	1	0	1	$m_{13} = AB\bar{C}D$
1	1	1	0	$m_{14} = ABC\bar{D}$
1	1	1	1	$m_{15} = ABCD$

Let, $f(A, B, C) = \bar{A}B + \bar{B}C = \bar{A}B(C + \bar{C}) + (A + \bar{A})\bar{B}C$
$f(A, B, C) = \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}C + \bar{A}\,\bar{B}C$

Another way of representing the function in standard or canonical SOP form is by showing the sum of minterms for which the function equals 1.

Thus,

$$f(A, B, C) = m_1 + m_2 + m_3 + m_5$$

Yet another way of representing the function in canonical form by listing the decimal codes of the minterms for which f = 1

Thus,

$$f(A, B, C) = \sum m(1,\ 2,\ 3,\ 5)$$

where, $\sum m$ represents the sum of all the minterms whose decimal codes are given in parenthesis.

Expansion of Boolean expression in SOP form to Standard SOP Form:

Steps to be followed in order to get Standard SOP form:

1. Write down all the terms
2. If one or more variables are missing in any term, expand that term by multiplying it with the sum of each one of missing variable and its complement.
3. Drop out the redundant terms

Example: Expand $\bar{A} + \bar{B}$ to minterms

Solution:

$$\begin{aligned} \bar{A} + \bar{B} &= \bar{A}(B + \bar{B}) + (A + \bar{A})\bar{B} \\ &= \bar{A}B + \bar{A}\bar{B} + A\bar{B} + \bar{A}\bar{B} \\ &= \bar{A}B + \bar{A}\bar{B} + A\bar{B} \\ &= \sum m(0,1,\ 2) \end{aligned}$$

<u>2. Standard Product-of-Sums (POS) Form:</u>

- This form is also called as Conjunctive Canonical Form (CCF) or Canonical Product-of-sum form.
- In this form, the expression is the product of a number of sum terms where each sum term contains all the variables of function either in complemented or uncomplemented form.
- It can be obtained from POS form algebraically as shown below:
  $f(A, B, C) = (\bar{A} + \bar{B})(A + B)$
  $f(A, B, C) = (\bar{A} + \bar{B} + C\bar{C})(A + B + C\bar{C})$
  $f(A, B, C) = (\bar{A} + \bar{B} + C)(\bar{A} + \bar{B} + \bar{C})(A + B + C)\ (A + B + \bar{C})$
- A sum term which contains all the variables of function either in complemented or uncomplemented form is called a Maxterm.
- An 'n' variable function can have 2^n maxterms.
- A maxterm is defined as the sum term of 'n' variables, in which each of the 'n' variables will appear once either in its complemented or un-complemented form.
- The maxterm is denoted as M_i where i is in the range of $0 \le i < 2^n$.

- A variable is in complemented form, if its value is assigned to 1, and the variable is un-complemented form, if its value is assigned to 0.
- For a 2-variable (A, B) Boolean function, possible maxterms are: $2^2 = 4$

A	B	Designation	Maxterms
0	0	M_0	$A + B$
0	1	M_1	$A + \bar{B}$
1	0	M_2	$\bar{A} + B$
1	1	M_3	$\bar{A} + \bar{B}$

For a 3-variable (A, B, C) Boolean function, the possible maxterms are: $2^3 = 8$

A	B	C	Designation	Maxterms
0	0	0	M_0	$A + B + C$
0	0	1	M_1	$A + B + \bar{C}$
0	1	0	M_2	$A + \bar{B} + C$
0	1	1	M_3	$A + \bar{B} + \bar{C}$
1	0	0	M_4	$\bar{A} + B + C$
1	0	1	M_5	$\bar{A} + B + \bar{C}$
1	1	0	M_6	$\bar{A} + \bar{B} + C$
1	1	1	M_7	$\bar{A} + \bar{B} + \bar{C}$

For a 4-variable (A,B,C,D) Boolean function, the possible maxterms are: $2^4 = 16$

A	B	C	D	Maxterm
0	0	0	0	$M_0 = A + B + C + D$
0	0	0	1	$M_1 = A + B + C + \bar{D}$
0	0	1	0	$M_2 = A + B + \bar{C} + D$
0	0	1	1	$M_3 = A + B + \bar{C} + \bar{D}$
0	1	0	0	$M_4 = A + \bar{B} + C + D$
0	1	0	1	$M_5 = A + \bar{B} + C + \bar{D}$
0	1	1	0	$M_6 = A + \bar{B} + \bar{C} + D$
0	1	1	1	$M_7 = A + \bar{B} + \bar{C} + \bar{D}$

A	B	C	D	Maxterm
1	0	0	0	$M_8 = \bar{A} + B + C + D$
1	0	0	1	$M_9 = \bar{A} + B + C + \bar{D}$
1	0	1	0	$M_{10} = \bar{A} + B + \bar{C} + D$
1	0	1	1	$M_{11} = \bar{A} + B + \bar{C} + \bar{D}$
1	1	0	0	$M_{12} = \bar{A} + \bar{B} + C + D$
1	1	0	1	$M_{13} = \bar{A} + \bar{B} + C + \bar{D}$
1	1	1	0	$M_{14} = \bar{A} + \bar{B} + \bar{C} + D$
1	1	1	1	$M_{15} = \bar{A} + \bar{B} + \bar{C} + \bar{D}$

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

Another way of representing the function in canonical POS form is by showing the product of maxterms for which the function equals 0.

Thus,

$$f(A, B, C) = M_0 + M_1 + M_6 + M_7$$

Yet another way of representing the function in canonical form by listing the decimal codes of the maxterms for which f = 0

Thus,

$$f(A, B, C) = \prod M(0,\ 1,\ 6,\ 7)$$

where, $\prod M$ represents the product of all the maxterms whose decimal codes are given in parenthesis.

Expansion of Boolean expression in POS form to Standard POS Form:

Steps to be followed in order to get Standard POS form:

1. Write down all the terms
2. If one or more variables are missing in any sum term, expand that term by adding the product of each one of missing variable and its complement.
3. Drop out the redundant terms

Example: Expand $A(\overline{B} + A)B$ to maxterms

Solution:

$$A(\overline{B} + A)B = (A + B)(A + \overline{B})(\overline{B} + A)(B + A)(B + \overline{A})$$
$$= (A + B)(A + \overline{B})(B + \overline{A})$$
$$= \prod M(0,1,\ 2)$$

Q.1 Expand $A + B\overline{C} + AB\overline{D} + ABCD$ to minterms and maxterms

Solution: The given expression is four variable function.

