Digital Electronics (203105201)

Saurabh Srivastava, Assistant Professor
Mechatronics Engineering

CHAPTER-2

Minimization Techniques

Boolean Algebra, , Boolean postulates and laws, De-Morgan's Theorem, Principle of Duality, Boolean expression, Minterm, Maxterm, Sum of Products (SOP), Product of Sums (POS), K-map representation, simplification and minimization of logic functions using K-map. Don't care conditions and Quine-McCluskey method of minimization. Variable Entered Maps, Realizing Logic Function with Gates.

Minimization Techniques

Boolean Algebra: Branch of mathematics that deals with operations on **logical values** with binary variables. The Boolean variables are represented as binary numbers to represent truths: 1 = true and 0 = false.

Boolean Algebra is used to **analyze and simplify** digital (logic) circuits.

- In Boolean Algebra, the values of the variables are the truth values: true and false, usually denoted 1 and 0, whereas in elementary algebra the values of the variables are numbers.
- Boolean algebra uses logical operators such as **AND** (conjunction), **OR** (disjunction), and **NOT** (negation).
- Elementary algebra, on the other hand, uses arithmetic operators such as addition, multiplication, subtraction, and division.
- Boolean algebra is, therefore, a formal way of describing logical operations, in the same way that elementary algebra describes numerical operations.

Minimization Techniques

Some sets of rules and laws are used in Boolean algebra which are called '**Laws of Boolean algebra**'.

These laws and rules are used to reduce the number of gates used in logic operations.

In Boolean algebra, the alphabets A, B, C, … are used as variables and their values can be 1 or 0.

1) Commutative law:

a) A **OR** B=B **OR** A, i.e., **A + B = B + A**

b) A **AND** B=B **AND** A, i.e., **A*B = B*A**

2) Associative law:

a) (A+B)+C= A+(B+C)

b) (A*B)*C= A*(B*C)

Minimization Techniques

3) Distributive law:

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

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

4) Identity law:

a) $A+0 = A$

b) $A*1 = A$

5) Complement law:

a) $A+\bar{A}=1$

b) $A* \bar{A} =0$

6) Idempotent law

a) $A+A=A$

b) $A*A=A$

Minimization Techniques

7) Absorption law

a) A*(A+B)=A
b) A+(A*B)=A

8) Double negation law

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

9) Annulment law

a) A+1= 1
b) A*0= 0

De-Morgan's Theorem:

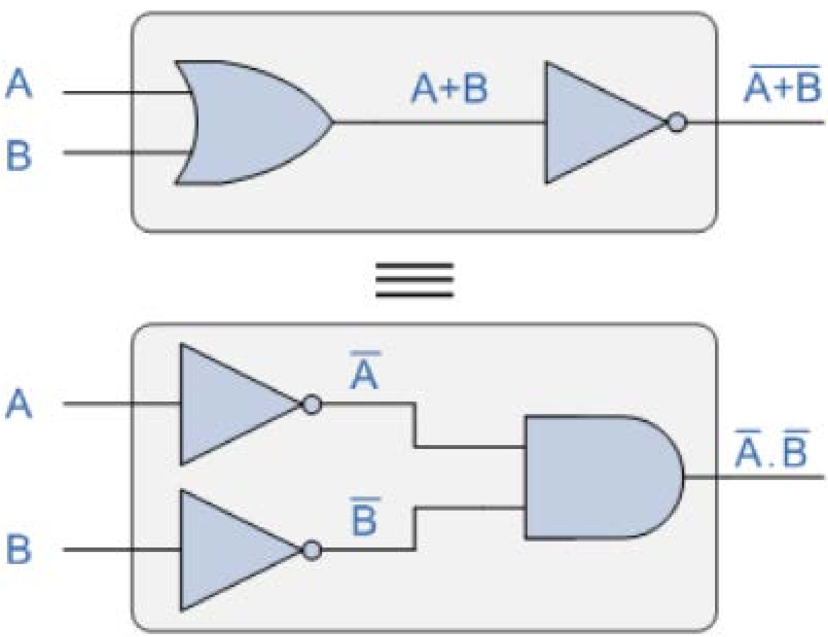
$$\overline{A + B} = \overline{A} * \overline{B}$$

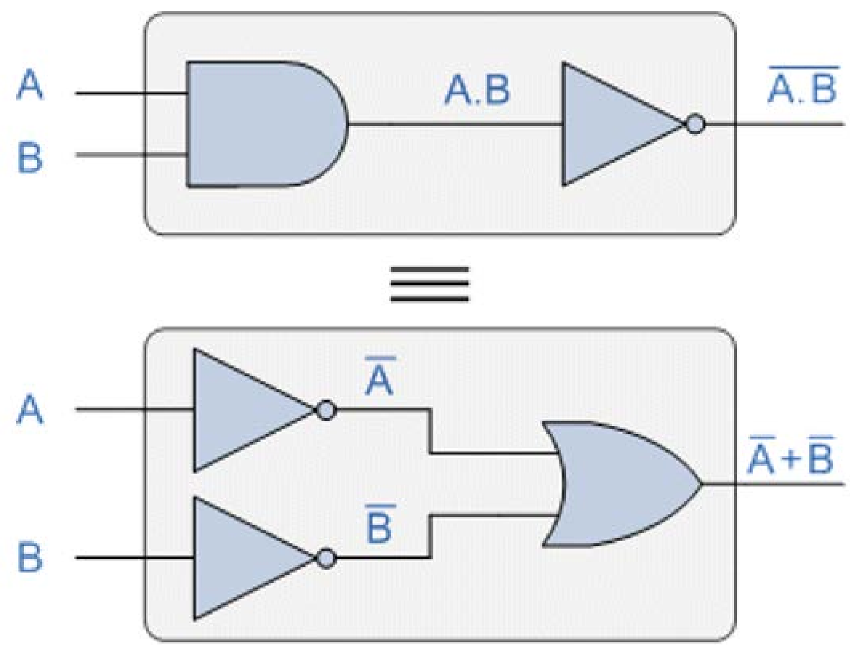
$$\overline{A * B} = \overline{A} + \overline{B}$$

