Veitch Karnaugh Map (K-map)

K-map is the graphical method of the representation of the truth table interms of either minterms or maxterms. we can also define the K-map is the two dimensional representation of the minterms or maxterms. K-map is used for the simplification of the boolean functions.
It can be used to represent functions of any number of variables. But they are difficult to handle if the variables exceeds 5. Therefore this method is really suitable for four or less variables.

K-map is a diagram interms of the squares, each square represent either minterm or maxterm. A map of n variables is made up of 2^n squares. we write 1's in the square which correspond to the sum of products or minterms. similarly we write 0s in the square that correspond to the product of the sum. The K-maps for the different variables are shown as follows.

① K-map for 2 variables.

A \ B	B'	B
A'	A'B'	A'B
A	AB'	AB

A \ B	0	1
0	m_0	m_1
1	m_2	m_3

② K-map for three variables:-

A \ BC	B'C'	B'C	BC	BC'
A'	A'B'C'	A'B'C	A'BC	A'BC'
A	AB'C'	AB'C	ABC	ABC'

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
A { 1	m_4	m_5	m_7	m_6

B (columns 11, 10); C (columns 01, 11)

③ K-map for four variables:-

AB \ CD	C'D'	C'D	CD	CD'
A'B'	A'B'C'D'	A'B'C'D	A'B'CD	A'B'CD'
A'B	A'BC'D'	A'BC'D	A'BCD	A'BCD'
AB	ABC'D'	ABC'D	ABCD	ABCD'
AB'	AB'C'D'	AB'C'D	AB'CD	AB'CD'

AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
A { 11	m_{12}	m_{13}	m_{15}	m_{14}
A { 10	m_8	m_9	m_{11}	m_{10}

C (columns 11, 10); B (rows 01, 11); D (columns 01, 11)

Rules for simplification of boolean function by using K-map

(I) Every square containing 1 must be considered at least once.

(II) Square containing 1 can be included in more than one groups.

(III) Look 1s in the map & make groups of 2, 4, 8, 16 etc adjacent squares with values 1, if possible

(IV) Group must be as large as possible

(V) Grouping of 1s should be made to minimize the total no. of literals.

(VI) The map is considered to be folded or cylindrical.

(VII) Logic expression must be in canonical form before using K-map

In any four variable K-map,

One square represents one minterm giving a term of four literals.

Two adjacent squares represent a term of three literals

Four " " " " " " two literals

Eight " " " " " " one literals

Sixteen " " " " the function equals to 1.

Don't Care Condition

Generally if the minterms values of specified function are available it is not difficult and we can proceed easily. Sometime we encounter the situation in which some of minterms can't occur. In such situation it doesnot matter that the function produce 0 or 1 for a given minterm. Since the function may be either 0 or 1, we say that we don't care what the function output is to be for this minterm. Hence the minterm that may produce either 0 or 1 for the function are said to be don't care conditions.

Don't care condition is indicated by either X or ϕ on K-map. A don't care square may be assumed ~~as~~ either as 1 square or 0 square as derived while forming groups of squares for simplification. Any one such square of some of them may be included or may not be included while forming groups. Consider the following boolean function together with don't care minterms.

$$F(a,b,c) = \sum(0,2,6) \quad \& \quad \sum_{\phi}(1,3,5)$$

A \ BC	00	01	11	10
0	1	ϕ	ϕ	1
1		ϕ		1

Here, making groups $F = A' + BC'$ is the simplified function.

(11) $F(w,x,y,z) = \sum(0,1,3,5,9,11,15) + \sum_{\phi}(2,13)$

Making the groups

wx \ yz	00	01	11	10
00	1	1	1	ϕ
01		1		
11		ϕ	1	
10		1	1	

$F = w'x' + y'z + wz$ is the

simplified function.

Ques.