Therefore, by adding all missing variables in corresponding terms, we get

$A + B\overline{C} + AB\overline{D} + ABCD$

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

$= A(BC + B\overline{C} + \overline{B}C + \overline{B}\overline{C})(D + \overline{D}) + B\overline{C}(AD + A\overline{D} + \overline{A}D + \overline{A}\overline{D}) + AB\overline{D}C + AB\overline{D}\overline{C} + ABCD$

$= (ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\overline{C})(D + \overline{D}) + (ADB\overline{C} + A\overline{D}B\overline{C} + \overline{A}DB\overline{C} + \overline{A}\overline{D}B\overline{C}) + AB\overline{D}C$
$+ AB\overline{D}\overline{C} + ABCD$

$= ABCD + AB\overline{C}D + A\overline{B}CD + A\overline{B}\overline{C}D + ABC\overline{D} + AB\overline{C}\overline{D} + A\overline{B}C\overline{D} + A\overline{B}\overline{C}\overline{D} + AB\overline{C}D +$
$AB\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D} + ABCD$

$= ABCD + AB\overline{C}D + A\overline{B}CD + A\overline{B}\overline{C}D + ABC\overline{D} + AB\overline{C}\overline{D} + A\overline{B}C\overline{D} + A\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}B\overline{C}\overline{D}$

$= m_{15} + m_{13} + m_{11} + m_9 + m_{14} + m_{12} + m_{10} + m_8 + m_5 + m_4$

$= m_4 + m_5 + m_8 + m_9 + m_{10} + m_{11} + m_{12} + m_{13} + m_{14} + m_{15}$

$$\therefore A + B\overline{C} + AB\overline{D} + ABCD = \sum m(4,5,8,9,10,11,12,13,14,15)$$

In the SOP form the minterms 0,1,2,3,6 and 7 are missing. Hence, in POS form maxterms 0,1,2,3,6 and 7 will be present. Therefore, POS form is

$$\prod M(0,1,\ 2,\ 3,\ 6,\ 7)$$

Q.2 Expand $A(\bar{A}+B)(\bar{A}+B+\bar{C})$ to maxterms and minterms

Solution: The given expression is three variable function. Therefore, by adding all missing variables in corresponding terms, we get

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

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

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

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

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

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

$= M_0 + M_1 + M_2 + M_3 + M_4 + M_5$

$\therefore A(\bar{A}+B)(\bar{A}+B+\bar{C}) = \prod M(0,1,\ 2,\ 3,\ 4,\ 5)$

In the POS form the maxterms 6 and 7 are missing. Hence, in SOP form minterms 6 and 7 will be present. Therefore, SOP form is $\sum m(6,\ 7)$

Logic expression minimization using K-map

- Boolean expressions can be simplified algebraically, but it is not a systematic method and its effectiveness depends on familiarity & ability to apply Boolean algebraic rules, laws and theorems.
- The Karnaugh map (K-map) method on the other hand is a systematic method of simplifying Boolean expressions.
- The K-map is a chart or a graph, composed of an arrangement of adjacent cells, each representing a particular combination of variables in sum or product form. It is a mean to show the relationship between the logic inputs and desired output.
- An 'n' variable function can have 2^n possible combinations of product terms in SOP form or sum terms in POS form. Since, K-map is a graphical representation of Boolean expressions, two variable K-map will have $2^2 = 4$ cells or squares, a three variable K-map will have $2^3 = 8$ cells or squares and four variable function will have $2^4 = 16$ cells and so on.
- Any Boolean expressions can be expressed in a standard (canonical) sum-of-products (SOP) form or in standard (canonical) product-of-sums (POS) form.
- A standard SOP form is one in which a number of product terms, each one of which contains all the variables of the function either in complemented or uncomplemented form, are summed together. Each of the product terms in standard SOP form is called a minterm (m_0, m_1, m_2, m_3 … etc).
- A standard POS form is one in which a number of sum terms, each one of which contains all the variables of the function either in complemented or uncomplemented form, are multiplied together. Each of the sum terms in standard POS form is called a maxterm (M_0, M_1, M_2, M_3 … etc).
- A standard SOP form can be always converted to a standard POS form, by treating the missing minterms of SOP form as the maxterms of POS form. Similarly, standard POS form can be always converted to standard SOP form, by treating the missing maxterms of POS form as the minterms of SOP form.

Two Variables K-map: Mapping of SOP Expressions

A two variable has $2^2 = 4$ squares. These squares are called cells. Each square on the K-map represents unique minterm. The minterm designation of squares are shown in the figure:

A \ B	0	1
0	$\bar{A}\bar{B} = m_0$	$\bar{A}B = m_1$
1	$A\bar{B} = m_2$	$AB = m_3$

A '1' is placed in any square indicates that the corresponding minterm is included in the output expression and a '0' or no entry in any square indicates that the corresponding minterm does not appear in the expression for output.

The mapping of the expression $f = \sum m(0,\ 2,\ 3)$ is shown in figure

A \ B	0	1
0	1 [0]	0 [1]
1	1 [2]	1 [3]

Ex: Map the expression $f = \sum m(1,\ 2)$

⇨ The given expression in minterms is

$$f = \sum m(1,2) = m_1 + m_2$$

The K-map for the given function is shown in figure:

A \ B	0	1
0	0 [0]	1 [1]
1	1 [2]	0 [3]

Minimization of SOP Expressions

To minimize a boolean expression given in the SOP form by using K-map, we have to look for adjacent squares having 1s (minterms adjacent to each other) and combine them to form larger squares to eliminate some variables. Two squares are said to be adjacent to each other, if their minterms differ in only one variable. For example, in a two variable K-map, m_0 and m_1(i.e., $\bar{A}\bar{B}$ and $\bar{A}B$) differ in only one variable B ($\bar{A}$ is common to both of them).

Similarly, minterms m_0 ($\bar{A}\bar{B}$) and m_2 ($A\bar{B}$); m_1 ($\bar{A}B$) and m_3 (AB); m_2 ($A\bar{B}$) and m_3 (AB) are adjacent to each other. However, minterms m_0 ($\bar{A}\bar{B}$) and m_3 (AB); m_1 ($\bar{A}B$) and m_2 ($A\bar{B}$) are not adjacent to each other, because they differ in more than one variable. Two minterms, which are adjacent to each other, can be combined to form a bigger square called a 2-square or a pair. This eliminates one variable – the variable that is not common to both the minterms.

m_0, m_1 can be combined to yield, $f_1 = m_0 + m_1 = \bar{A}\bar{B} + \bar{A}B = \bar{A}(B + \bar{B}) = \bar{A}$
m_0, m_2 can be combined to yield, $f_2 = m_0 + m_2 = \bar{A}\bar{B} + A\bar{B} = \bar{B}(\bar{A} + A) = \bar{B}$

m_1, m_3 can be combined to yield, $f_3 = m_1 + m_3 = \bar{A}B + AB = B(\bar{A} + A) = B$

m_2, m_3 can be combined to yield, $f_4 = m_2 + m_3 = A\bar{B} + AB = A(\bar{B} + B) = A$