Minimization Techniques

De-Morgan's Theorem:

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





Minimization Techniques

Proof:

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

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

Minimization Techniques

Principle of Duality

“The Dual of the expression can be achieved by replacing the AND operator with OR operator, along with the binary variables, such as replacing 1 with 0 and 0 with 1”.

- This law explains that replacing the variables doesn’t change the value of the Boolean function.
- While interchanging the names of the variables, must change the binary operators also.
- “If the operators and variables of an equation or function produce no change in the output of the equation, though they are interchanged are called “Duals”.

Operator/ Variable	**Dual**
AND	OR
OR	AND
1	0
0	1
A	A’
A’	A

Minimization Techniques

Sum of Products (SOP) & Product of Sums (POS)

The output Boolean variable of a digital system can be expressed in terms of input Boolean variables which form the 'Boolean Expression'.

- Representation of Boolean expression can be primarily done in two ways. They are as follows: **Sum of Products (SOP)** form & **Product of Sums (POS)** form

Sum of Products (SOP):
Way of writing a Boolean expression.
It is formed by adding (OR operation) the product terms.
These product terms are also called **'min-terms'.** Min-terms are represented with '**m**', they are **product (AND operation)** of Boolean variables either in normal form or complemented form.

```
In SOP FORM:
If variable A is Low(0) : A'
If variable A is High(1): A
```

Minimization Techniques

Sum of Products (SOP)

Consider a 3-variable function with following truth table:

A	B	C	Y (suppose)
0	0	0	0
0	0	1	**1**
0	1	0	0
0	1	1	**1**
1	0	0	0
1	0	1	0
1	1	0	**1**
1	1	1	0

Min-term
m_0
$\boldsymbol{m_1}$
m_2
$\boldsymbol{m_3}$
m_4
m_5
$\boldsymbol{m_6}$
m_7

$$\boldsymbol{Y = m(001, 011, 110)}$$

$$= \sum_{\boldsymbol{m}}(\boldsymbol{1, 3, 6}) = \sum_{\boldsymbol{m}}(\boldsymbol{m_1, m_3, m_6})$$

$$\boldsymbol{Y = \overline{A}\,\overline{B}C + \overline{A}BC + AB\overline{C}}$$

```
In SOP FORM:
If variable A is Low(0) : A’
If variable A is High(1): A
```

Minimization Techniques

Product of Sums (POS)

Consider a 3-variable function with following truth table:

A	B	C	Y (suppose)	Max-term
0	0	0	0	M_0
0	0	1	1	M_1
0	1	0	0	M_2
0	1	1	1	M_3
1	0	0	0	M_4
1	0	1	0	M_5
1	1	0	1	M_6
1	1	1	0	M_7

$$\boldsymbol{Y} = \boldsymbol{M}(\mathbf{000}, \mathbf{010}, \mathbf{100}, \mathbf{101}, \mathbf{111})$$

$$= \prod_{\boldsymbol{M}}(\mathbf{0}, \mathbf{2}, \mathbf{4}, \mathbf{5}, \mathbf{7})$$

```
In POS FORM:
If variable A is Low(0) : A
If variable A is High(1): A'
```

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

Minimization Techniques

Sum of Products (SOP) & Product of Sums (POS)

S.No.	SOP	POS
1.	A way of representing boolean expressions as sum of product terms.	A way of representing boolean expressions as product of sum terms.
2.	SOP uses minterms. Minterm is product of boolean variables either in normal form or complemented form.	POS uses maxterms. Maxterm is sum of boolean variables either in normal form or complemented form.
3.	It is sum of minterms. Minterms are represented as ‘m’	It is product of maxterms. Maxterms are represented as ‘M’
4.	SOP is formed by considering all the minterms, whose output is HIGH(1)	POS is formed by considering all the maxterms, whose output is LOW(0)
5.	While writing minterms for SOP, input with value 1 is considered as the variable itself and input with value 0 is considered as complement of the input.	While writing maxterms for POS, input with value 1 is considered as the complement and input with value 0 is considered as the variable itself.

