Min terms

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 min term is denoted as mi where i is in the range of $0 \leq i < 2^n$.

A variable is in complemented form, if its value is assigned to 0, and the variable is un-complimented form, if its value is assigned to 1.

For a 2-variable (x and y) Boolean function, the possible minterms are:

x’y’, x’y, xy’ and xy.

For a 3-variable (x, y and z) Boolean function, the possible minterms are:

x’y’z’, x’y’z, x’yz’, x’yz, xy’z’, xy’z, xyz’ and xyz.

Inputs		Output	Minterm
X	Y	F	M
0	0	0	X'Y'
0	1	1	X'Y
1	0	1	XY'
1	1	1	XY

Max terms

A max term is defined as the product of n variables, within the range of $0 \leq i < 2^n$. The max term is denoted as Mi. In max term, each variable is complimented, if its value is assigned to 1, and each variable is un-complimented if its value is assigned to 0.

For a 2-variable (x and y) Boolean function, the possible max terms are:

x + y, x + y’, x’ + y and x’ + y’.

For a 3-variable (x, y and z) Boolean function, the possible maxterms are:

x + y + z, x + y + z’, x + y’ + z, x + y’ + z’, x’ + y + z, x’ + y + z’, x’ + y’ + z and x’ + y’ + z’.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

MAX TERMS

A+B

A+B’

A’+B

A’+B’

			Minterms		Maxterms	
X	Y	Z	Term	Designation	Term	Designation
0	0	0	$\bar{X}\bar{Y}\bar{Z}$	mo	$X+Y+Z$	Mo
0	0	1	$\bar{X}\bar{Y}Z$	m1	$X+Y+\bar{Z}$	M1
0	1	0	$\bar{X}Y\bar{Z}$	m2	$X+\bar{Y}+Z$	M2
0	1	1	$\bar{X}YZ$	m3	$X+\bar{Y}+\bar{Z}$	M3
1	0	0	$X\bar{Y}\bar{Z}$	m4	$\bar{X}+Y+Z$	M4
1	0	1	$X\bar{Y}Z$	m5	$\bar{X}+Y+\bar{Z}$	M5
1	1	0	$XY\bar{Z}$	m6	$\bar{X}+\bar{Y}+Z$	M6
1	1	1	XYZ	m7	$\bar{X}+\bar{Y}+\bar{Z}$	M7

Variables			Min terms	Max terms
A	B	C	m_i	M_i
0	0	0	A' B' C' = m 0	A + B + C = M 0
0	0	1	A' B' C = m 1	A + B + C' = M 1
0	1	0	A' B C' = m 2	A + B' + C = M 2
0	1	1	A' B C = m 3	A + B' + C' = M 3
1	0	0	A B' C' = m 4	A' + B + C = M 4
1	0	1	A B' C = m 5	A' + B + C' = M 5
1	1	0	A B C' = m 6	A' + B' + C = M 6
1	1	1	A B C = m 7	A' + B' + C' = M 7

SOP AND POS FORM

- 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

If the input variable (let A) value is :

- Zero (0) – a is LOW -It should be represented as A’ (Complement of A)
- One (1) – a is HIGH -It should be represented as A

- Considering number of input variables =3, Say A, B and C. Total number of combinations are: 2^3=8.

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

Sum of Products (SOP): SUM OF MIN-TERMS

- It is one of the ways of writing a boolean expression.
- As the name suggests, it is formed by adding (OR operation) the product terms. These product terms are also called as 'min-terms'.
- Min-terms are represented with 'm', they are the product(AND operation) of boolean variables either in normal form or complemented form.

- Therefore, SOP is sum of minterms and is represented as:
- F in SOP = \Sigma m(0, 3)
- Here, F is sum of minterm0 and minterm3.

- For Example:
- A=0, B=0, C=0   Minterm is A'.B'.C'
- A=1, B=0, C=1   Minterm is A.B'.C

SOP form can be obtained by

Writing an AND term for each input combination, which produces HIGH output.
Writing the input variables if the value is 1, and write the complement of the variable if its value is 0.
OR the AND terms to obtain the output function.
Ex: Boolean expression for majority function F = A’BC + AB’C + ABC ‘ + ABC

A	B	C	F	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	A’BC
1	0	0	0	
1	0	1	1	AB’C
1	1	0	1	ABC’
1	1	1	1	ABC

Consider a function X, whose truth table is as follows

(**INPUTS**) A B C	(**OUTPUT**) X	DECIMAL
0 0 0	0	0
0 0 1	**1**	1
0 1 0	0	2
0 1 1	**1**	3
1 0 0	0	4
1 0 1	0	5
1 1 0	**1**	6
1 1 1	0	7

The function X can be written in SOP form by adding all the min-terms when X is HIGH(1).

While writing SOP, the following convention is to be followed:

If variable A is Low(0) - A'

A is High(1) - A

X (SOP) = Σ m(1, 3, 6)

= A'.B'.C + A'.B.C + A.B.C'

SOP Implementation from a Truth Table

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

$$F = A\overline{B}C + \overline{A}BC + \overline{A}\,\overline{B}C$$

Product of Sums (POS) Form/ PRODUCT OF MAX TERMS

POS form can be obtained by

Writing an OR term for each input combination, which produces LOW output.
Writing the input variables if the value is 0, and write the complement of the variable if its value is 1.
AND the OR terms to obtain the output function.
Ex: Boolean expression for majority function F = (A + B + C) (A + B + C ‘) (A + B’ + C) (A’ + B + C)

Boolean expression for majority function F = (A + B + C) (A + B + C ‘) (A + B’ + C) (A’ + B + C)

A	B	C	F
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

	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.

Write a Boolean SOP expression for this truth table, then simplify that expression as much as possible

A	B	C	Output	
0	0	0	0	
0	0	1	0	
0	1	0	1	A’BC’
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	ABC’
1	1	1	0	

Original SOP expression:

A’BC’+ABC’ = BC’(A’+A)= BC’

(SINCE A+A’=1)

Conversion of SOP form to standard SOP form or Canonical SOP form

For getting the standard SOP form of the given non-standard SOP form, we will add all the variables in each product term which do not have all the variables. By using the Boolean algebraic law, ($x + x' = 0$) and by following the below steps we can easily convert the normal SOP function into standard SOP form.

- Multiply each non-standard product term by the sum of its missing variable and its complement.
- Repeat step 1, until all resulting product terms contain all variables
- For each missing variable in the function, the number of product terms doubles.

Convert the non standard SOP function F = AB + A C + B C INTO CANONICAL

Sol:

$$F = A\,B + A\,C + B\,C$$
$$= A\,B\,(C + C') + A\,(B + B')\,C + (A + A')\,B\,C$$
$$= A\,B\,C + A\,B\,C' + A\,B\,C + A\,B'\,C + A\,B\,C + A'\,B\,C$$
$$= A\,B\,C + A\,B\,C' + A\,B'\,C + A'\,B\,C$$

Conversion of POS form to standard POS form or Canonical POS form

For getting the standard POS form of the given non-standard POS form, we will add all the variables in each product term that do not have all the variables. By using the Boolean algebraic law (x * x' = 0) and by following the below steps, we can easily convert the normal POS function into a standard POS form.

- **By adding each non-standard sum term to the product of its missing variable and its complement, which results in 2 sum terms**
- **Applying Boolean algebraic law, x + y z = (x + y) * (x + z)**
- **By repeating step 1, until all resulting sum terms contain all variables**
-

By these three steps, we can convert the POS function into a standard POS function.

F = (p' + q + r) * (q' + r + s') * (p + q' + r' + s)

1. Term (p' + q + r)

As we can see that the variable s or s' is missing in this term. So we add s*s' = 1 in this term.

(p' + q + r + s*s') = (p' + q + r + s) * (p' + q + r + s')

2. Term (q' + r + s')
Similarly, we add p*p' = 1 in this term for getting the term containing all the variables.

(q' + r + s' + p*p') = (p + q' + r + s') * (p' + q' + r + s')

3. Term (q' + r + s')

Now, there is no need to add anything because all the variables are contained in this term.

So, the standard POS form equation of the function is

F = (p' + q + r + s)* (p' + q + r + s')* (p + q' + r + s')* (p' + q' + r + s') * (p + q' + r' + s)

CONVERT THE EXP INTO CANONICAL FORM

$F(P,Q,R)= PQ+PR'+QR'$

$F(..) = PQ(R+R') + PR'(Q+Q') + QR'(P+P')$

$= PQR+PQR'+PQR'+PQ'R'+PQR'+P'QR'$

$= 111 , 110 , 100, 110,010$

$= M7, M6, M4, M2$

SIMPLIFICATION OF BOOLEAN TERM _(A+B’+C) (A+B+C)(A+B’+C’)

= (A+B’+C) (A+B+C)(A+B’+C’)

= LETS SUPPOSE A+C = X

= X+B’) (X+B)(A+B’+C’)

= DISTRIBUTIVE LAW : X+BB’ = (X+B)(X+B’)

= X+BB’ (A+B’+C’)

= X (A+B’+C’)

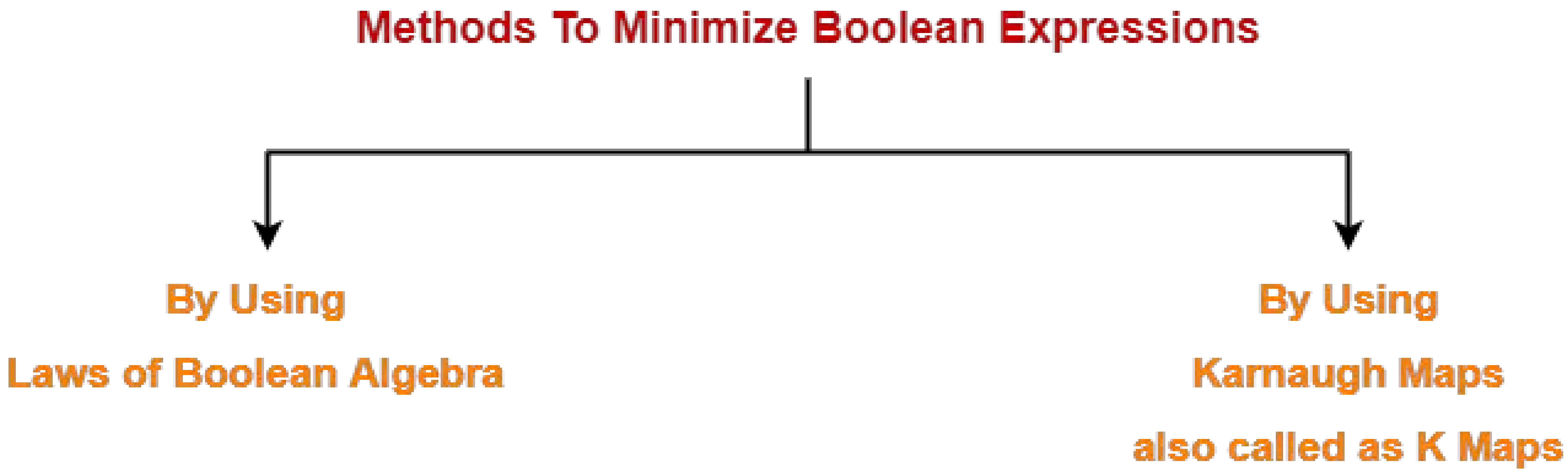
= (A+C)(A+B’+C’)

= AGAIN DISTRIBUTIVE LAW A+BC= (A+B) (A+C)

= A+ C.(B’+C’)

= A+B’C+CC’

= A+B’C


Methods To Minimize Boolean Expressions
By Using
Laws of Boolean Algebra
By Using
Karnaugh Maps
also called as K Maps

Karnaugh Map-

The Karnaugh Map also called as K Map is a graphical representation that provides a systematic method for simplifying the boolean expressions.