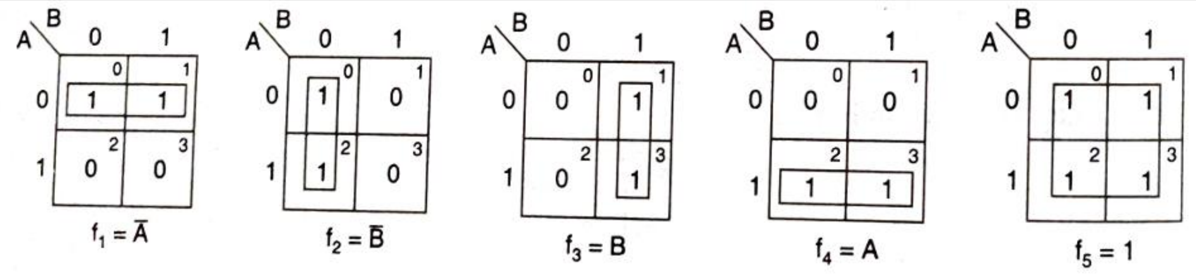
Also, m_0, m_1, m_2, m_3 can be combined to yield,

$$f_5 = m_0 + m_1 + m_2 + m_3$$

$$f_5 = \overline{A}\overline{B} + \overline{A}B + A\overline{B} + AB$$

$$f_5 = \overline{A}(\overline{B} + B) + A(\overline{B} + B)$$

$$f_5 = \overline{A} + A = 1$$



The possible minterm groupings in a two-variable K-map

- Two pairs adjacent to each other can be combined to form a 4-square or quad. A quad can eliminate 2 variables.
- To read the map after minimization, consider only those variables which remain constant throughout the square and ignore the variable which is varying.
- Write the uncomplemented variable if variable is remaining constant as 1 and the complemented variable if the variable is remaining constant as 0 and write variables in product term.
- In the figure, f_1 is read as $\overline{A}$, because along the square, A remains constant as 0 whereas B is changing from 0 to 1. f_3 is read as B, because along the square, B remains constant as 1 whereas A is changing from 0 to 1. f_5 is read as 1, because, no variable remains constant throughout the square. Which means output remains 1 for any combination of inputs.

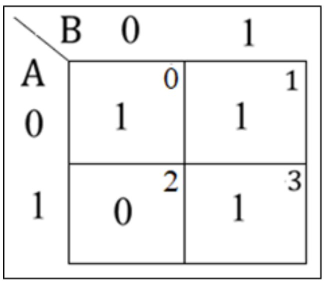
Example: Reduce the expression $f = \overline{A}\overline{B} + \overline{A}B + AB$ using K-map.

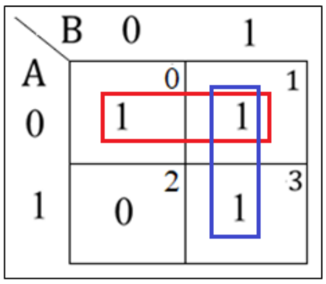
Solution: The given expression can be written in terms of minterms as

$$f = m_0 + m_1 + m_3 = \sum m(0,1,3)$$

Following figure shows K-map for function f and its reduction:

In one pair (indicated by red colour), A is constant as 0 but B is varying from 0 to 1 whereas in other pair (indicated by blue colour), B is constant as 1 but A is varying from 0 to 1. So, the reduced equation becomes $\overline{A} + B$





Two Variables K-map: Mapping of POS Expressions

A two variable has $2^2 = 4$ squares. These squares are called cells. Each square on the K-map represents unique maxterm. The maxterm designation of squares are shown in fig: A '0' is placed in any square indicates that the corresponding maxterm is included in the output expression and a '1' or no entry in any square indicates that them corresponding maxterm does not appear in the expression for output.

A \ B	0	1
0	$A+B=M_0$	$A+\overline{B}=M_1$
1	$\overline{A}+B=M_2$	$\overline{A}+\overline{B}=M_3$

The mapping of the expression $f = (A + B)(\overline{A} + B)(\overline{A} + \overline{B})$
$= \prod M(0,\ 2,\ 3)$ is shown in figure

A \ B	0	1
0	0 (0)	1 (1)
1	0 (2)	0 (3)

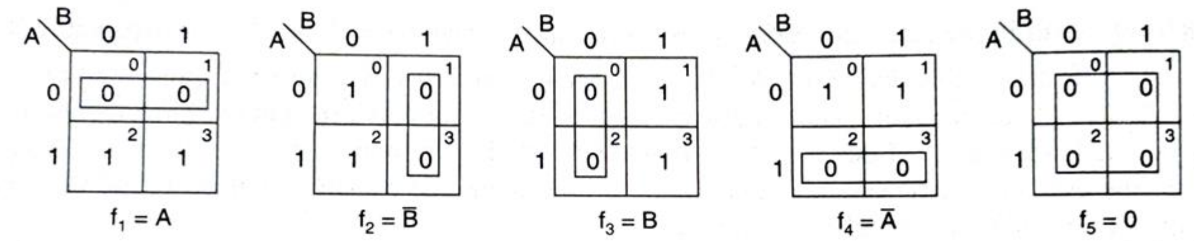
Minimization of POS Expressions

To minimize a Boolean expression given in the POS form by using K-map, we have to look for adjacent squares having 0s, that is, maxterms adjacent to each other and combine them to form larger squares to eliminate variable(s). Two squares are said to be adjacent to each other, if their maxterms differ in only one variable. For example, in a two variable K-map, M_0 and M_1(i.e., A+B and $A + \overline{B}$) differ in only one variable B (A is common to both of them).

Similarly, maxterms M_0 $(A + B)$ and M_2 $(\overline{A} + B)$; M_1 $(A + \overline{B})$ and M_3 $(\overline{A} + \overline{B})$; M_2 $(\overline{A} + B)$ and M_3 $(\overline{A} + \overline{B})$ are adjacent to each other. However, maxterms M_0 $(A + B)$ and M_3 $(\overline{A} + \overline{B})$; M_1 $(A + \overline{B})$ and M_2 $(\overline{A} + B)$ are not adjacent to each other, because they differ in more than one variable. Two maxterms, which are adjacent to each other, can be combined to form a bigger square called a 2-square or a pair. This eliminates one variable – the variable that is not common to both the maxterms.

M_0, M_1 can be combined to yield, $f_1 = M_0.M_1 = (A + B)(A + \overline{B}) = A$
M_1, M_3 can be combined to yield, $f_2 = M_1.M_3 = (A + \overline{B})(\overline{A} + \overline{B}) = \overline{B}$
M_0, M_2 can be combined to yield, $f_3 = M_0.M_2 = (A + B)(\overline{A} + B) = B$
M_2, M_3 can be combined to yield, $f_4 = M_2.M_3 = (\overline{A} + B)(\overline{A} + \overline{B}) = \overline{A}$

Also, M_0, M_1, M_2, M_3 can be combined to yield,
$f_5 = M_0.M_1.M_2.M_3 = (A + B)(A + \overline{B})(\overline{A} + B)(\overline{A} + \overline{B})$
$= (AA + A\overline{B} + BA + B\overline{B})(\overline{A}\overline{A} + \overline{A}\overline{B} + B\overline{A} + B\overline{B})$
$= (A + A\overline{B} + BA)(\overline{A} + \overline{A}\overline{B} + B\overline{A})$
$= A\overline{A} = 0$



The possible maxterm groupings in a two-variable K-map.