Minimization Techniques

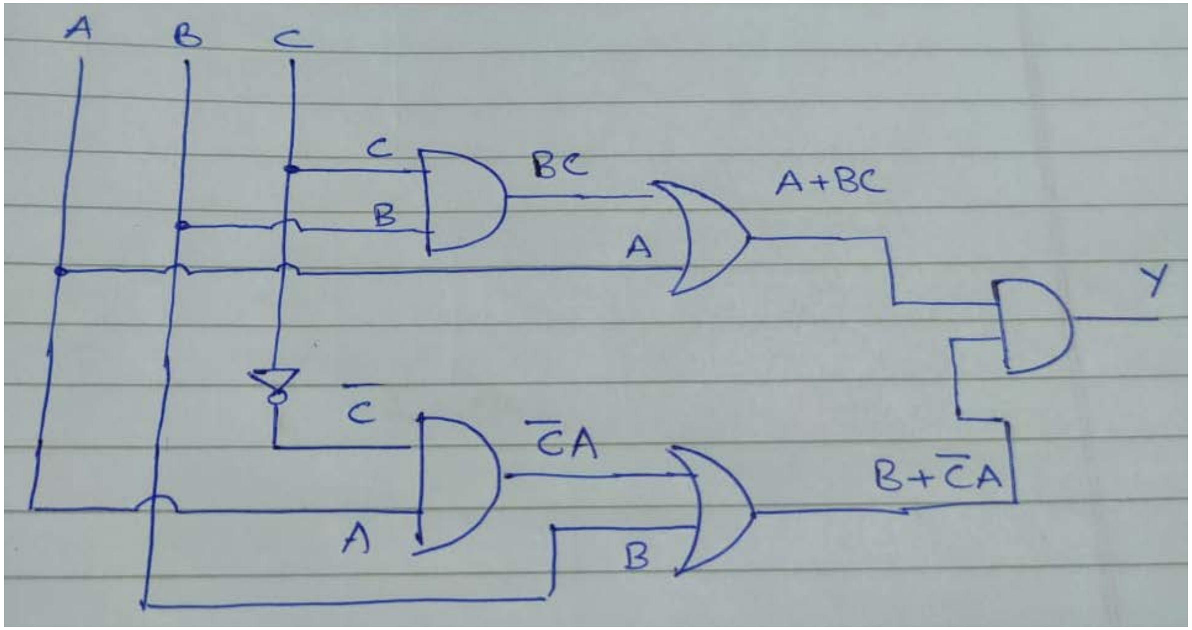
Realization:

Realize the logic equation: $Y = (A + BC)(B + \overline{C}A)$: POS Form

- **Requires one AND gate (inputs: $A + BC$, & $B + \overline{C}A$)**

$A + BC$: Requires one AND Gate (BC) followed by one OR Gate ($A + BC$)

$B + \overline{C}A$: Requires one NOT Gate ($\overline{C}$), one AND Gate ($\overline{C}A$), followed by one OR Gate ($B + \overline{C}A$)



Minimization Techniques

Example:

Logic equation: $Y = (A + BC)(B + \overline{C}A)$

- **The form is POS**

$$Y = (A)(B + \overline{C}A) + (BC)(B + \overline{C}A)$$
$$= AB + A\overline{C}A + BCB + BC\overline{C}A$$
$$= AB + A\overline{C} + BC + B.0.A$$
$$= AB(C + \overline{C}) + A(B + \overline{B})\overline{C} + (A + \overline{A})BC$$
$$= ABC + AB\overline{C} + AB\overline{C} + A\overline{B}\overline{C} + ABC + \overline{A}BC$$
$$= ABC + AB\overline{C} + A\overline{B}\overline{C} + \overline{A}BC$$

The form is SOP now (Canonical – involves all literals ABC)

$AB + A\overline{C} + BC$: ***Non-canonical SOP***

Minimization Techniques

K-Map representation:

In many digital circuits and practical problems, we need to find expressions with minimum variables.

- We can minimize Boolean expressions of 3, and 4 variables very easily using K-map without using any Boolean algebra theorems.
- K-map can take two forms Sum of Product (SOP) and Product of Sum (POS) according to the need of the problem.
- K-map is a table-like representation but it gives more information than TRUTH TABLE.
- We fill a grid of K-map with 0’s and 1’s then solve it by making groups.

Just like the truth table, a K-map contains all the possible values of input variables and their corresponding output values. However, in K-map, the values are stored in cells of the array.

- In each cell, a binary value of each input variable is stored.

Minimization Techniques

K-Map representation:

Following steps are used to solve the expressions using a K-map:

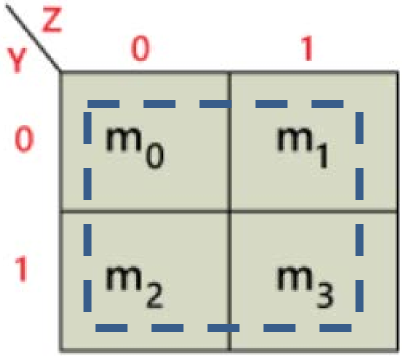
1. First, find the K-map as per the number of variables.
2. Find the maxterm and minterm in the given expression.
3. Fill cells of K-map for SOP with 1 respective to the minterms.
4. Fill cells of the block for POS with 0 respective to the maxterm.
5. Next, create rectangular groups that contain total terms in the power of two like 2, 4, 8, … and try to cover as many elements as we can in one group.
6. With the help of these groups, find the product terms and sum them up for the SOP form.

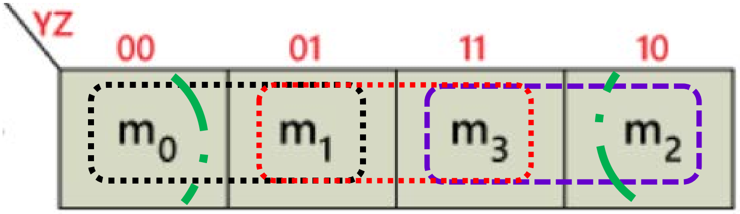
Minimization Techniques

K-Map representation:

Example: **2-Variable K-map**

There is a **total of 4 cells** in a 2-variable K-map.