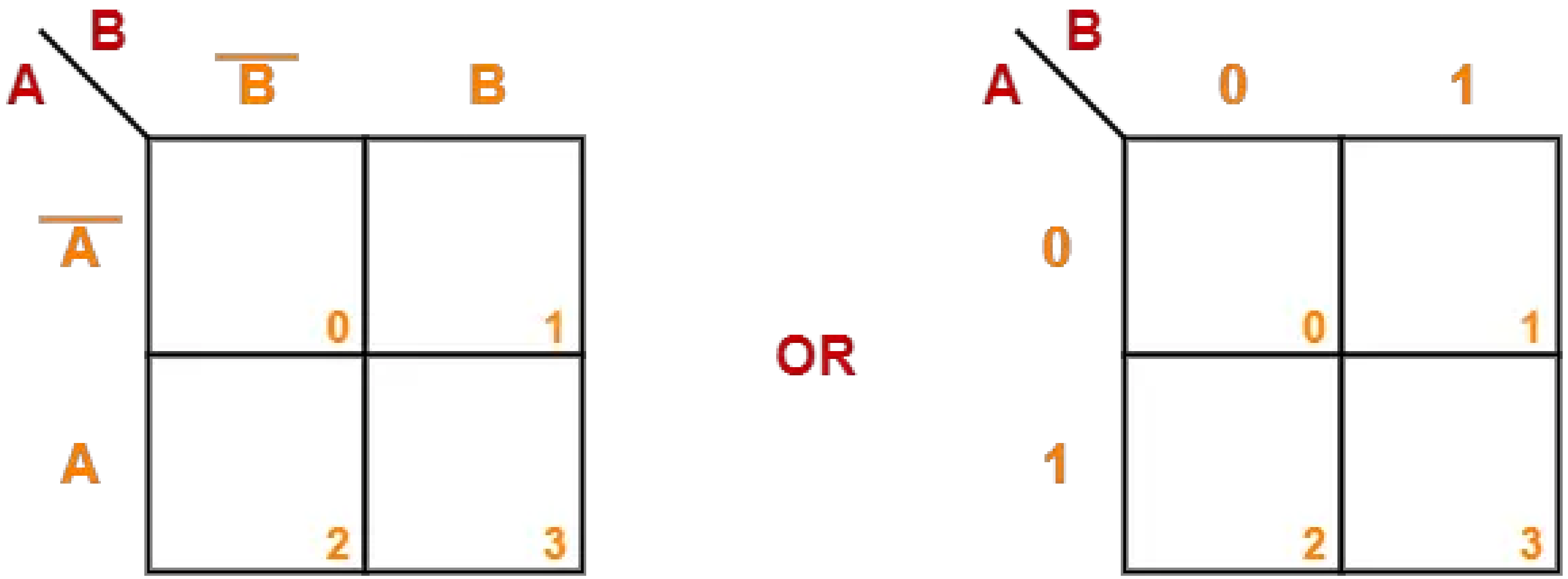
For a boolean expression consisting of n-variables, number of cells required in K Map = 2n cells.

Two Variable K Map-

Two variable K Map is drawn for a boolean expression consisting of two variables.

The number of cells present in two variable K Map = 22 = 4 cells.

So, for a boolean function consisting of two variables, we draw a 2 x 2 K Map.



Two Variable K Map

Three Variable K Map-

Three variable K Map is drawn for a boolean expression consisting of three variables.
The number of cells present in three variable K Map = 2^3 = 8 cells. 2^3
So, for a boolean function consisting of three variables, we draw a 2 x 4 K Map.

Three variable K Map may be represented as-

A	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	0	1	3	2
A	4	5	7	6

OR

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

Four Variable K Map-

Four variable K Map is drawn for a boolean expression consisting of four variables.
The number of cells present in four variable K Map = 24 = 16 cells.
So, for a boolean function consisting of four variables, we draw a 4 x 4 K Map.

Four variable K Map may be represented as-

AB \ CD	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	1	3	2
$\overline{A}B$	4	5	7	6
AB	12	13	15	14
$A\overline{B}$	8	9	11	10

Karnaugh Map Simplification Rules-

To minimize the given boolean function,

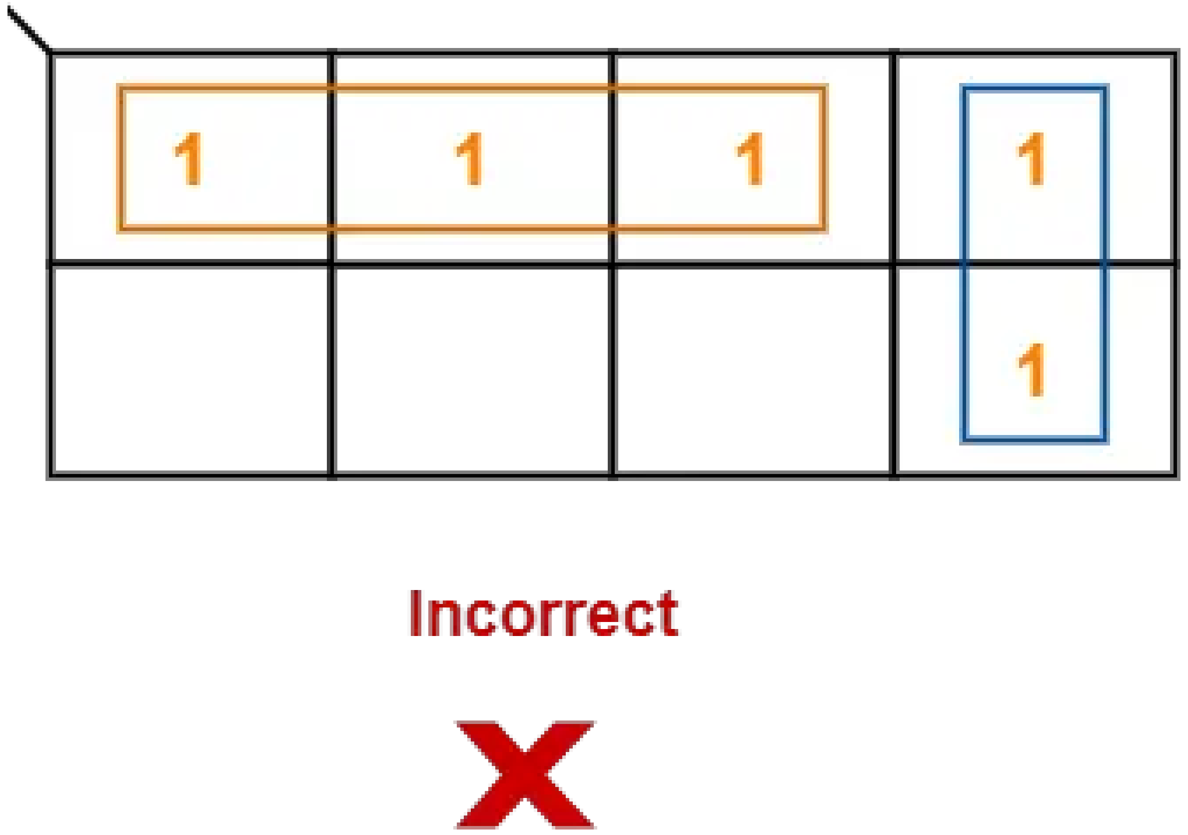
- We draw a K Map according to the number of variables it contains.
- We fill the K Map with 0’s and 1’s according to its function.
- Then, we minimize the function in accordance with the following rules.

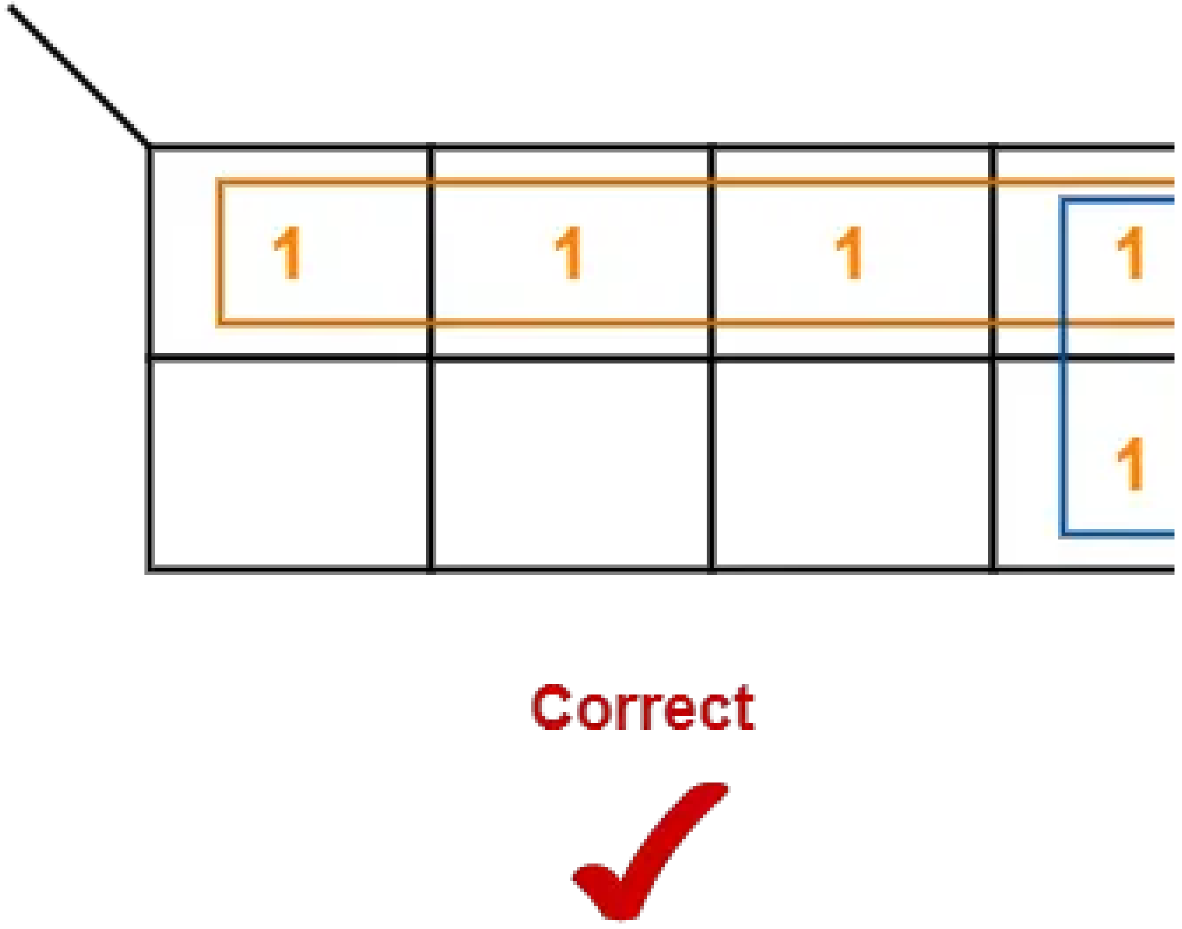
Rule-01: We can either group 0's with 0's or 1's with 1's but we can not group 0's and 1's together.