Example: Reduce the expression $\mathbf{f = (A + B)(A + \overline{B})(\overline{A} + \overline{B})}$ using K-map.
Solution: The given expression can be written in terms of maxterms as

$$f = M_0 . M_1 . M_3 = \prod M(0,1,3)$$

Following figure shows K-map for function f and its reduction:
In one pair (indicated by red colour), A is constant as 0 but B is varying from 0 to 1 whereas in other pair (indicated by blue colour), B is constant as 1 but A is varying from 0 to 1. So, the reduced equation becomes $A.\overline{B}$

<u>Three Variables K-map: Mapping of SOP Expressions</u>

A function in three variables (A,B,C) expressed in the standard SOP form can have 8 possible combinations: $\overline{A}\overline{B}\overline{C}, \overline{A}\overline{B}C, \overline{A}B\overline{C}, \overline{A}BC, A\overline{B}\overline{C}, A\overline{B}C, AB\overline{C}, ABC$. Each of these combinations are designated by $m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7$ respectively, is called a minterm. A is the MSB of minterm and C is LSB. Observe that the binary numbers along the top of K-map are not in normal binary order. They are, in fact, in grey code. This is to ensure that two physically adjacent cells are really adjacent i.e., their minterms differ by only one variable.

A \ BC	00	01	11	10
0	0: $\overline{A}\overline{B}\overline{C} = m_0$	1: $\overline{A}\,\overline{B}C = m_1$	3: $\overline{A}\,BC = m_3$	2: $\overline{A}B\overline{C} = m_2$
1	4: $A\overline{B}\overline{C} = m_4$	5: $A\overline{B}C = m_5$	7: $ABC = m_7$	6: $AB\overline{C} = m_6$

Example: Map the expression $\mathbf{f = \overline{A}\overline{B}C + A\overline{B}C + \overline{A}B\overline{C} + AB\overline{C} + ABC}$

Solution: In the given expression, minterms are:
$\overline{A}\overline{B}C = 001 = m_1$; $A\overline{B}C = 101 = m_5$; $\overline{A}B\overline{C} = 010 = m_2$;
$AB\overline{C} = 110 = m_6$; $ABC = 111 = m_7$

So the expression is, $f = m_1 + m_2 + m_5 + m_6 + m_7 = \sum m(1,2,5,6,7)$
The corresponding K-map is shown in the figure:

A \ BC	00	01	11	10
0	(0)	(1) 1	(3)	(2) 1
1	(4)	(5) 1	(7) 1	(6) 1

Three Variables K-map: Minimization of SOP Expressions

Generalized procedure to simplify Boolean expression is as follows:

1. Plot the K-map and place 1s corresponding to minterms of SOP expression
2. Check for quads (4 cells) and octets (8 cells) of adjacent 1s. They must geometrically form a square or rectangle.
3. Check for those 1s which is adjacent to only other 1 and make them pairs.
4. Check the K-map for 1s which are not adjacent to any other 1. They are isolated minterms. They are to be read as they are because they cannot be combined even into a pair.
5. Form minimal expression by summing the product terms of all the groups

Example: Obtain the minimal expression $f = \sum m(1, 2, 4, 6, 7)$ using K-map

Solution: The corresponding K-map is shown in the figure:

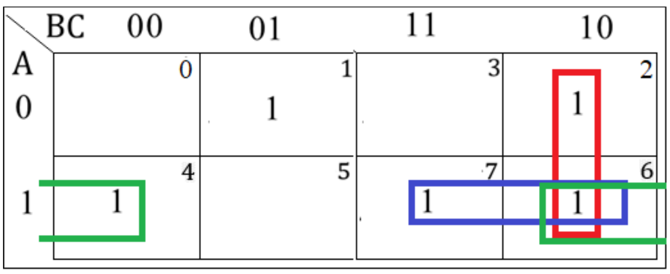
A \ BC	00	01	11	10
0	^{0}	1 ^{1}	^{3}	1 ^{2}
1	1 ^{4}	^{5}	1 ^{7}	1 ^{6}

1. A minterm m_1 has no adjacent cells and hence, it cannot form any pair/quad. Thus, it will remain as it is (i.e., for A=0, B=0 and C=1 minterm becomes $\overline{A}\overline{B}C$)

2. A pair of adjacent 1s can be combined by wrapping minterms m_2 & m_6. Along this pair, variable A is changing from 0 to 1 but, B and C have constant value of 1 & 0 respectively. Thus, variable A will be eliminated and B,C with values 1,0 will read as $B\overline{C}$.

3. A pair of adjacent 1s can be combined by wrapping minterms m_6 & m_7. Along this pair, variable C is changing from 0 to 1 but, A and B have constant value of 1 & 1 respectively. Thus, variable C will be eliminated and A,B with values 1,1 will read as AB.

4. A pair of adjacent 1s can be combined by wrapping minterms m_4 & m_6. Along this pair, variable B is changing from 0 to 1 but, A and C have constant value of 1 & 0 respectively. Thus, variable B will be eliminated and A,C with values 1,0 will read as $A\overline{C}$.
Hence, minimal/reduced form of given expression is $f = \overline{A}\overline{B}C + B\overline{C} + AB + A\overline{C}$



Example: Reduce the expression $f = \sum m(0, 2, 3, 4, 5, 6)$ using K-map

Solution: The corresponding K-map is shown in the figure:

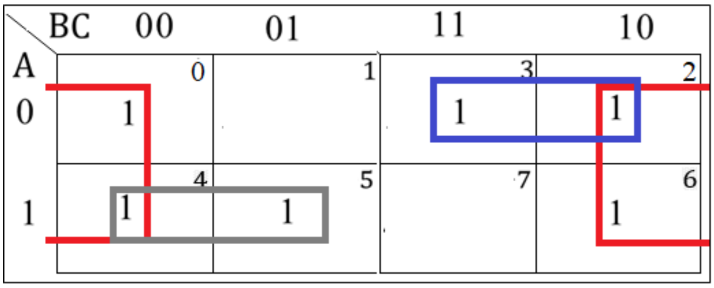
A \ BC	00	01	11	10
0	1 (0)	(1)	1 (3)	1 (2)
1	1 (4)	1 (5)	(7)	1 (6)

1. A quad of adjacent 1s can be combined by wrapping minterms m_0, m_2, m_4, m_6. Along this quad, variables A and B are changing from 0 to 1 but, C has constant value of 0. Thus, variables A, B will be eliminated and C with 0 will read as $\bar{C}$.

2. A pair of adjacent 1s can be combined by wrapping minterms m_4 & m_5. Along this pair, variable C is changing from 0 to 1 but, A and B have constant value of 1 & 0 respectively. Thus, variable C will be eliminated and A,B with values 1,0 will read as $A\bar{B}$.

3. A pair of adjacent 1s can be combined by wrapping minterms m_2 & m_3. Along this pair, variable C is changing from 0 to 1 but, A and B have constant value of 0 & 1 respectively. Thus, variable C will be eliminated and A,B with values 0,1 will read as $\bar{A}B$.