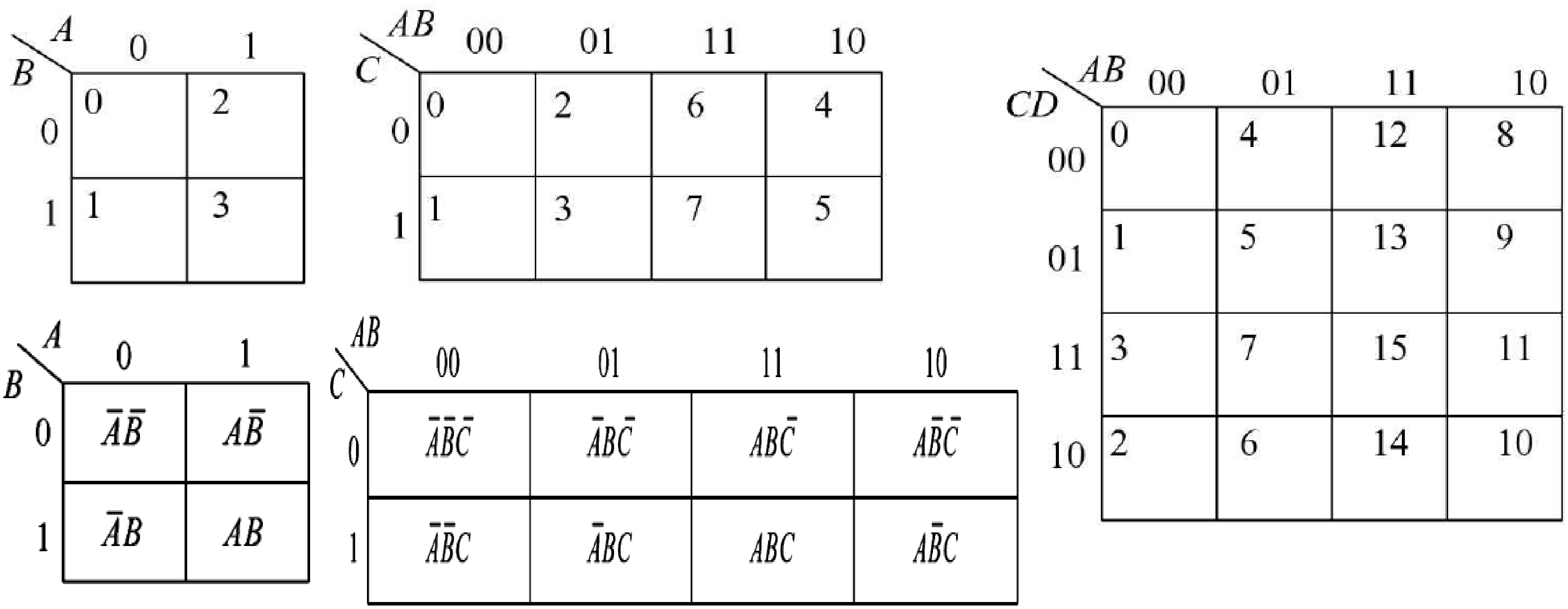
Each square (called a cell) corresponds to a minterm. Cells are called adjacent if the minterms that they represent differ in exactly one literal.

In the above figure, there is only one possibility of grouping four adjacent minterms.
The possible combinations of grouping 2 adjacent minterms are {(m_0, m_1), (m_2, m_3), (m_0, m_2) and (m_1, m_3)}.

Minimization Techniques

K-Map representation:

Example: **2-Variable, 3-Variable and 4-Variable K-maps**



Minimization Techniques

K-Map representation:

$B \backslash A$	0	1
0	$\bar{A}\bar{B}$	$A\bar{B}$
1	$\bar{A}B$	AB

SOP Form

$C \backslash AB$	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

$B \backslash A$	0	1
0	$A+B$	$\bar{A}+B$
1	$A+\bar{B}$	$\bar{A}+\bar{B}$

POS Form

$C \backslash AB$	00	01	11	10
0	$A+B+C$	$A+\bar{B}+C$	$\bar{A}+\bar{B}+C$	$\bar{A}+B+C$
1	$A+B+\bar{C}$	$A+\bar{B}+\bar{C}$	$\bar{A}+\bar{B}+\bar{C}$	$\bar{A}+B+\bar{C}$

Minimization Techniques

K-Map representation:

CD \ AB	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}B\bar{C}\bar{D}$	$AB\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$
01	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}B\bar{C}D$	$AB\bar{C}D$	$A\bar{B}\bar{C}D$
11	$\bar{A}\bar{B}CD$	$\bar{A}BCD$	$ABCD$	$A\bar{B}CD$
10	$\bar{A}\bar{B}C\bar{D}$	$\bar{A}BC\bar{D}$	$ABC\bar{D}$	$A\bar{B}C\bar{D}$

CD \ AB	00	01	11	10
00	$A+B+C+D$	$A+\bar{B}+C+D$	$\bar{A}+\bar{B}+C+D$	$\bar{A}+B+C+D$
01	$A+B+C+\bar{D}$	$A+\bar{B}+C+\bar{D}$	$\bar{A}+\bar{B}+C+\bar{D}$	$\bar{A}+B+C+\bar{D}$
11	$A+B+\bar{C}+\bar{D}$	$A+\bar{B}+\bar{C}+\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+\bar{D}$	$\bar{A}+B+\bar{C}+\bar{D}$
10	$A+B+\bar{C}+D$	$A+\bar{B}+\bar{C}+D$	$\bar{A}+\bar{B}+\bar{C}+D$	$\bar{A}+B+\bar{C}+D$

Minimization Techniques

K-Map rules

Rule-01:

- We can either group 0's with 0's or 1's with 1's but we can not group 0's and 1's together.
- X representing don't care can be grouped with 0's as well as 1's.