1. Find the truth tables for following functions. Also draw the logic circuits.

(i) $F = x\bar{y} + \bar{x}\bar{y}$

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

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

(iv) $F = x(\bar{y}+\bar{z}) + x\bar{y}$

(v) $F = xz + x\bar{y} + \bar{x}\bar{z}$

2. Examine the validity of the following boolean equations.

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

ii. $\bar{A}B\bar{C} + \bar{A}B\bar{C} + B\bar{C} = B\bar{C}$

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

iv. $xz + y\bar{x} + yz = \bar{x}z + \bar{y}x$

v. $z\bar{x} + zxy = zx$

3. Simplify the following boolean equation using boolean algebra & draw their circuits

a. $a\bar{b}c + \bar{a}\bar{b}c + a\bar{b}\bar{c} + \bar{a}\bar{b}\bar{c}$

b. $x\bar{y}\bar{z} + x\bar{y}\bar{z} + x\bar{z}$

c. $\bar{x}\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + xy\bar{z} + xyz$

d. $\bar{x}\bar{y}\bar{z} + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xy\bar{z}$

Simplify the following functions

i. $F(a,b,c) = \sum(0,2,6) + \sum_{\phi}(1,3,5) \rightarrow \bar{a} + b\bar{c}$

ii. $F(a,b,c,d) = \sum(0,1,3,5,9,11,15) + \sum_{\phi}(2,13) \rightarrow \bar{a}\bar{b} + \bar{c}d + ad$

iii. $Z(w,x,y,z) = \sum(5,6,7,8,9) + \sum_{\phi}(10,11,12,13,14,15)$

iv. $Z(w,x,y,z) = \sum(0,2,3,6,7,12,13,14) + \sum_{\phi}(1,4,11,15) \rightarrow w'x + wx + wy$

v. $F(w,x,y,z) = \sum(1,3,7,11,15) + \sum_{\phi}(0,2,5) \longrightarrow yz + w'z$

Using K-Map simplify the following boolean functions

(I) $Z = a'b' + ab'$

(II) $Z = a'bc' + abc' + abc + ab'c$

(III) $F = a'bc' + abc' + ab'c'$

(IV) $F = ab'c + abc' + a'bc + abc + ab'c' + a'b'c'$

(V) $F = a'b'c'd' + a'b'c'd + a'b'cd + a'b'cd'$

(VI) $Z = w'x'y'z' + w'x'y'z + w'x'yz + w'x'yz' + w'xyz$

Simplify the following functions using K-map.

i. $F(A,B,C) = \Sigma(3,4,6,7)$

ii. $F(A,B,C) = \Sigma(0,2,4,5,6)$

iii. $F(A,B,C,D) = \Sigma(0,1,2,8,9,10)$

iv. $F(A,B,C,D) = \Sigma(2,3,4,5,6,7,11,14,15)$

v. $F(w,x,y,z) = \Sigma(0,3,6,7,8,10,11,12,13)$

vi. $F(w,x,y) = \Sigma(0,1,2,6,7)$

vii. $f(w,x,y) = \Sigma(4,5,6,7)$

viii. $Z(a,b,c) = \Sigma(0,1,2,3,4,5)$

ix. $Z(x,y,z) = \Sigma(2,3,4,6,7)$

x. $F(a,b,c,d) = \Sigma(0,2,4,6,8,10,12,14,15)$

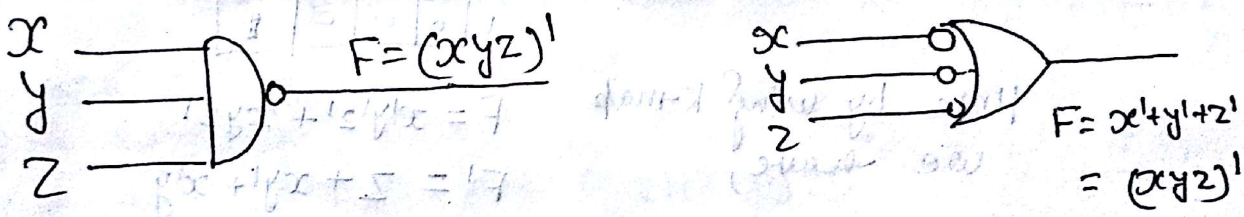
xi. $F(a,b,c,d) = \Sigma(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)$