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 2n i.e. 1, 2, 4, 8, 16 and so on number of cells.

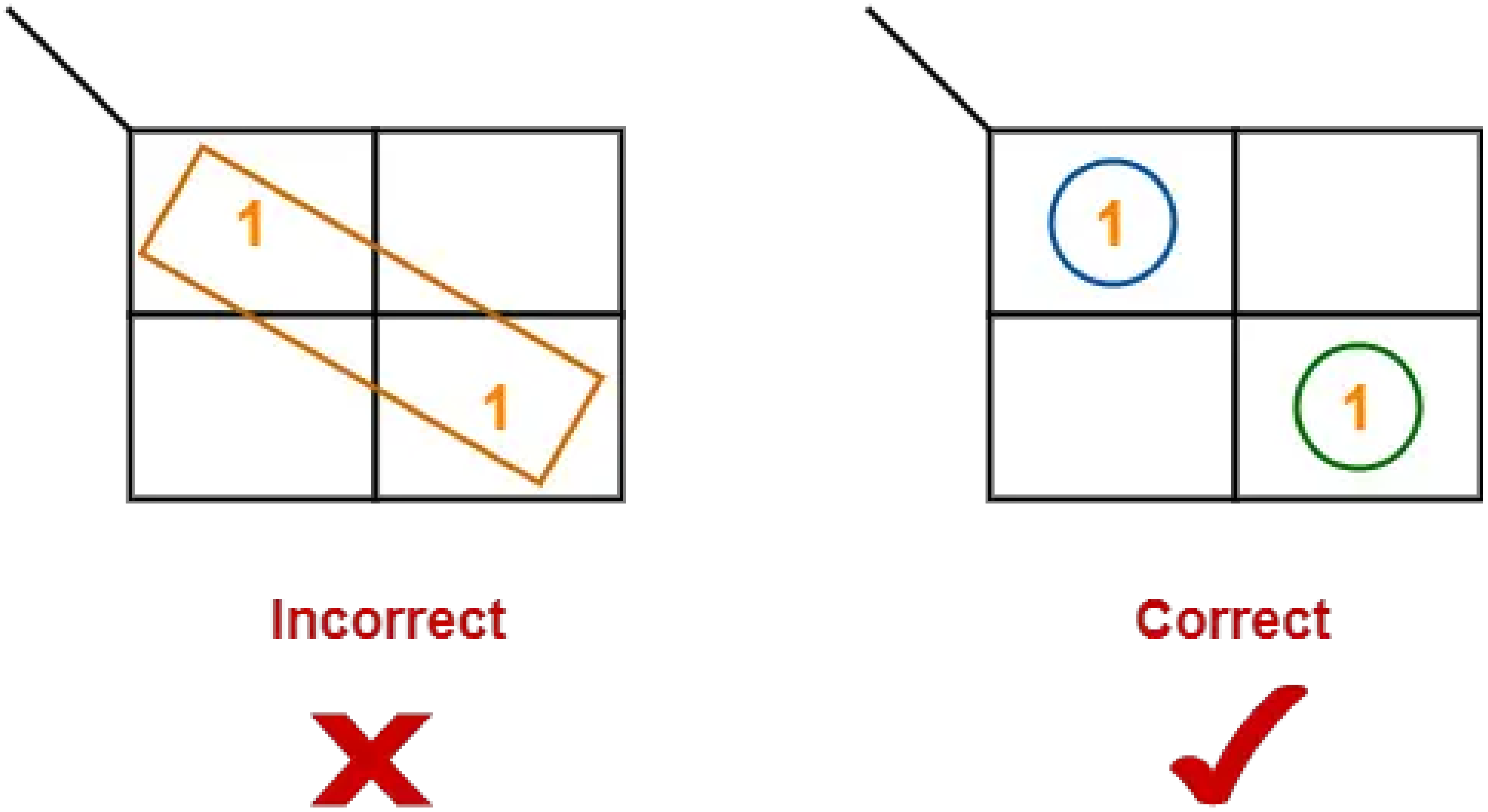




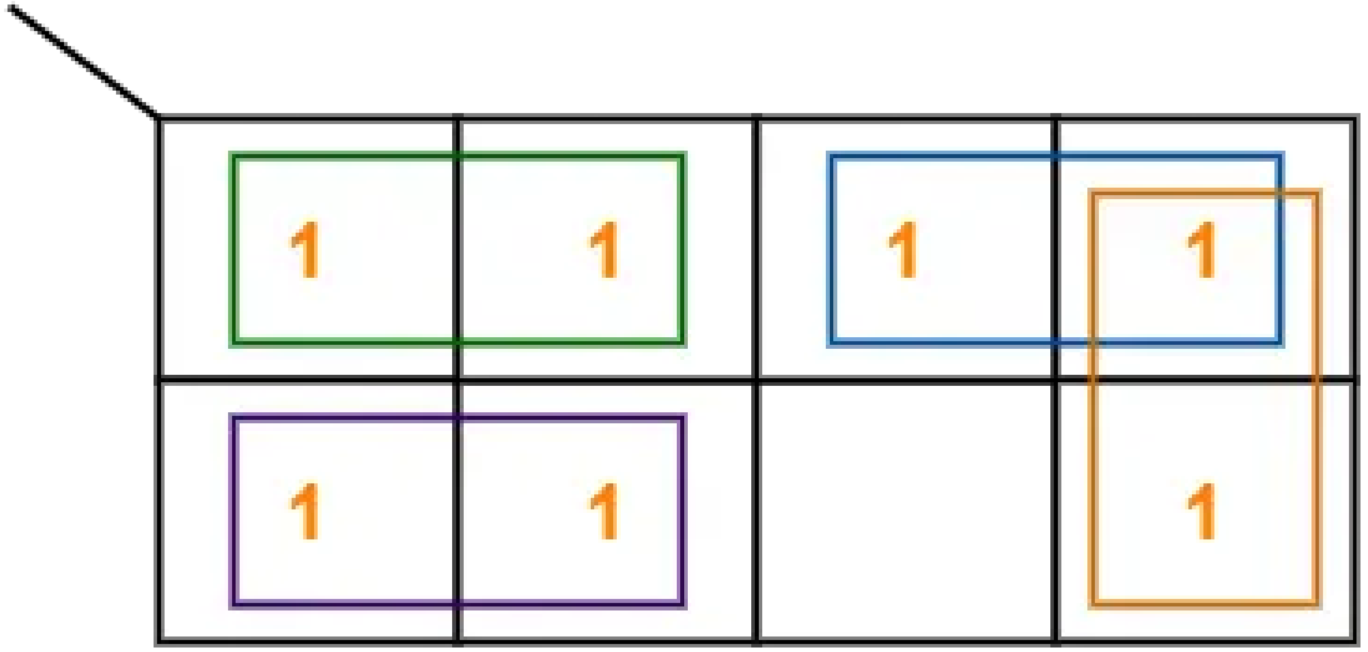
Rule-04 :

Groups can be only either horizontal or vertical.

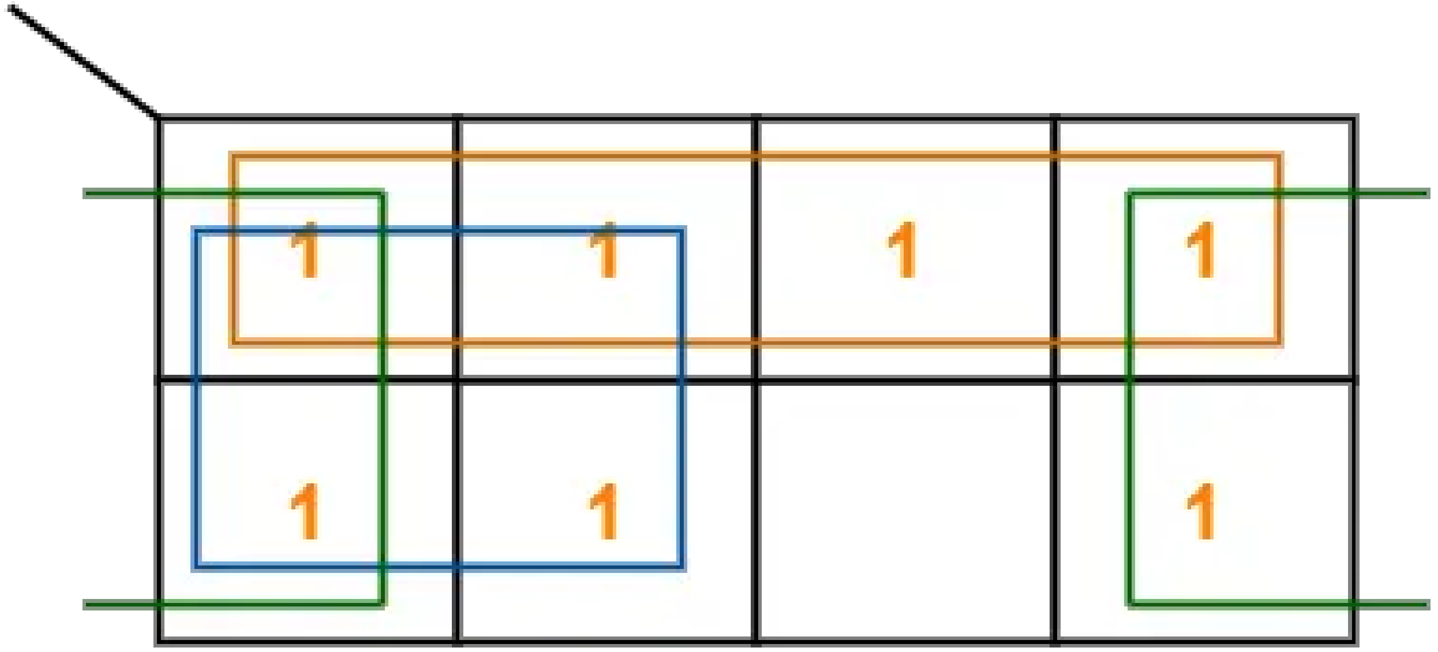
We can not create groups of diagonal or any other shape.