Rule-02:

- Groups may overlap each other.

Rule-03:

- We can only create a group whose number of cells can be represented in the power of 2.
- In other words, a group can only contain 2^n i.e. 1, 2, 4, 8, 16, and so on number of cells.

Rule-04:

- Groups can be only either horizontal or vertical.
- We can not create groups of diagonal or any other shape.

Minimization Techniques

K-Map rules

Rule-05:

- Each group should be as large as possible.

Rule-06:

- Opposite grouping and corner grouping are allowed.

Rule-07:

- There should be as few groups as possible.

Minimization Techniques

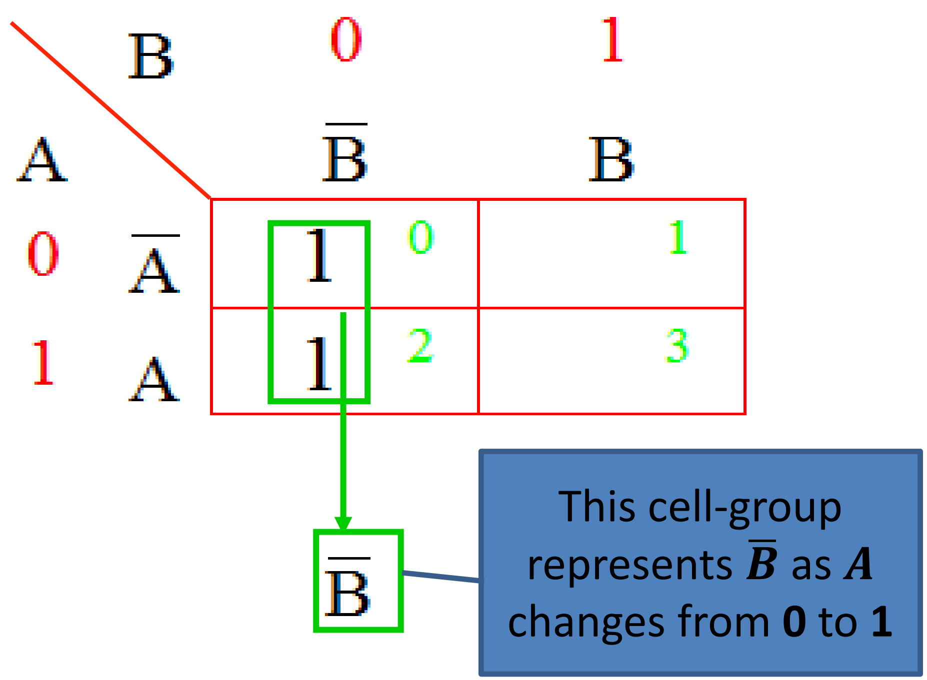
Consider a two-variable Karnaugh Map

If a minterm is present in F(A,B), then we place a 1 in the cell corresponding to the minterm, otherwise the cell is left empty or 0 is placed.

The resulting array is called a K-map corresponding to the expression

$$\boxed{F(A,B)=\bar{A}\,\bar{B}+A\,\bar{B}}$$

$$F(A,B)=\bar{A}\,\bar{B}+A\,\bar{B}$$
$$F(A,B)=\bar{B}\,(\bar{A}+A)$$
$$F(A,B)=\bar{B}$$



Minimization Techniques

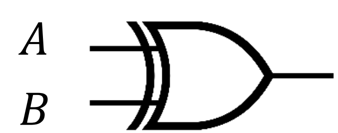
$$F(A,B) = \bar{A}B + A\bar{B}$$
$$= \sum m\,(1,2)$$

$$F(A,B) = 01 + 10$$

B / A	0 $(=\bar{B})$	1 $(=B)$
0 $(=\bar{A})$		1
1 $(=A)$	1	

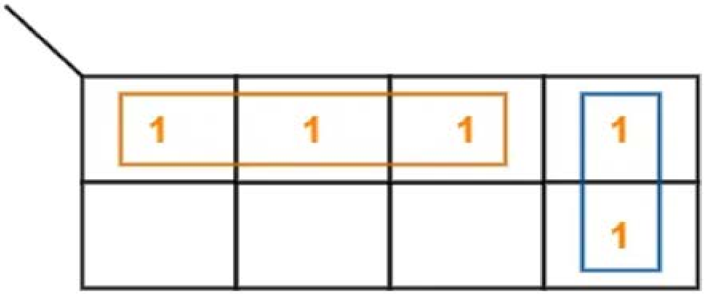
A	$\bar{A}$	B	$\bar{B}$	$\bar{A}B$	$A\bar{B}$	F
0		0				
0		1				
1		0				
1		1				