xii. $F(x,y,z) = \Sigma(2,3,6,7)$

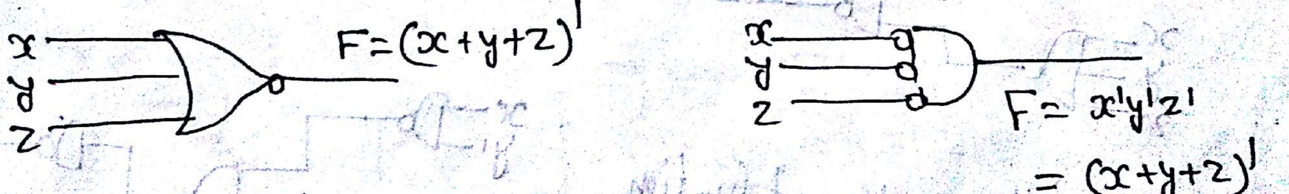
NAND and NOR gate implementation:

As we know that both of these NAND and NOR gates are the universal gates. Also most of the digital circuits are constructed with the help of these gates than the AND or OR gates. Similarly these gates are easy to fabricate within the electronic circuits.

The NAND gate can also be represented by AND-invert or the inverted-OR gate. Similarly The NOR gate can also be represented by OR-invert or the inverted AND gate. which can be shown as follow—



NAND gate representation



NOR gate representation

NAND gate implementation

In this case first of all we simplify the function in Sum of product form. After that we simplify the complement of the function in the sum of product form. This can be done by combining the 0's in the map. After the simplification of both true and the complement function we use the NAND gates for each product terms and finally again use the NAND gate in the 2nd level for finding the result. But in the case of complement function once NAND gate is again used to get the true function.

Example Implement the following function with NAND gates.

$F(x,y,z) = \Sigma(0,6)$

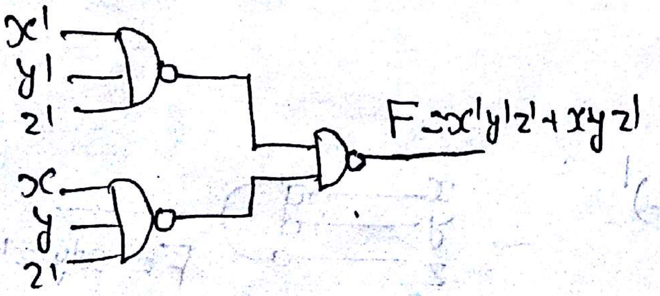
$x \backslash yz$	00	01	11	10
0	1	0	0	0
1	0	0	0	1

Here by using K-map we have,

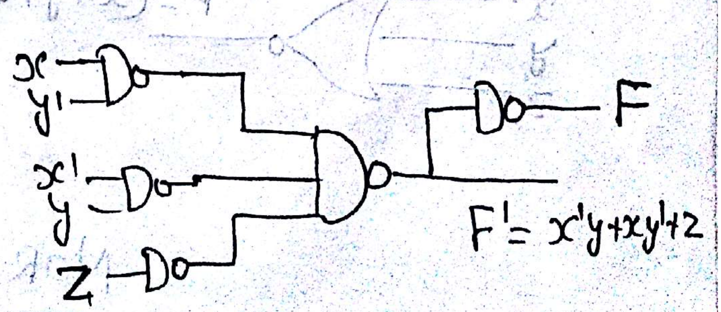
$F = x'y'z' + xyz'$

$F' = z + xy' + x'y$

Now using NAND gates for each product in true function



IIly In the complement function



NOR gate implementation

Actually the NOR function is the dual of the NAND function. Hence all the procedures of NAND implementation are done and duality principle is applied here. Here we have to use the NOR gates and hence the function be simplified in the product of sums form. This expression specifies a group of OR gates for the sum terms.