Rule-05: Each group should be as large as possible.

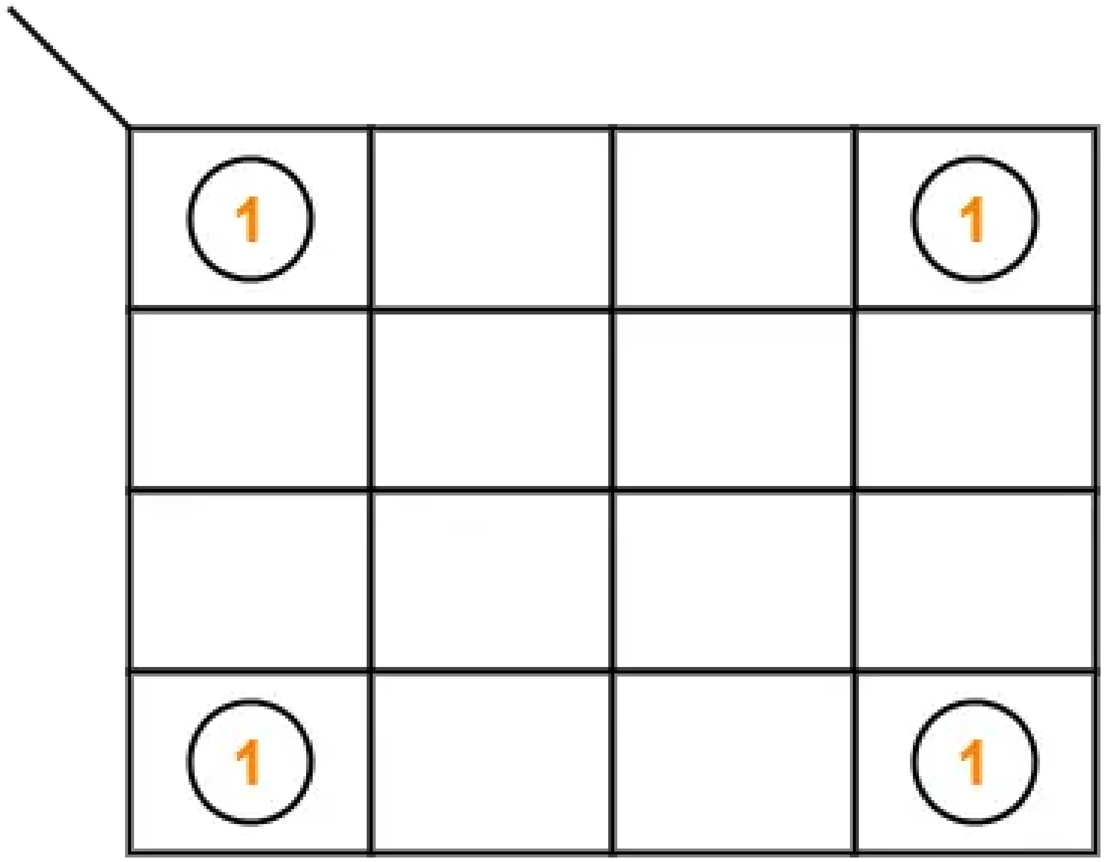


Incorrect

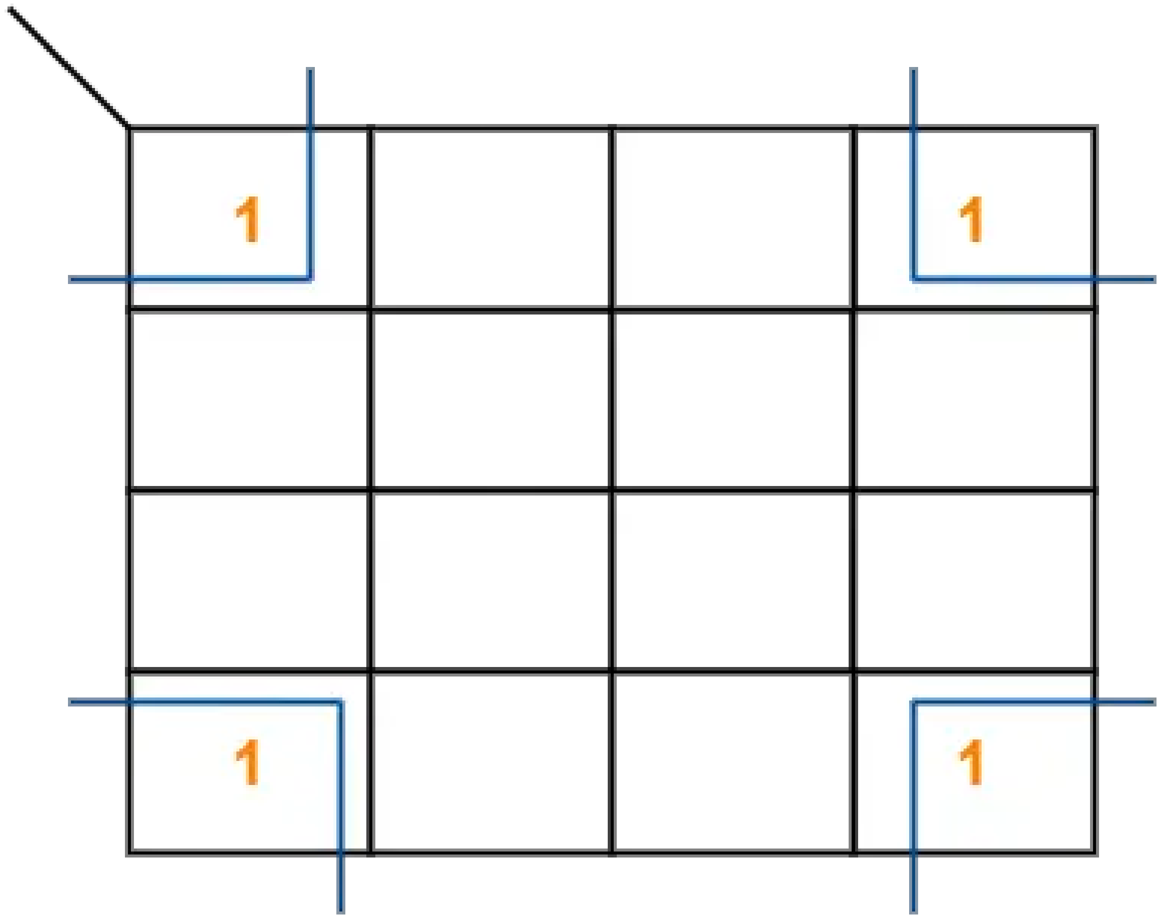


Correct

Rule-06: Opposite grouping and corner grouping are allowed.
The example of opposite grouping is shown illustrated in Rule-05.
The example of corner grouping is shown below.



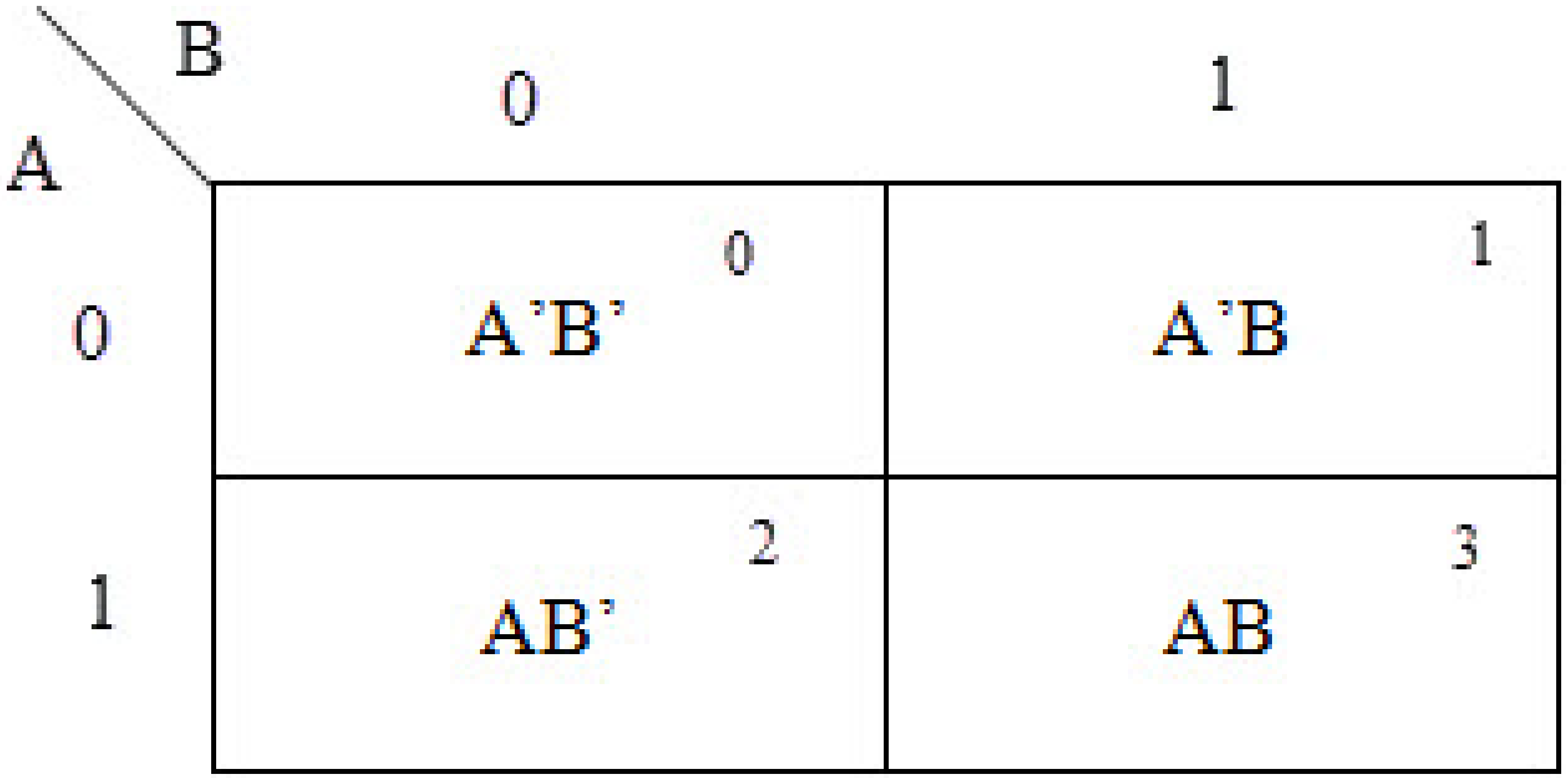
Incorrect



Correct

Rule-07: There should be as few groups as possible.