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

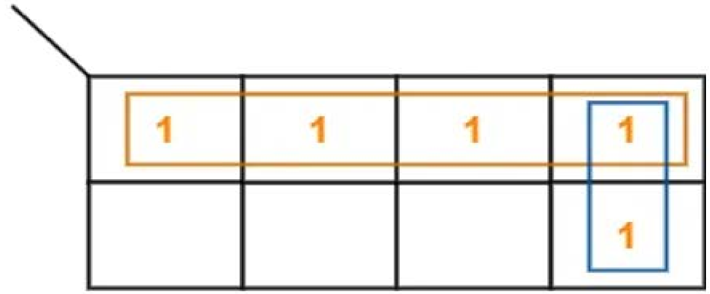


$F = A \oplus B$

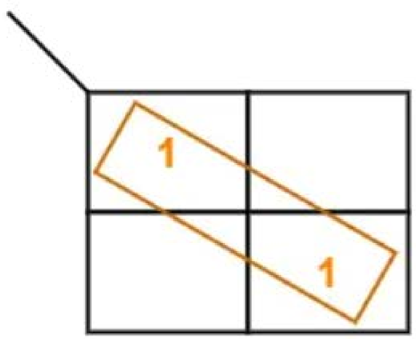
Minimization Techniques



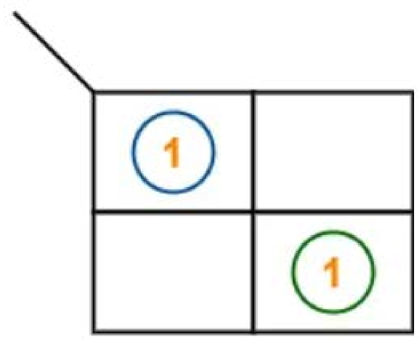
Incorrect



Correct

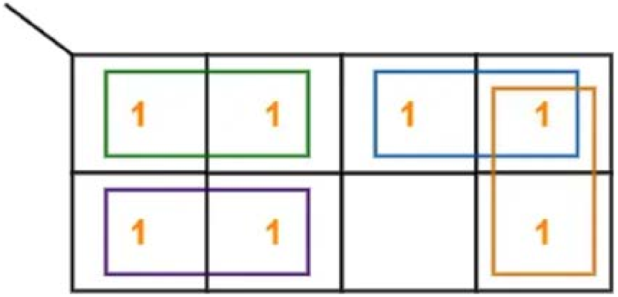


Incorrect



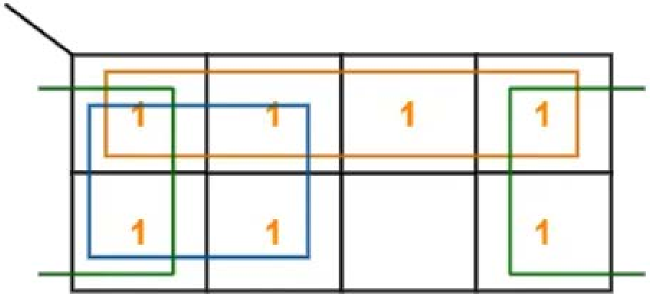
Correct

Minimization Techniques



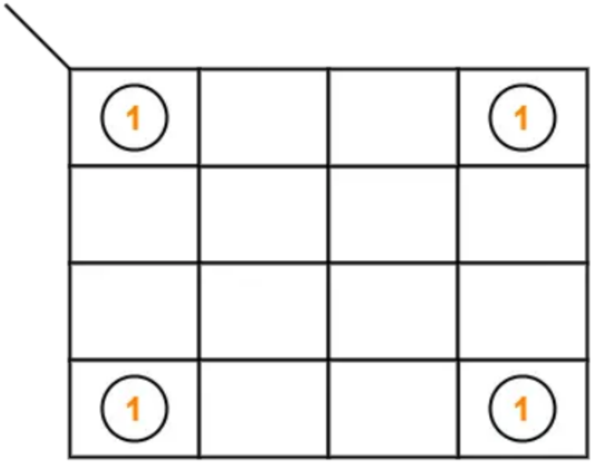
Incorrect

✗



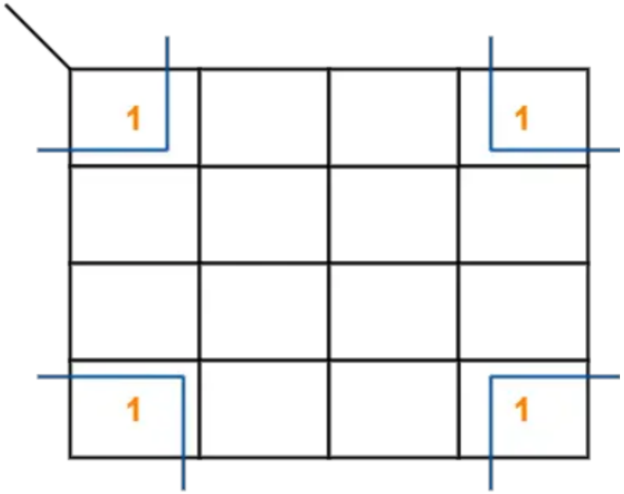
Correct

✓



Incorrect

✗



Correct

✓

Minimization Techniques

A	B	C	Cell #	Minterm
0	0	0	0	$\bar{A}\,\bar{B}\,\bar{C}$
0	0	1	1	$\bar{A}\,\bar{B}\,C$
0	1	0	2	$\bar{A}\,B\,\bar{C}$
0	1	1	3	$\bar{A}\,B\,C$
1	0	0	4	$A\,\bar{B}\,\bar{C}$
1	0	1	5	$A\,\bar{B}\,C$
1	1	0	6	$A\,B\,\bar{C}$
1	1	1	7	$A\,B\,C$

A \ BC		00 $\bar{B}\,\bar{C}$	01 $\bar{B}\,C$	11 $B\,C$	10 $B\,\bar{C}$
0	$\bar{A}$	$\bar{A}\,\bar{B}\,\bar{C}$ 0	$\bar{A}\,\bar{B}\,C$ 1	$\bar{A}\,B\,C$ 3	$\bar{A}\,B\,\bar{C}$ 2
1	A	$A\,\bar{B}\,\bar{C}$ 4	$A\,\bar{B}\,C$ 5	$A\,B\,C$ 7	$A\,B\,\bar{C}$ 6

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

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

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

$$\boxed{F(A,B,C) = \bar{C}}$$

Minimization Techniques

Don't care conditions: A Don't Care cell can be represented by a cross(X) or don't care(d) in K-Maps representing an invalid combination.