Hence, minimal/reduced form of given expression is $f = A\bar{B} + \bar{A}B + \bar{C}$



Three Variables K-map: Mapping of POS Expressions

A function in three variables (A,B,C) expressed in the standard POS form can have 8 possible combinations: $(A + B + C)$, $(A + B + \bar{C})$, $(A + \bar{B} + C)$, $(A + \bar{B} + \bar{C})$, $(\bar{A} + B + C)$, $(\bar{A} + B + \bar{C})$, $(\bar{A} + \bar{B} + C)$, $(\bar{A} + \bar{B} + \bar{C})$

Each of these combinations are designated by $M_0, M_1, M_2, M_3, M_4, M_5, M_6, M_7$ respectively, is called a maxterm. A is the MSB of maxterm and C is LSB.

Observe that the binary numbers along the top of K-map are not in normal binary order. This is to ensure that two physically adjacent cells are really adjacent i.e., their maxterms differ by only one variable.

A \ BC	00	01	11	10
0	[0] $A+B+C=M_0$	[1] $A+B+\bar{C}=M_1$	[3] $A+\bar{B}+\bar{C}=M_3$	[2] $A+\bar{B}+C=M_2$
1	[4] $\bar{A}+B+C=M_4$	[5] $\bar{A}+B+\bar{C}=M_5$	[7] $\bar{A}+\bar{B}+\bar{C}=M_7$	[6] $\bar{A}+\bar{B}+C=M_6$

Ex: Map $\mathbf{f = (A + B + C)(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + C)(\bar{A} + \bar{B} + \bar{C})\ (A + \bar{B} + \bar{C})}$

Solution: In the given expression, maxterms are:
$(A + B + C) = 000 = M_0$;$(\bar{A} + B + \bar{C}) = 101 = M_5$;$(\bar{A} + \bar{B} + C) = 110 = M_6$; $(\bar{A} + \bar{B} + \bar{C}) = 111 = M_7$; $(A + \bar{B} + \bar{C}) = 011 = M_3$
So the expression is, $f = M_0 . M_3 . M_5 . M_6 . M_7 = \prod M(0,3,5,6,7)$
The corresponding K-map is shown in the figure:

A \ BC	00	01	11	10
0	[0] 0	[1]	[3] 0	[2]
1	[4]	[5] 0	[7] 0	[6] 0

Three Variables K-map: Minimization of POS Expressions

Generalized procedure to simplify Boolean expression is as follows:

1. Plot the K-map and place 0s corresponding to maxterms of POS expression
2. Check for quads (4 cells) and octets (8 cells) of adjacent 0s. They must geometrically form a square or rectangle.
3. Check for those 0s which is adjacent to only other 0 and make them pairs.
4. Check the K-map for 0s which are not adjacent to any other 0. They are isolated maxterms. They are to be read as they are because they cannot be combined even into a pair.
5. Form minimal expression by multiplying the sum terms of all the groups

Example: Reduce the expression $\mathbf{f = \prod M(0, 1, 2, 3, 4, 7)}$ **using K-map**

Solution: The corresponding K-map is shown in the figure:

A \ BC	00	01	11	10
0	[0] 0	[1] 0	[3] 0	[2] 0
1	[4] 0	[5]	[7] 0	[6]

1. A quad of four adjacent 0s can be combined by wrapping maxterms M_0, M_1, M_2 & M_3. Along this quad, variables B and C are changing from 0 to 1 but, A has constant value of 0 respectively. Thus, variables B, C will be eliminated and A with value 0 will read as A.

2. A pair of adjacent 0s can be combined by wrapping maxterms M_4 & M_0. Along this pair, variable A is changing from 0 to 1 but, B and C have constant value of 0 & 0 respectively. Thus, variable A will be eliminated and B,C with values 0,0 will read as $(B + C)$

3. A pair of adjacent 1s can be combined by wrapping maxterms M_7 & M_3. Along this pair, variable A is changing from 0 to 1 but, B and C have constant value of 1 & 1 respectively. Thus, variable A will be eliminated and B,C with values 1,1 will read as $(\bar{B} + \bar{C})$

Hence, minimal/reduced form of given expression is $f = A(B + C)(\bar{B} + \bar{C})$

A \ BC	00	01	11	10
0	0 (0)	0 (1)	0 (3)	0 (2)
1	0 (4)	(5)	0 (7)	(6)

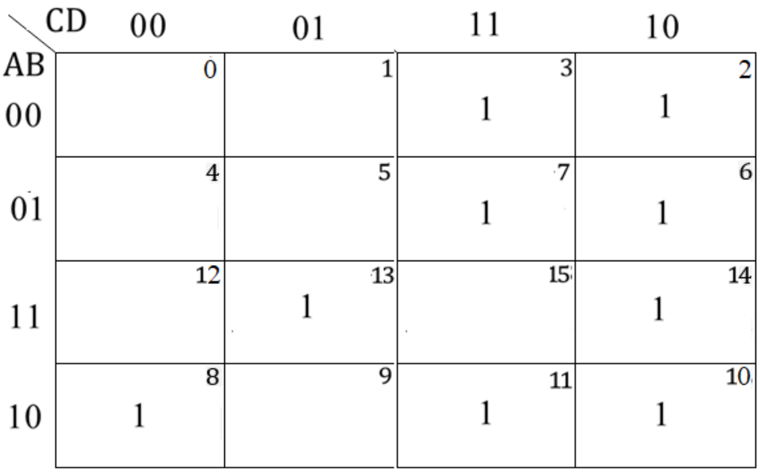
Four Variables K-map: Minimization of SOP Expressions

A function in four variables (A,B,C,D) expressed in the standard SOP form can have 16 possible combinations:

AB \ CD	00	01	11	10
00	0: $\bar{A}\bar{B}\bar{C}\bar{D} = m_0$	1: $\bar{A}\bar{B}\bar{C}D = m_1$	3: $\bar{A}\bar{B}CD = m_3$	2: $\bar{A}\bar{B}C\bar{D} = m_2$
01	4: $\bar{A}B\bar{C}\bar{D} = m_4$	5: $\bar{A}B\bar{C}D = m_5$	7: $\bar{A}BCD = m_7$	6: $\bar{A}BC\bar{D} = m_6$
11	12: $AB\bar{C}\bar{D} = m_{12}$	13: $AB\bar{C}D = m_{13}$	15: $ABCD = m_{15}$	14: $ABC\bar{D} = m_{14}$
10	8: $A\bar{B}\bar{C}\bar{D} = m_8$	9: $A\bar{B}\bar{C}D = m_9$	11: $A\bar{B}CD = m_{11}$	10: $A\bar{B}C\bar{D} = m_{10}$