A	B	Possible Outputs	Location on K-map
0	0	A'B'	0
0	1	A'B	1
1	0	AB'	2
1	1	AB	3

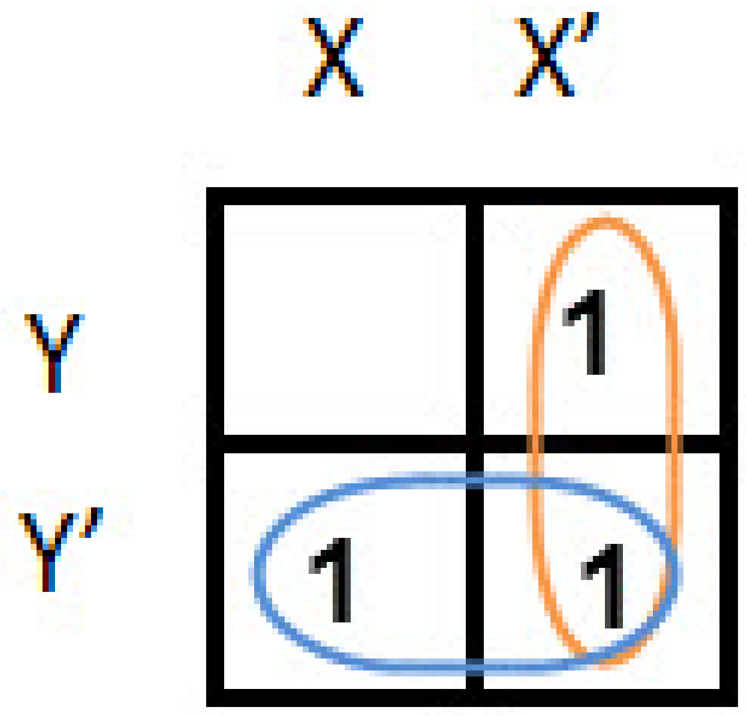


Simplify the given 2-variable Boolean equation by using K-map.

$$F = X\,Y' + X'\,Y + X'Y'$$

First, let's construct the truth table for the given equation,

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	1



By reducing each group, we obtain a conjunction of the minimized expression such as by taking out the common terms from two groups, i.e. X' and Y'. So the reduced equation will be X' +Y

3 variable K-maps

For a 3-variable Boolean function, there is a possibility of 8 output min terms. The general representation of all the min terms using 3-variables is shown below.

A	B	C	Output Function	Location on K-map
0	0	0	A'B'C'	0
0	0	1	A'B'C	1
0	1	0	A'BC'	2
0	1	1	A'BC	3
1	0	0	AB'C'	4
1	0	1	AB'C	5
1	1	0	ABC'	6
1	1	1	ABC	7

A \ BC	00	01	11	10
0	A'B'C' 0	A'B'C 1	A'BC 3	A'BC' 2
1	AB'C' 4	AB'C 5	ABC 7	ABC' 6

Simplify the given 3-variable Boolean equation by using k-map.

$$F = X'YZ + X'Y'Z + XYZ' + X'Y'Z' + XYZ + XY'Z'$$

First, let's construct the truth table for the given equation,

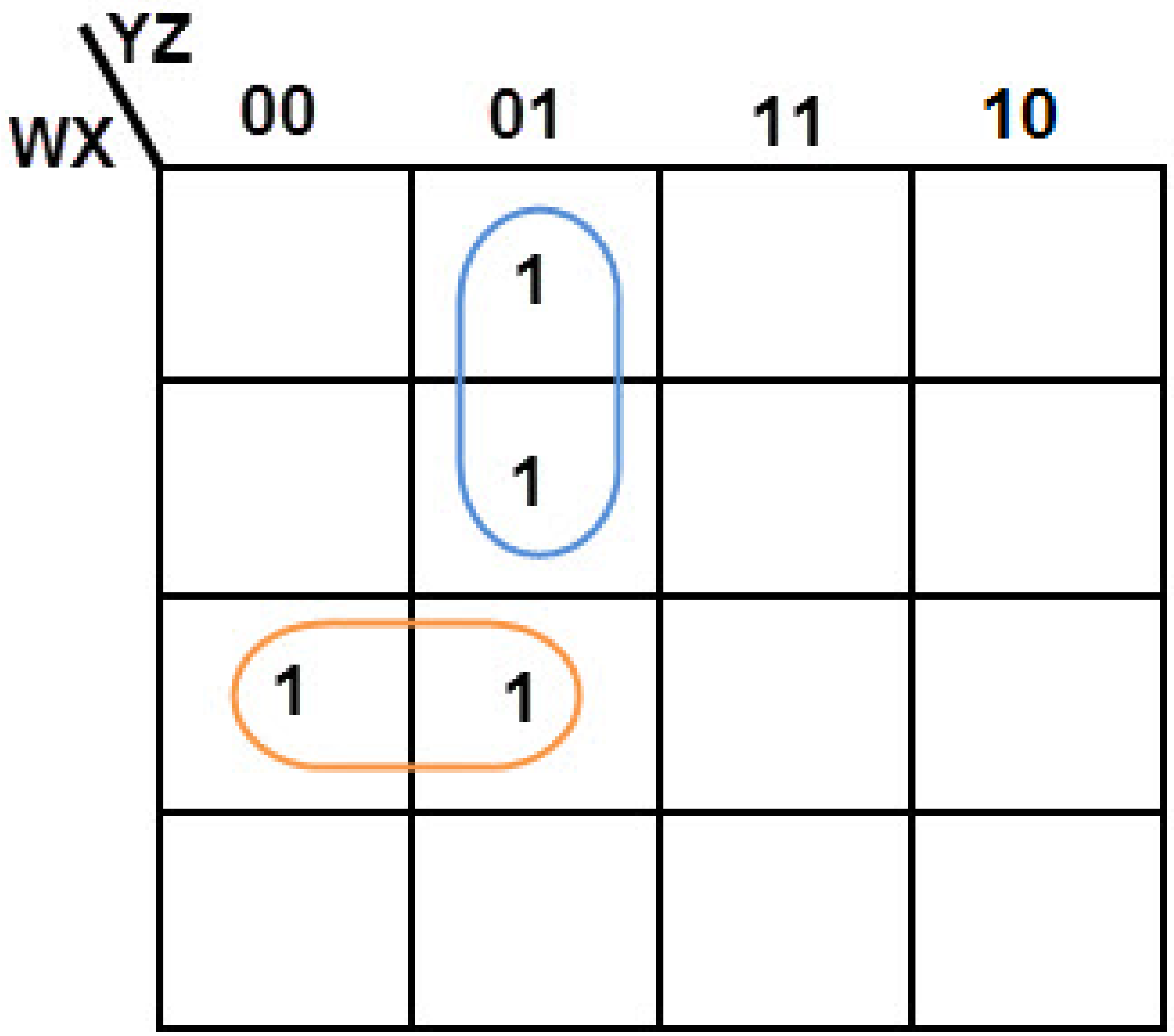
x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

A	B	C	D	Output function	K-map location
0	0	0	0	A’ B’ C’ D’	0
0	0	0	1	A’ B’ C’ D	1
0	0	1	0	A’ B’ C D’	2
0	0	1	1	A’ B’ C D	3
0	1	0	0	A’ B C’ D’	4
0	1	0	1	A’ B C’ D	5
0	1	1	0	A’ B C D’	6
0	1	1	1	A’ B C D	7
1	0	0	0	A B’ C’ D’	8
1	0	0	1	A B’ C’ D	9
1	0	1	0	A B’ C D’	10
1	0	1	1	A B’ C D	11
1	1	0	0	A B C’ D’	12
1	1	0	1	A B C’ D	13
1	1	1	0	A B C D’	14
1	1	1	1	A B C D	15

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

Simplify the given 4-variable Boolean equation by using k-map. F (W, X, Y, Z) = (1, 5, 12, 13)

Sol: F (W, X, Y, Z) = (1, 5, 12, 13)



By preparing k-map, we can minimize the given Boolean equation as

F = W Y’ Z + W ‘Y’ Z

Problem-01:

Minimize the following boolean function-

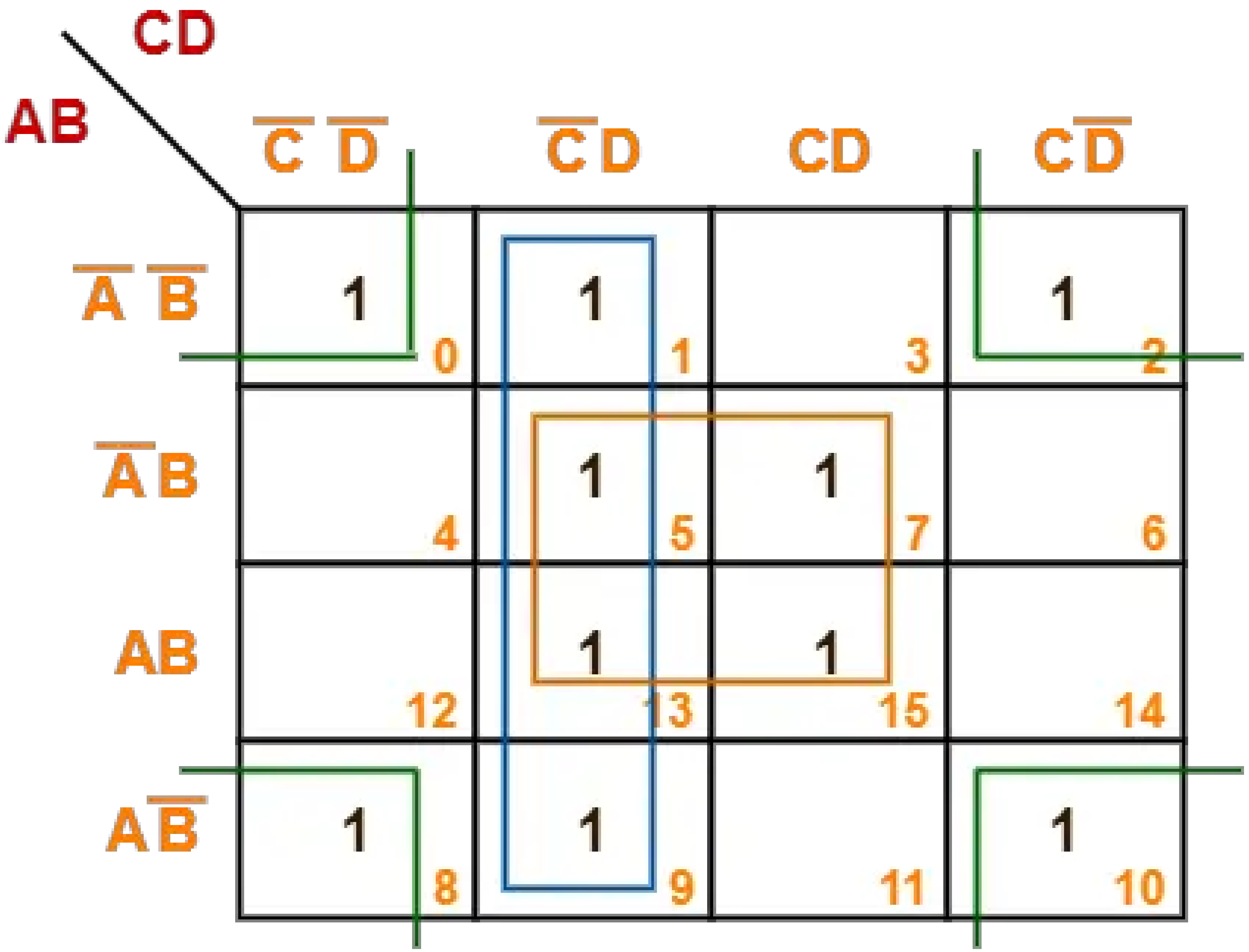
$$F(A, B, C, D) = \Sigma m(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$$