- In the Excess-3 code system, the states 0000, 0001, 0010, 1101, 1110, and 1111 are invalid or unspecified.

These states are called don't cares.

Q. Minimize the following function in SOP form using K-Map:

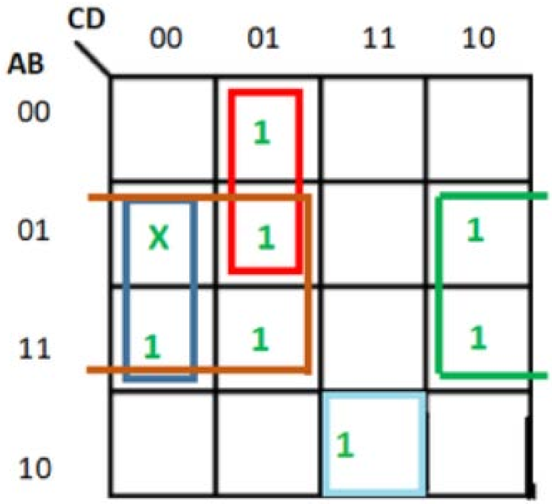
$$\boldsymbol{F = m(1, 5, 6, 11, 12, 13, 14) + d(4)}$$

Since values are upto m_{14}, this indicates 16 combinations.
Now, $2^4 = 16$. So, we have 4 variables.
Let them be A,B,C and D.

Boolean Function $F(A, B, C, D) = m(1,5,6,11,12,13,14) + d(4)$

So, ***one's*** exist at $(\mathbf{0001}, \mathbf{0101}, \mathbf{0110}, \mathbf{1011}, \mathbf{1100}, \mathbf{1101}, \mathbf{1110})$,
and ***d*** exists at $\mathbf{0100}$

Minimization Techniques



$$F(A,B,C,D) = \underbrace{B\bar{C} + B\bar{D}}_{Due\ to\ don't\ care=1} + \bar{A}\bar{C}D + A\bar{B}CD$$

$$F(A,B,C,D) = B\bar{C} + B\bar{D} + \bar{A}\bar{C}D + A\bar{B}CD$$

GATE (EC)-2016

Following is the K-map of a Boolean function of five variables P, Q, R, S, and X. The minimum sum-of-product (SOP) expression for the function is

X=0

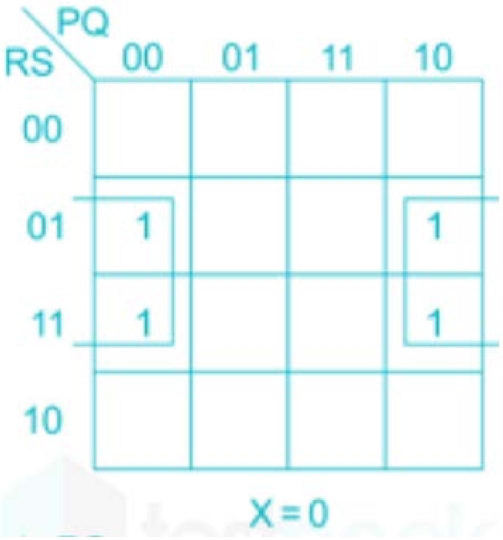
RS \ PQ	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	1	0	0	1
10	0	0	0	0

X=1

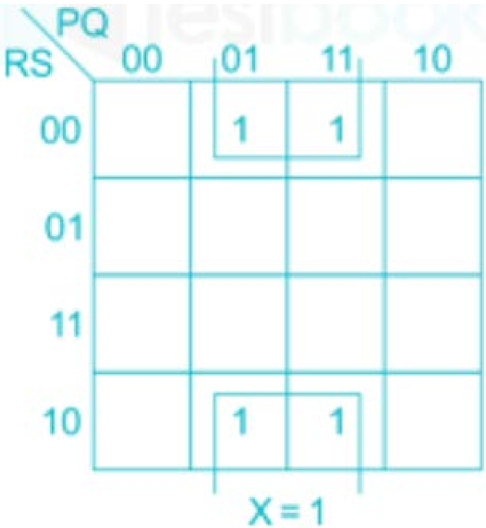
RS \ PQ	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	0	0	0
10	0	1	1	0

1. $\bar{P}\bar{Q}S\bar{X} + P\bar{Q}S\bar{X} + Q\bar{R}\bar{S}X + QR\bar{S}X$
2. $\bar{Q}S\bar{X} + Q\bar{S}X$
3. $\bar{Q}SX + Q\bar{S}X$
4. $\bar{Q}S + Q\bar{S}$

Minimization Techniques



$\bar{X}S\bar{Q}$



$X\bar{S}Q$

$$F = \bar{X}S\bar{Q} + X\bar{S}Q$$

Minimization Techniques

Quine-McCluskey Tabular Method

Step 1 – Arrange the given min terms in an **ascending order** and make the groups based on the number of ones present in their binary representations. So, there will be **at most ‘n+1’ groups** if there are ‘n’ Boolean variables in a Boolean function or ‘n’ bits in the binary equivalent of min terms.
Step 2 – Compare the min terms present in **successive groups**. If there is a change in only one-bit position, then take the pair of those two min terms. Place this symbol ‘_’ in the differed bit position and keep the remaining bits as it is.
Step 3 – Repeat step2 with newly formed terms till we get all **prime implicants**.
Step 4 – Formulate the **prime implicant table**. It consists of set of rows and columns. Prime implicants can be placed in row wise and min terms can be placed in column wise. Place ‘1’ in the cells corresponding to the min terms that are covered in each prime implicant.
Step 5 – Find the essential prime implicants by observing each column. If the min term is covered only by one prime implicant, then it is **essential prime implicant**. Those essential prime implicants will be part of the simplified Boolean function.
Step 6 – Reduce the prime implicant table by removing the row of each essential prime implicant and the columns corresponding to the min terms that are covered in that essential prime implicant. Repeat step 5 for Reduced prime implicant table. Stop this process when all min terms of given Boolean function are over.