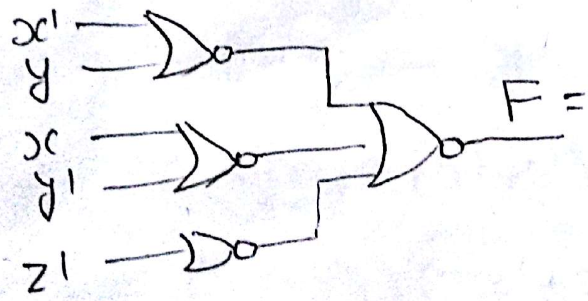
Example Implement $F(x,y,z) = \Sigma(0,6)$ with NOR gate using K Map

x \ yz	00	01	11	10
0	1	0	0	0
1	0	0	0	1

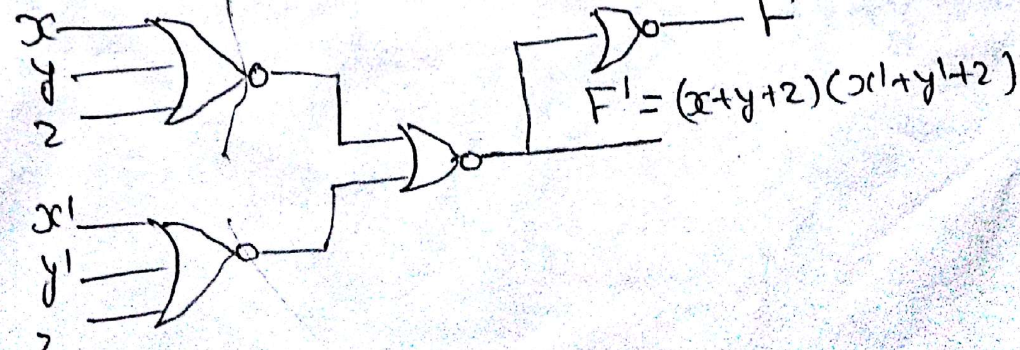
Here $F = x'y'z' + xyz'$
$F' = z + xy' + x'y$

Now, complementing these function $F' = (x+y+z)(x'+y'+z)$
$F = z'(x'+y)(x+y')$

Now using NOR gates for each sum term in true function



Similarly in the complemented function



Using Boolean algebra Prove the followings

(I) $xy + \bar{x}z + yz = xy + \bar{x}z$

$LHS = xy + \bar{x}z + yz$

$= xy + \bar{x}z + yz(x + \bar{x})$

$= xy + \bar{x}z + xyz + \bar{x}yz$

$= xy(1+z) + \bar{x}z(1+y)$

$= xy + \bar{x}z$

$=$ RHS Hence proved.

(II) $(x+y)(x\cancel{z}+z)\cdot(\bar{y}+xz)' = \bar{x}yz$

$LHS = (x+y)(xz+z)\cdot(\bar{y}+xz)'$

$= (x+y)(x+1)z \quad y\cdot(xz)'$

$x+1 = 1$

$= (x+y)yz(x'+z')$

$= (xy+yy)(zx'+zz')$

$= (xy+y)(zx') \qquad \because zz' = 0$

$= yzx' \qquad \because xy+y = y$

$=$ RHS proved.

Simplify the following boolean expression

(I) $xy + xyz + x(y+xy)$

$= xy + xyz + xy + xy$ $\quad \because xx = x$

$= xy + xyz + xy$ $\quad \because xy + xy = xy$

$= xyz + xy$ $\quad \because$ "

$= xy(z+1)$

$\quad \because z+1 = 1$

$= xy$

(II) $(x+y)(x+\bar{y})(\bar{x}+z)$

$= (x + y\bar{y})(\bar{x}+z)$ [Using distributive law]

$= x(\bar{x}+z)$ $\quad \because y\bar{y} = 0$

$= x\bar{x} + xz$

$= xz$ $\quad \because x\bar{x} = 0$

(III) $(x+y)(yz+z)$

$= (x+y)(y+1)z$

$= (x+y)z$

$= xz + yz$

Simplify the following Boolean function in POS form using K-map.