Solution-

Since the given boolean expression has 4 variables, so we draw a 4 x 4 K Map.

We fill the cells of K Map in accordance with the given boolean function.

Then, we form the groups in accordance with the above rules.



Now,

F(A, B, C, D) = (A'B + AB)(C'D + CD) + (A'B' + A'B + AB + AB')C'D + (A'B' + AB')(C'D' + CD') = BD + C'D + B'D'

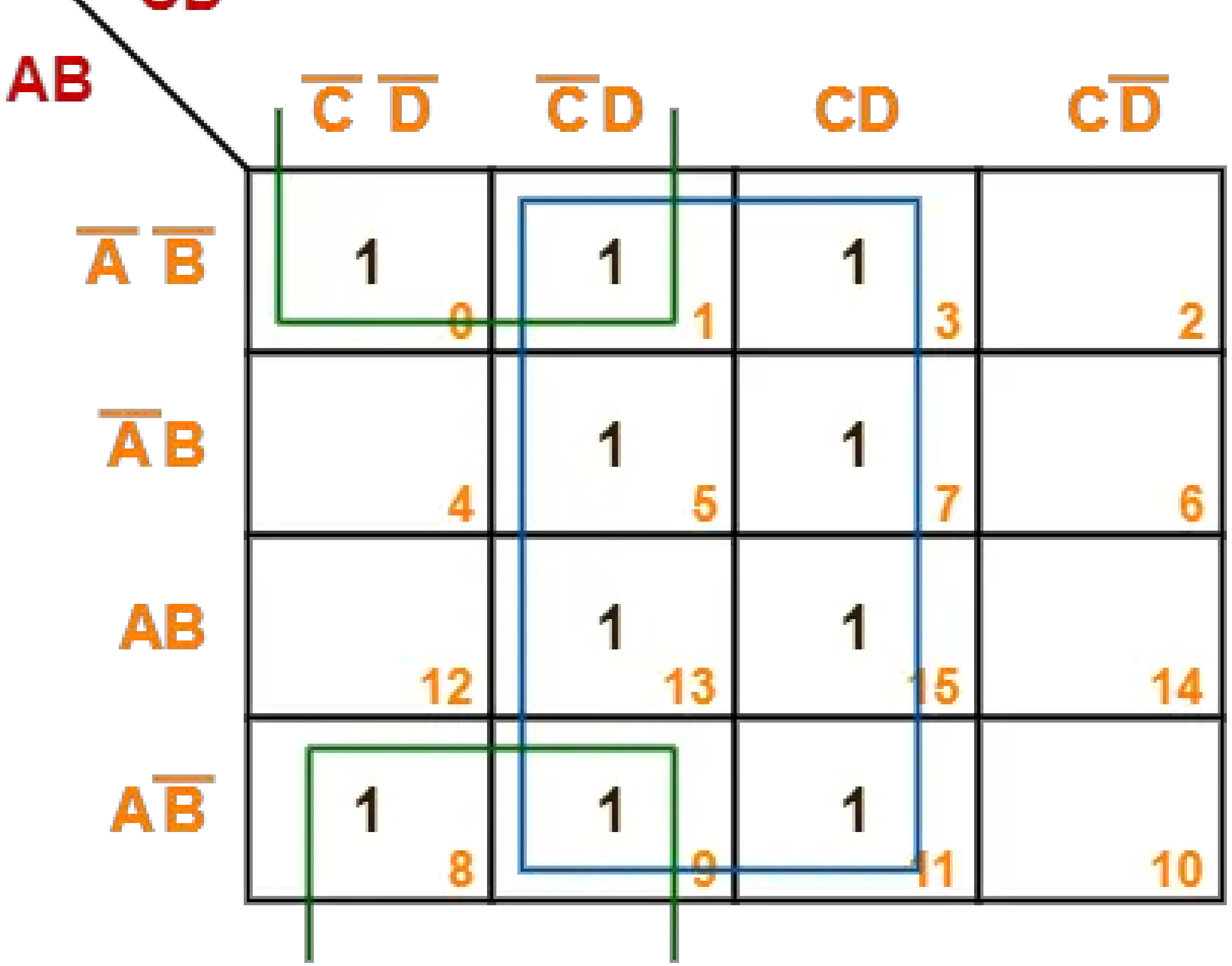
F(A, B, C, D) = BD + C'D + B'D'

Minimize the following boolean function-

$$F(A, B, C, D) = \Sigma m(0, 1, 3, 5, 7, 8, 9, 11, 13, 15)$$

Solution-

Since the given boolean expression has 4 variables, so we draw a 4 x 4 K Map.
We fill the cells of K Map in accordance with the given boolean function.
Then, we form the groups in accordance with the above rules.



F(A, B, C, D) = (A’B’ + A’B + AB + AB’)(C’D + CD) + (A’B’ + AB’)(C’D’ + C’D)

= D + B’C’

Thus, minimized boolean expression is-

F(A, B, C, D) = B’C’ + D

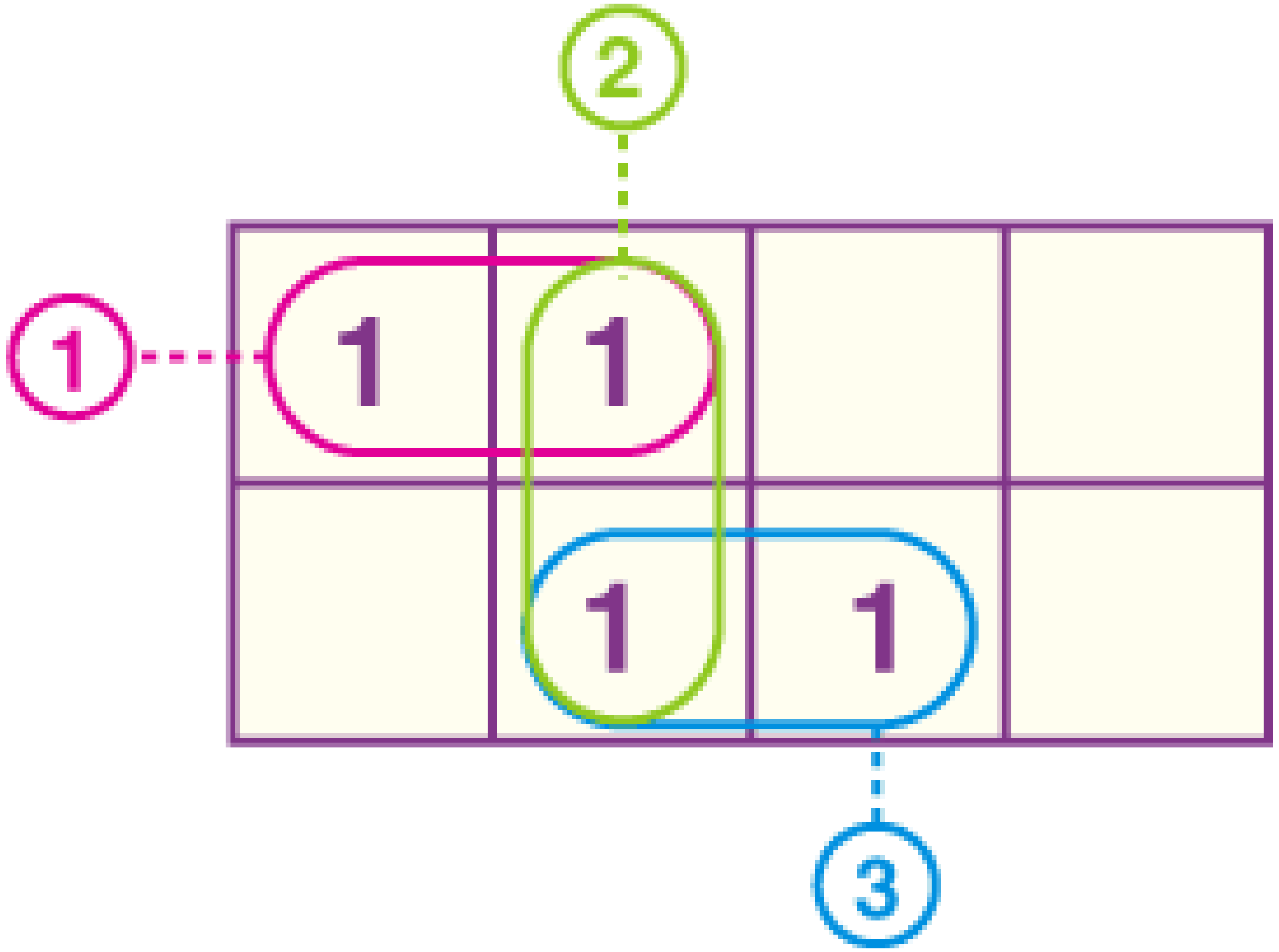
Various Implicants in K-Map

An implicant refers to the product/minterm term in the SOP (Sum of Products) or the sum/maxterm term in the POS (Product of Sums) of a Boolean function. For example, let us consider any boolean function, F = MN + MNO + NO, then implicants are MN, MNO and NO.

Prime Implicants

A group of squares or rectangles made up of a bunch of adjacent minterms that is allowed by definition of a Karnaugh Map are known as prime implicants or PI, i.e. all the possible groups that are formed in K-Map.