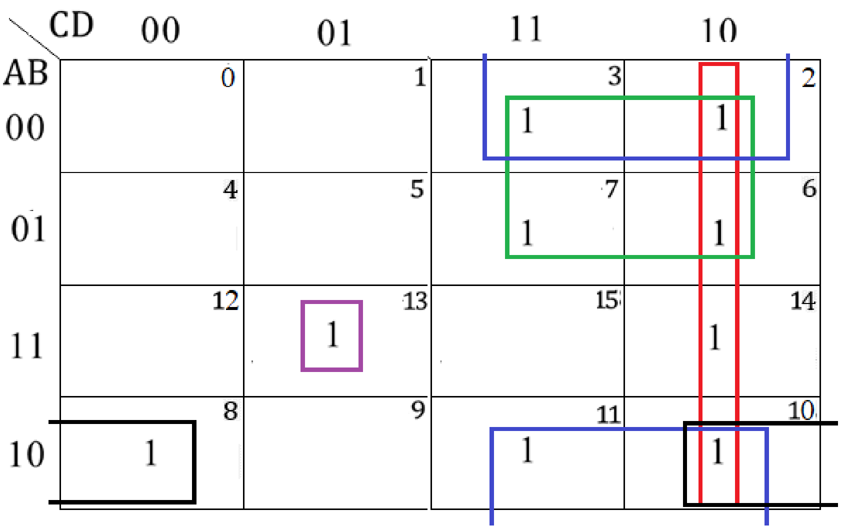
Q.1 Reduce the expression f = $\sum m(2, 3, 6, 7, 8, 10, 11, 13, 14)$ using K-map.

Solution: Given expression f = $\sum m(2,3,6,7,8,10,11,13,14)$ can be mapped as shown:



1. A minterm m_{13} has no adjacent cells and hence, it cannot form any pair/quad. Thus, it will remain as it is (for A=1, B=1, C=0, D=1 minterm becomes $AB\overline{C}D$)

2. A quad of adjacent 1s can be combined by wrapping minterms m_2, m_6, m_{14}, m_{10} Along this quad, variables A, B are changing from 0 to 1 but C, D have constant value of 1 & 0 respectively. Thus, variables A, B will be eliminated and C, D with values 1,0 can be read as $C\overline{D}$.



3. A quad of adjacent 1s can be combined by wrapping minterms m_2, m_3, m_6, m_7 Along this quad, variables B, D are changing from 0 to 1 but A, C have constant value of 0 & 1 respectively. Thus, variables B, D will be eliminated and A, C with values 0,1 can be read as $\overline{A}C$.

4. A quad of adjacent 1s can be combined by wrapping minterms m_2, m_3, m_{10}, m_{11}

Along this quad, variables A, D are changing from 0 to 1 but B, C have constant value of 0 & 1 respectively. Thus, variables A, D will be eliminated and B, C with values 0,1 can be read as $\overline{B}C$.

5. A pair of adjacent 1s can be combined by wrapping minterms m_8 & m_{10}
Along this pair, variable C is changing from 0 to 1 but, A, B, D have constant value of 1, 0, 0 respectively. Thus, variable C will be eliminated and A, B, D with values 1, 0, 0 will read as $A\overline{B}\overline{D}$. Hence, reduced form of given expression is $f = AB\overline{C}D + A\overline{B}\overline{D} + A\overline{C} + B\overline{C} + C\overline{D}$

Q.2 Reduce the expression $f = \sum m(0, 1, 2, 3, 5, 7, 8, 9, 10, 12, 13)$ using K-map.

➔ Given expression $f = \sum m(0,1,2,3,5,7,8,9,10,12,13)$ can be mapped as shown:

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

1. A quad of adjacent 1s can be combined by wrapping minterms m_1, m_3, m_5, m_7
Along this quad, variables B, C are changing from 0 to 1 but A, D have constant value of 0 & 1 respectively. Thus, variables B, C will be eliminated and A, D with values 0, 1 can be read as $\overline{A}D$.

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

2. A quad of adjacent 1s can be combined by wrapping minterms m_8, m_9, m_{12}, m_{13}
Along this quad, variables B, D are changing from 0 to 1 but A, C have constant value of 1 & 0 respectively. Thus, variables B, D will be eliminated and A, C with values 1, 0 can be read as $A\overline{C}$.
3. A quad of adjacent 1s can be combined by wrapping minterms m_0, m_2, m_8, m_{10}
Along this quad, variables A, C are changing from 0 to 1 but B, D have constant value of 0 & 0 respectively. Thus, variables A, C will be eliminated and B, D with values 0, 0 can be read as $\overline{B}\overline{D}$.
Hence, reduced form of given expression is $f = \overline{A}D + A\overline{C} + \overline{B}\overline{D}$

Four Variables K-map: Minimization of POS Expressions

A function in four variables (A,B,C,D) expressed in the standard POS form can have 16 possible maxterms:

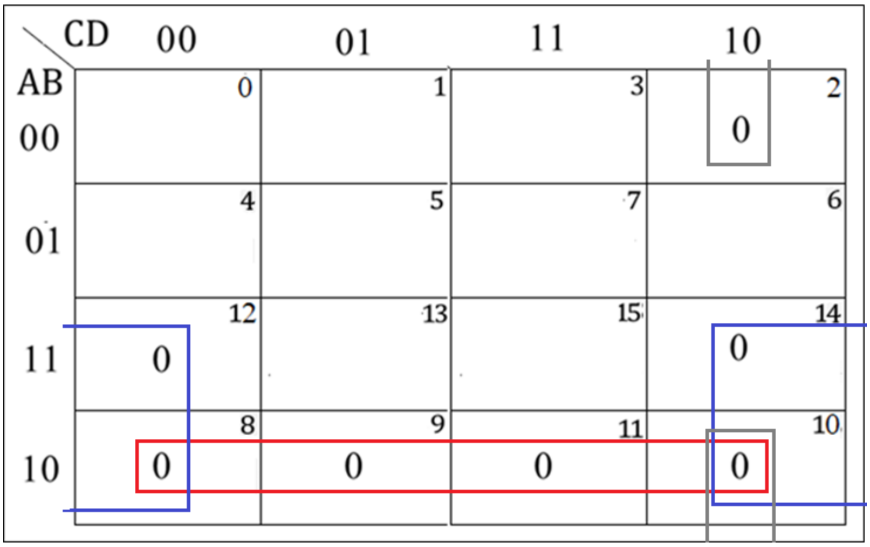
AB \ CD	00	01	11	10
00	0 $A+B+C+D = M_0$	1 $A+B+C+\overline{D} = M_1$	3 $A+B+\overline{C}+\overline{D} = M_3$	2 $A+B+\overline{C}+D = M_2$
01	4 $A+\overline{B}+C+D = M_4$	5 $A+\overline{B}+C+\overline{D} = M_5$	7 $A+\overline{B}+\overline{C}+\overline{D} = M_7$	6 $A+\overline{B}+\overline{C}+D = M_6$
11	12 $\overline{A}+\overline{B}+C+D = M_{12}$	13 $\overline{A}+\overline{B}+C+\overline{D} = M_{13}$	15 $\overline{A}+\overline{B}+\overline{C}+\overline{D} = M_{15}$	14 $\overline{A}+\overline{B}+\overline{C}+D = M_{14}$
10	8 $\overline{A}+B+C+D = M_8$	9 $\overline{A}+B+C+\overline{D} = M_9$	11 $\overline{A}+B+\overline{C}+\overline{D} = M_{11}$	10 $\overline{A}+B+\overline{C}+D = M_{10}$