$F = \prod M(0,1,2,9,11,14)$ & don't care conditions

$D = \prod M(7,8,12)$

Soln drawing the K-map for four variables & filling the maxterms as given

wx \ yz	00	01	11	10
00	0	0		0
01			X	
11	X			0
10	X	0	0	0

Making groups of maxterms we get 4 groups & can write the Boolean function as

$$F = (w'+x)(x+z)(x+y)(w'+z).$$

POS simplification method using K-map.

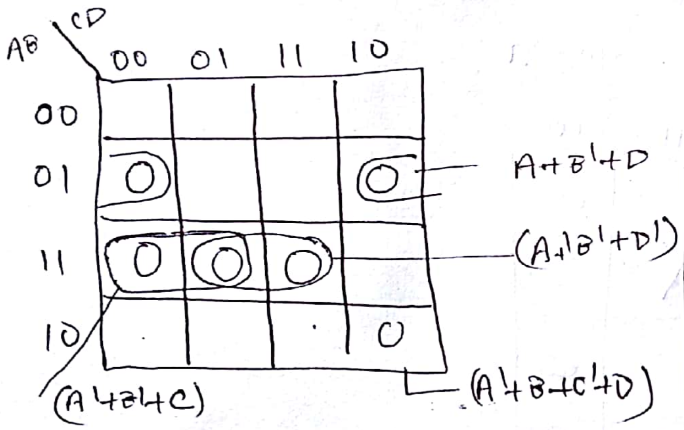
\# Simplify the Boolean function

ⓐ $f(ABCD) = \Pi(4, 6, 10, 12, 13, 15)$

ⓑ $f(ABCD) = \Pi(4, 5, 6, 14, 15)$

Soln for ⓐ

Drawing the K map for ⓐ we get

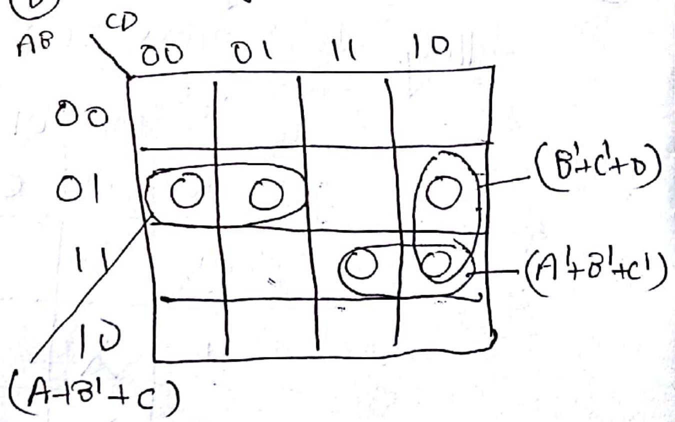


Here we are using POS & hence making the pairs we get the simplified function as

$$F = (A'+B'+D')\cdot(A'+B'+C)\cdot(A+B'+D)\cdot(A'+B+C'+D)$$

Soln for ⓑ

Drawing the K-map for ⓑ we get



Here we are using POS & hence the pairs give the simplified function as

$$F = (A+B'+C)\cdot(A'+B'C')\cdot(B'+C'+D)$$

Simplification of boolean function by prime implicant method

The tabulation method consists of the list of minterms that specify the function. The 1st operation is to find the prime implicants by using the matching process. The process compares each minterm with every other minterm. If two minterms differ in only one variable, that variable is removed & a term with one less literal is found. This process is repeated for every minterm until the exhaustive search is completed. The matching process cycle is repeated for those new terms just found. Third & further cycles are continued until a single pass through a cycle yields no further elimination of literals. The remaining terms and all the terms that did not match during the process comprise the prime implicants. Consider the following example.