Example

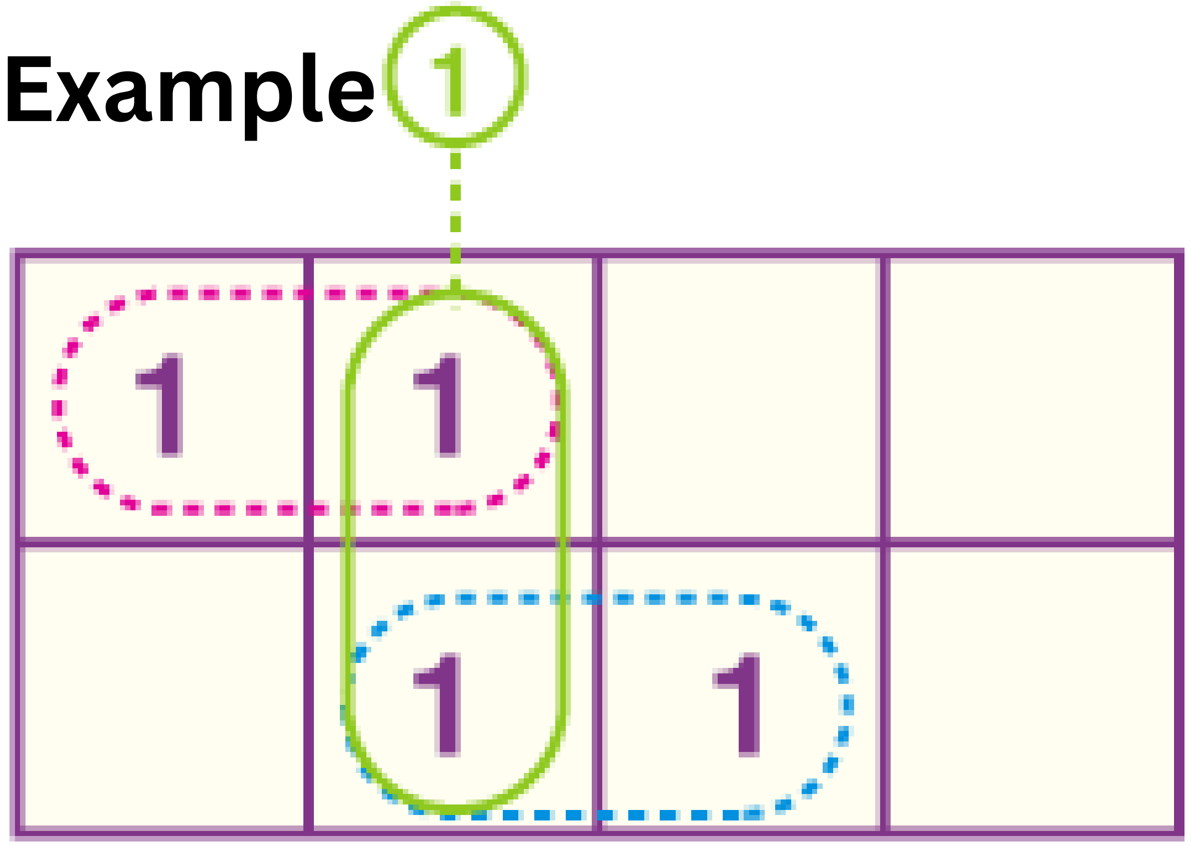


No. of Prime Implicants = 3

Redundant Prime Implicants

The redundant prime implicants or RPI refer to the prime implicants for which every one of its minterms gets covered by some important prime implicants. This type of prime implicant never happens to appear in the final solution.

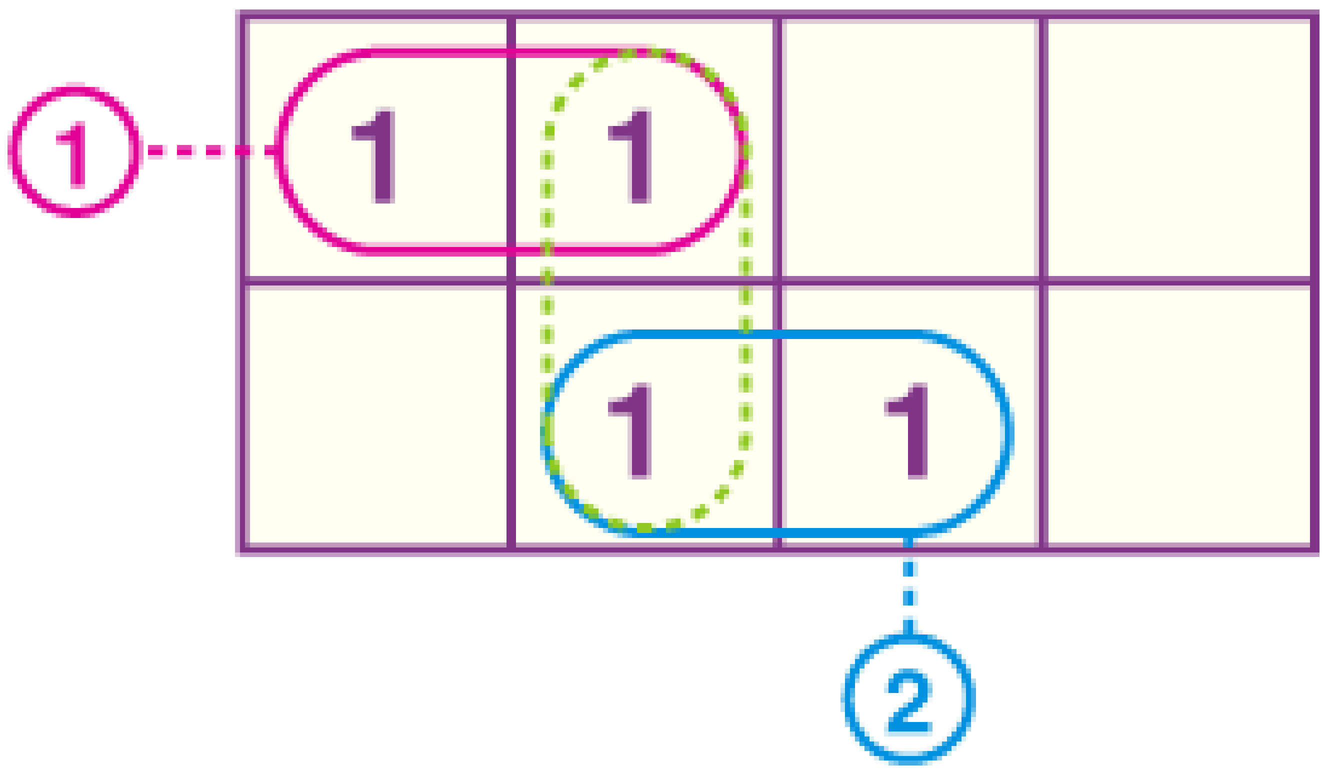
Example



No. of Redundant Prime Implicants = 1

Essential Prime Implicants

These refer to those subcubes or groups that cover at least one of the minterms that can't get covered by another prime implicant. The EPI or essential prime implicants refer to the prime implicants that always appear in the final solution.

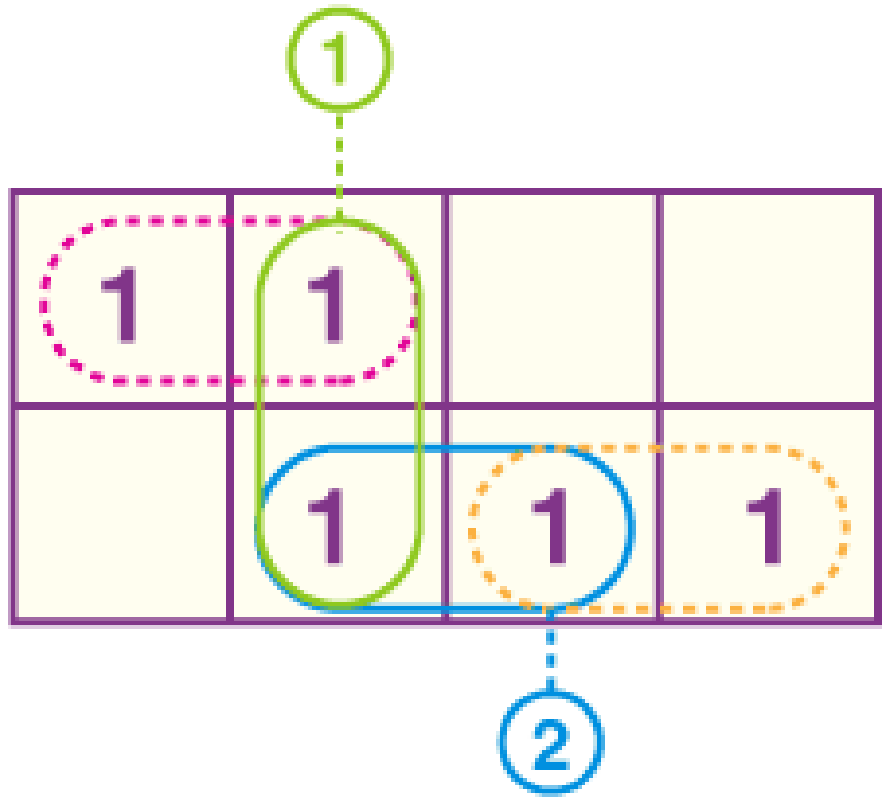


No. of Essential Prime Implicants = 2

Selective Prime Implicants

The SPI or selective prime implicants refer to those prime implicants for which neither the redundant nor essential prime implicants are there. They are also called non-essential prime implicants. These may appear in certain types of solutions or may not even appear in some solutions at all.