Q.1 Reduce the expression $f = \prod M(2, 8, 9, 10, 11, 12, 14)$ using K-map

Solution: The corresponding K-map is shown in the figure:

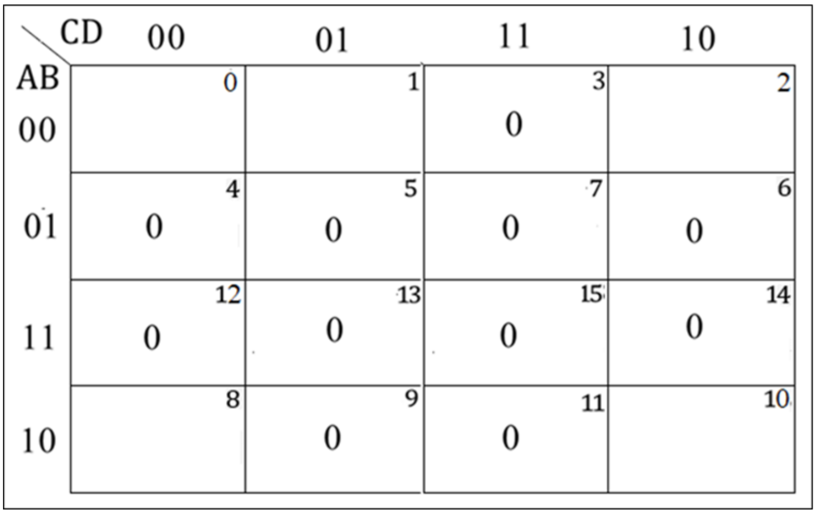
AB \ CD	00	01	11	10
00	0	1	3	2 0
01	4	5	7	6
11	12 0	13	15	14 0
10	8 0	9 0	11 0	10 0

1. A quad of adjacent 0s can be combined by wrapping maxterms M_8, M_9, M_{10}, M_{11}
Along this quad, variables C, D are changing from 0 to 1 but A, B have constant value of 1 & 0 respectively. Thus, variables C, D will be eliminated and A, B with values 1, 0 can be read as $(\bar{A} + B)$.



2. A quad of adjacent 0s can be combined by wrapping maxterms $M_8, M_{10}, M_{12}, M_{14}$
Along this quad, variables B, C are changing from 0 to 1 but A, D have constant value of 1 & 0 respectively. Thus, variables B, C will be eliminated and A, D with values 1, 0 can be read as $(\bar{A} + D)$.
3. A pair of adjacent 0s can be combined by wrapping maxterms M_2, M_{10}
Along this pair, variable A is changing from 0 to 1 but B, C, D have constant value of 0, 1 & 0 respectively. Thus, variable A will be eliminated and B, C, D with values 0, 1, 0 can be read as $(B + \bar{C} + D)$.
Hence, reduced form of given expression is $f = (\bar{A} + B)(\bar{A} + D)(B + \bar{C} + D)$.

Q.2 Reduce the expression $f = \prod M(3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15)$ using K-map
Solution: The corresponding K-map is shown in the figure:



1. An octet of adjacent 0s can be combined by wrapping maxterms $M_4, M_5, M_6, M_7, M_{12}, M_{13}, M_{14}, M_{15}$. Along this octet, variables A, C, D are changing from 0 to 1 but B has constant value of 1. Thus, variables A, C, D will be eliminated and B with values 1 can be read as $\overline{B}$.

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

2. A quad of adjacent 0s can be combined by wrapping maxterms $M_9, M_{11}, M_{13}, M_{15}$
Along this quad, variables B, C are changing from 0 to 1 but A, D have constant value of 1 & 1 respectively. Thus, variables B, C will be eliminated and A, D with values 1, 1 can be read as $(\overline{A} + \overline{D})$.
3. A quad of adjacent 0s can be combined by wrapping maxterms M_3, M_7, M_{11}, M_{15}
Along this quad, variables A, B are changing from 0 to 1 but C, D have constant value of 1 & 1 respectively. Thus, variables A, B will be eliminated and C, D with values 1, 1 can be read as $(\overline{C} + \overline{D})$.
Hence, reduced form of given expression is $f = \overline{B}(\overline{A} + \overline{D})(\overline{C} + \overline{D})$

Adders:

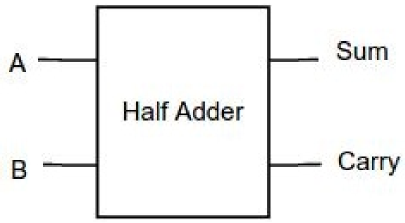
- Digital computer performs a variety of information processing tasks. Among the basic tasks encountered are various arithmetic operations. The most important arithmetic operation is the addition of two binary digits. This simple addition consists of 4 possible operations, namely:
  $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$ and $1 + 1 = 10$

- First three operations produce a sum whose length is one digit, but when two 1s are added, the binary sum consists of two digits. The higher significant bit of this result is called Carry.

- The carry obtained from the addition of two bits is added to the next higher-order pair of significant bits.

1. Half Adder

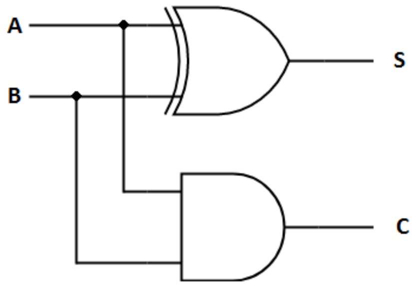
The half adder is a combinational circuit with two binary inputs and two binary outputs (Sum & Carry bits). It adds two inputs (A, B) and produces the sum (S) and carry (C) bits.



$$S = A\overline{B} + \overline{A}B \text{ and } C = A.B$$

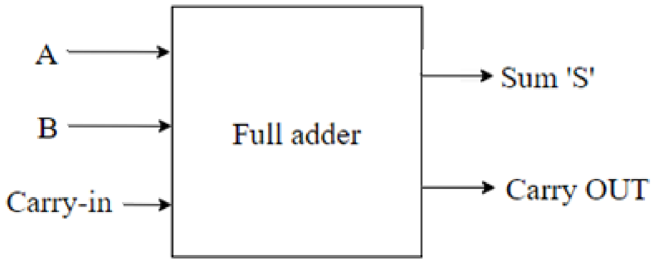
Truth Table:

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



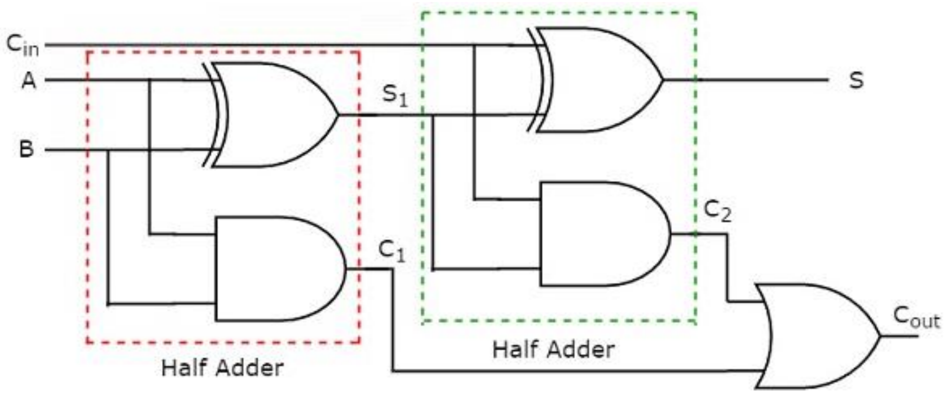
2. Full Adder

- The full adder is a combinational circuit that adds two bits and a carry, and its outputs are Sum & Carry. It adds three inputs (A, B and Cin) and produces the sum (S) and carry (C) bits.
- When two binary numbers each having two or more bits, the LSBs can be added by using half-adder. But the carry resulted in the binary addition is not taken into consideration in Half Adder.
- Full Adder adds the bits A and B along with the Carry from previous operation called Carry-in (Cin) and outputs the Sum bit (S) and the Carry bit Carry-out (Cout). The block diagram of full adder is shown as follows



- The circuit diagram of the Full Adder using two Ex-OR, two AND and one OR gate is shown in the figure. Also, it can be observed that Full adder can be realized using two half adders.
- Output expressions of full adder are:

$$S = A \oplus B \oplus C_{in} \quad \text{and} \quad C_{out} = AB + (A \oplus B)C_{in}$$



Truth Table

Inputs			Outputs	
A	**B**	**C_{in}**	**S**	**C_{out}**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Subtractor:

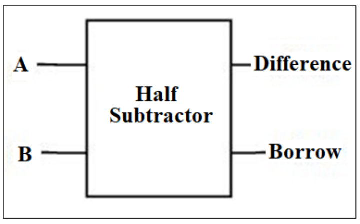
- The Subtractor is another type of combinational arithmetic circuit that produces an output which is the subtraction of two binary numbers. This process of subtraction is similar to arithmetic subtraction. The resultant terms can be denoted with the difference and borrow.
  0 - 0 = 0, 0 - 1 = 1, borrow 1 1 - 0 = 1 1 - 1 = 0

- Three operations (except second) produce a difference whose length is one digit, but when 1 is subtracted from 0, the binary subtraction consists of two bits. One is 'difference' and other one is 'borrow'

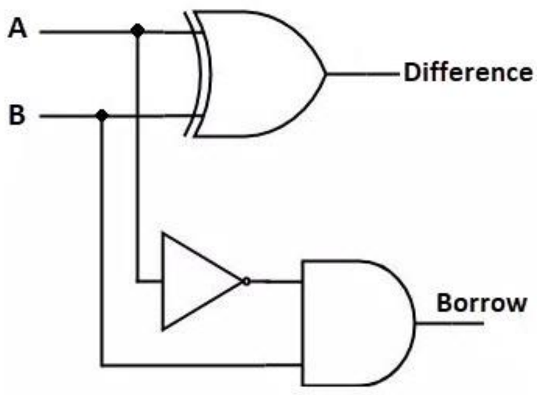
- The fact that 1 has been borrowed must be conveyed to the next higher pair of bits. For all the three remaining operations the value of 'borrow' will be zero.

1. Half Subtractor

The half subtractor is a combinational circuit with two binary inputs and two binary outputs (Difference & Borrow bits). It adds two inputs (A, B) and produces the difference (D) and borrow (B_{out}) bits.



$D = A \oplus B = A\overline{B} + \overline{A}B$ and $B_{Out} = \overline{A}.B$

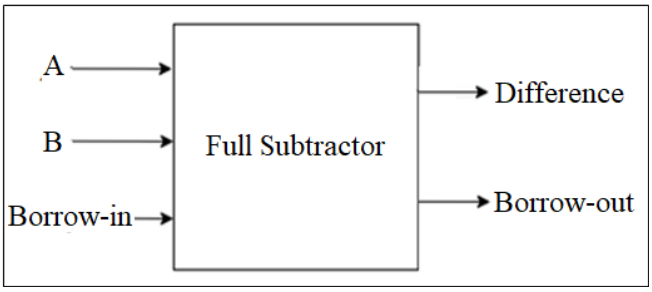


Truth Table:

A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

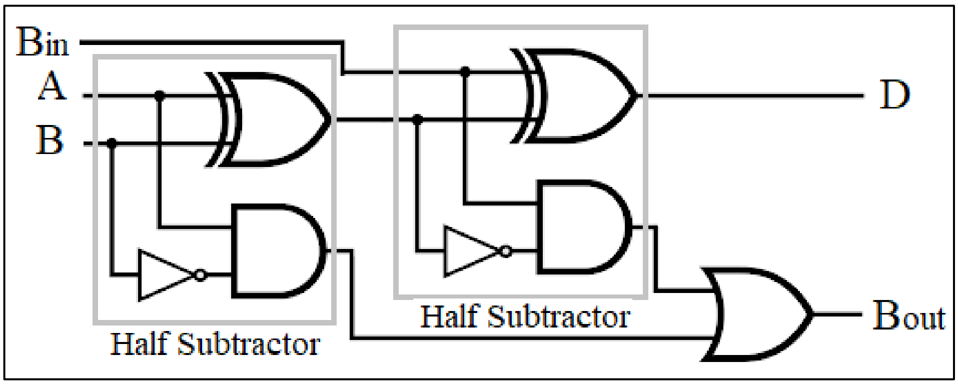
2. Full Subtractor

- The full subtractor is a combinational circuit that subtracts two bits and a borrow, and its outputs are Difference & Borrow. It adds three inputs (A, B and Bin) and produces the difference (D) and borrow (Bout) bits.
- When two binary numbers each having two or more bits, the LSBs can be subtracted by using half-subtractor. But, the borrow resulted in the binary subtraction is not taken into consideration in Half Subtractor.
- Full Subtractor subtracts the bits A and B along with the borrow from previous operation called Borrow-in (Bin) and outputs the difference bit (D) and the borrow bit Borrow-out (Bout).



- The circuit diagram of the Full Subtractor using two Ex-OR, two AND, two NOT and one OR gate is shown in the figure. Also, it can be observed that Full subtractor can be realized using two half subtractors.
- Output expressions of full subtractor are:

$$D = A \oplus B \oplus B_{in} \quad \text{and} \quad B_{out} = \overline{A}B + \overline{(A \oplus B)}\, B_{in}$$



Truth table:

Inputs			Outputs	
A	**B**	**B_{in}**	**D**	**B_{out}**
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Important Questions:

1. Reduce the expression $f = \sum m(0,2,3,4,5,6)$ using K-map
2. Obtain the minimal expression $f = \sum m(1,2,4,6,7)$ using K-map
3. Solve and reduce following boolean function using 4-variables K-map
   $f = \sum m\ (0,1,2,3,5,7,8,9,10,12,13)$
4. Solve and reduce following boolean function using 4-variables K-map
   $f = \prod M\ (2,8,9,10,11,12,14)$
5. Describe full adder circuit along with its truth table.
6. Explain the working of full subtractor with its truth table.