Minimization Techniques

Example

Let us simplify the following Boolean function, $f(W,X,Y,Z)=\sum m(2,6,8,9,10,11,14,15)$, using Quine-McCluskey tabular method.

- The given Boolean function is in sum of min terms form.
- It is having 4 variables W, X, Y & Z.
- The given min terms are 2, 6, 8, 9, 10, 11, 14 and 15.
- The ascending order of these min terms based on the number of ones present in their binary equivalent is 2, 8, 6, 9, 10, 11, 14 and 15. The following table shows these min terms and their equivalent binary representations.

Group Name	Min terms	W	X	Y	Z
GA1	2	0	0	1	0
	8	1	0	0	0
GA2	6	0	1	1	0
	9	1	0	0	1
	10	1	0	1	0
GA3	11	1	0	1	1
	14	1	1	1	0
GA4	15	1	1	1	1

Minimization Techniques

The given min terms are arranged into 4 groups based on the number of ones present in their binary equivalents.

The min terms, which are differed in only one-bit position from adjacent groups are merged. That differed bit is represented with this symbol, '-'.
In this case, there are three groups and each group contains combinations of two min terms. The given table shows the possible merging of min term pairs from adjacent groups.

Group Name	Min terms	W	X	Y	Z
GB1	2,6	0	-	1	0
	2,10	-	0	1	0
	8,9	1	0	0	-
	8,10	1	0	-	0
GB2	6,14	-	1	1	0
	9,11	1	0	-	1
	10,11	1	0	1	-
	10,14	1	-	1	0
GB3	11,15	1	-	1	1
	14,15	1	1	1	-

The given table shows the possible merging of min terms from adjacent groups.

Minimization Techniques

The successive groups of min term pairs, which are differed in only one-bit position are merged.
That differed bit is represented with this symbol, ‘-‘.
In this case, there are two groups and each group contains combinations of four min terms.
Here, these combinations of 4 min terms are available in two rows.
So, we can remove the repeated rows.

Group Name	Min terms	W	X	Y	Z
GB1	2,6,10,14	-	-	1	0
	2,10,6,14	-	-	1	0
	8,9,10,11	1	0	-	-
	8,10,9,11	1	0	-	-
GB2	10,11,14, 15	1	-	1	-
	10,14,11, 15	1	-	1	-

Minimization Techniques

Further merging of the combinations of min terms from adjacent groups is not possible, since they are differed in more than one-bit position.

There are three rows in the above table.

So, each row will give one prime implicant.

Therefore, the **prime implicants** are YZ', WX' & WY.

The **prime implicant table** is shown alongside.

Group Name	Min terms	W	X	Y	Z
GC1	2,6,10,14	-	-	1	0
	8,9,10,11	1	0	-	-
GC2	10,11,14, 15	1	-	1	-

Min terms / Prime Implicants	2	6	8	9	10	11	14	15
YZ'	1	1			1		1	
WX'			1	1	1	1		
WY					1	1	1	1

Minimization Techniques

The prime implicants are placed in row wise and min terms are placed in column wise.
1s are placed in the common cells of prime implicant rows and the corresponding min term columns.
The min terms 2 and 6 are covered only by one prime implicant **YZ'**.
So, it is an **essential prime implicant**.
This will be part of simplified Boolean function. Now, remove this prime implicant row and the corresponding min term columns.

Min terms / Prime Implicants	2	6	8	9	10	11	14	15
YZ'	1	1			1		1	
WX'			1	1	1	1		
WY					1	1	1	1

Min terms / Prime Implicants	8	9	11	15
WX'	1	1	1	
WY			1	1

Minimization Techniques

The min terms 8 and 9 are covered only by one prime implicant **WX’**.
So, it is an **essential prime implicant**.
This will also be part of simplified Boolean function.

Min terms / Prime Implicants	8	9	11	15
WX’	1	1	1	
WY			1	1

Now, remove this prime implicant row and the corresponding min term columns.
The reduced prime implicant table is shown along.

Min terms / Prime Implicants	15
WY	1

The min term 15 is covered only by one prime implicant WY. So, it is an essential prime implicant. This will also be part of simplified Boolean function.

we got three prime implicants and all three are essential. Therefore, the simplified Boolean function is: f(W,X,Y,Z) = YZ’ + WX’ + WY.

Minimization Techniques

Variable Entered Maps

K-map is the best manual technique to solve Boolean equations, but it becomes difficult to manage when the number of variables exceeds 5 or 6.

So, a technique called **Variable Entrant Map (VEM)** is used to increase the effective size of the K-map.

It allows a smaller map to handle a large number of variables.

This is done by writing output in terms of input.

Example – A 3-variable function can be defined as a function of 2-variables if the output is written in terms of third variable.

Consider a function **F(A,B,C) = m(0,1,2,5)**

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

A	B	F
0	0	1
0	1	C'
1	0	C
1	1	0

B \ A	0	1
0	1	C
1	C'	0