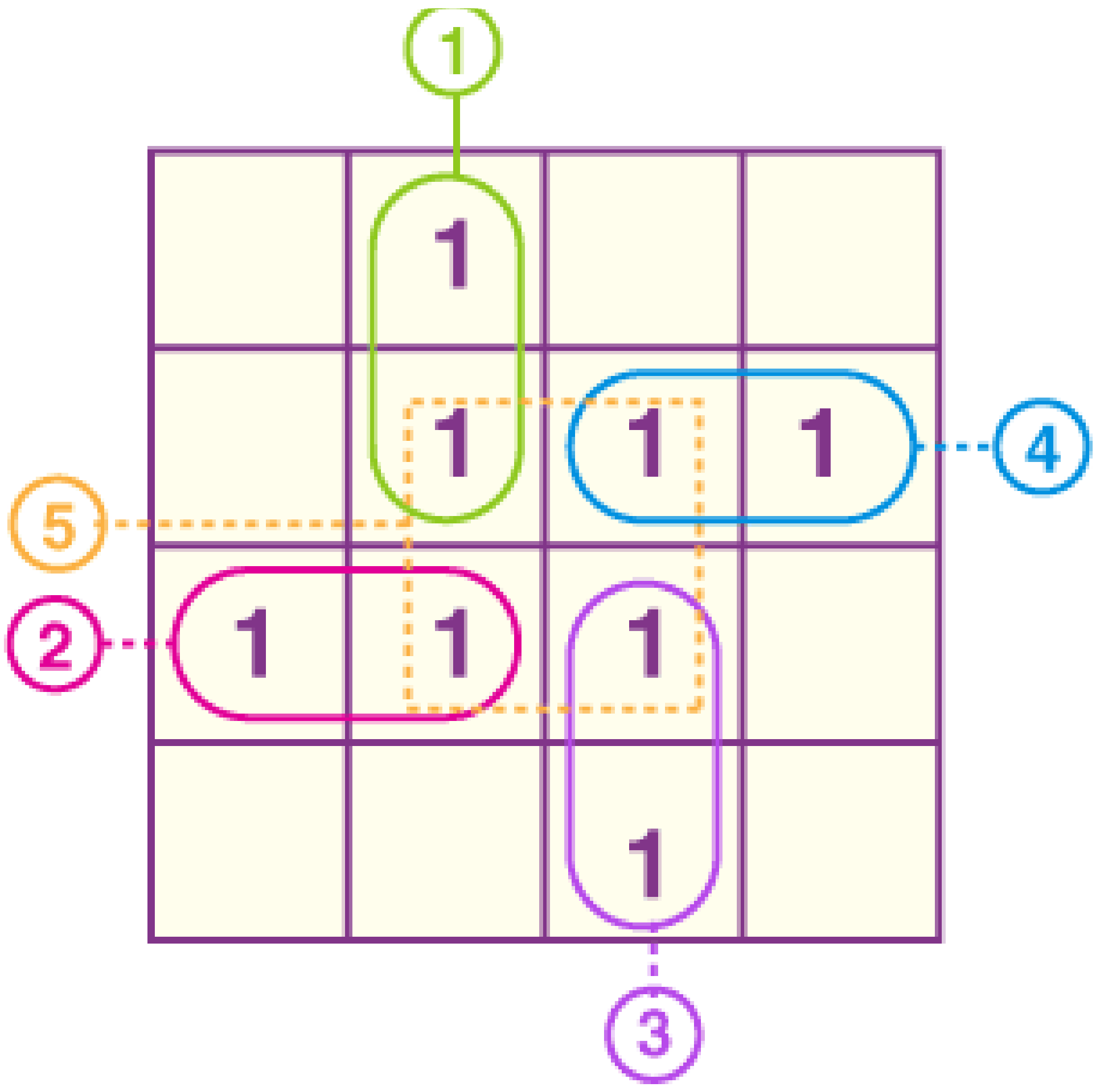
Example



No. of Selective Prime Implicants = 2

Example-1

Find the number of implicants, EPI, PI, RPI and SPI if F = ∑(1, 5, 6, 7, 11, 12, 13, 15)



F = 1 + 2 + 3 + 4

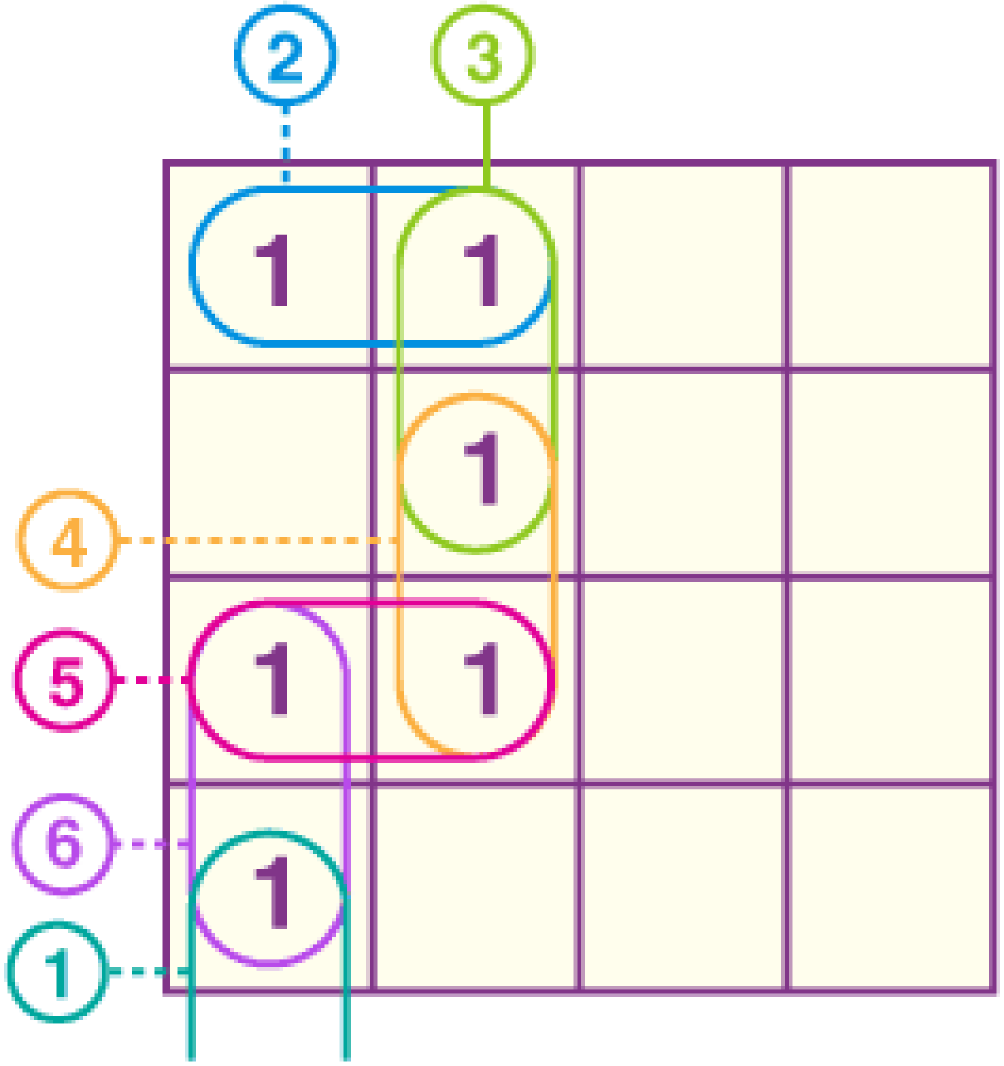
No. of Implicants = 8

PI = (1,2,3,4,5)

EPI = (1,2,3,4)

RPI = (5)

Find the number of implicants, EPI, PI, RPI and SPI if F = ∑(0, 1, 5, 8, 12, 13)



F = 1 + 3 + 5

OR

F = 2 + 4 + 6

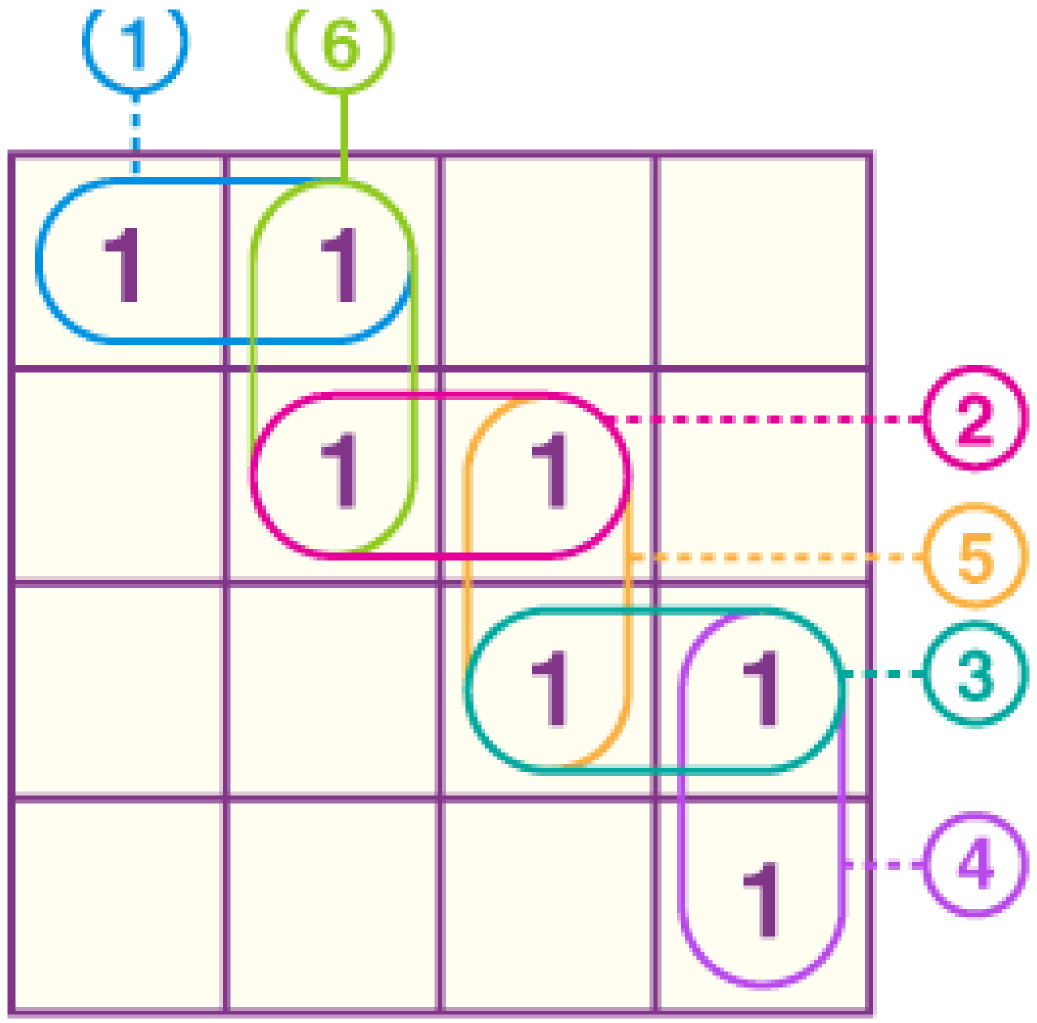
No. of Implicants = 6

PI = (1,2,3,4,5,6)

SPI = (1,2,3,4,5,6)

Example-3

Find the number of implicants, EPI, PI, RPI and SPI if F = ∑(0, 1, 5, 7, 15, 14, 10)



No. of Implicants = 7

PI = (1,2,3,4,5,6)

EPI = (1,4)

SPI = (2,3,5,6)

F = 1 + 2 + 3 + 4

OR

F = 1 + 5 + 6 + 4