① $F(a,b,c,d) = \Sigma(0,1,2,8,10,11,14)$

Here the function consists of four literals hence representing the minterms interms of 4 bit binary equivalent we get & making groups of different patterns having no any 1, having only one, having 2 ones & so on

0	0000	no any 1
1	0001	1 no. of 1
2	0010	
8	1000	
10	1010	2 no. of 1
11	1011	3 no. of 1
14	1110	

Now comparing the minterms having no any ones with having 1 no. of 1, minterms having 1 no. of 1 with minterms having 2 no. of 1 and so on we get

0	0000✓				
1	0001✓	(0,1)	000— X	(0,2,8,10)	—0—0 X
2	0010✓	(0,2)	00—0✓	(0,8,2,10)	—0—0 X
8	1000✓	(0,8)	—000✓		
10	1010	(2,10)	—010✓		
11	1011	(8,10)	10—0✓		
14	1110	(10,11)	101— X		
		(10,14)	1—10 X		

Here the prime implicants are $a'b'c'$, $ab'c$, acd' and $b'd'$

Hence the simplified function is

$$F(a,b,c,d) = a'b'c' + ab'c + acd' + b'd'$$

By K-map

	$c'd'$	$c'd$	cd	cd'
$a'b'$	1	1		1
$a'b$				
ab				1
ab'	1		1	1

$$\boxed{F(a,b,c,d) = b'd' + ab'c + acd' + a'b'c'}$$

Scanned by CamScanner

* $F(w,x,y,z) = \sum(1,4,6,7,8,9,10,11,15)$

Representing the minterms in 4 bit binary equivalent & arranging them in different groups.

1	0001
4	0100
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
15	1111

1	0001 ✓	(1,9)	-001 ×	(8,10,9,11)	10-- ×
4	0100 ✓	(4,6)	01-0 ×	(8,9,10,11)	10--
8	1000 ✓	(8,9)	100- ✓		
6	0110 ✓	(8,10)	10-0 ✓		
9	1001 ✓	(6,7)	011- ×		
10	1010 ✓	(9,11)	10-1 ✓		
7	0111 ✓	(10,11)	101- ✓		
11	1011 ✓	(7,15)	-111 ×		
15	1111 ✓	11,15	1-11 ×		

Here the prime implicants

Decimal	Binary	Term
(1,9)	-001	x'y'z
(4,6)	01-0	w'xz'
(6,7)	011-	w'xy
(7,15)	-111	xyz
(11,15)	1-11	wyz
(8,9,10,11)	10--	wx'

> The selection of prime implicants that form the minimized function is made from prime implicant table. The table consists of the row & column as follow.

			1	4	6	7	8	9	10	11	15
✓	$x'y'z$	1,9	X					X			
✓	$w'xz'$	4,6		X	X						
	$w'xy$	6,7			X	X					
	xyz	7,15				X					X
	wyz	11,15								X	X
✓	wx'	8,9,10,11					X	X	X	X	
			✓	✓	✓		✓	✓	✓	✓	

Here the essential prime implicants are selected (The prime implicants that covers minterms with a single cross in their column are called essential prime implicants.)

Here the four minterms whose column have a single cross 1, 4, 8 & 10. Minterm 1 is covered by $x'y'z$, 4 is covered by $w'xz'$, 8 & 10 is covered by wx' & guarantees that these minterms are included in the function.

Now we include essential prime implicants & a checkmark is placed in the table next to essential prime implicants to indicate that they have been selected.

Next we check each column whose minterm is covered by the selected essential prime implicants. eg the selected prime implicant $x'y'z$ covers the minterms 1 & 9. A check is inserted in the bottom of the columns. Illy prime implicant $w'xz'$ covers minterms 4 & 6 & wx' covers minterms 8, 9, 10 & 11. Inspection of prime implicant table shows that the selection of the essential prime implicants covers all the minterms

of

of the function except 7 & 15. These two minterms must be included by the selection of one or more prime implicants. In this it is clear that prime implicant xyz covers both minterms and is therefore the one to be selected. Hence we found the minimum set of prime implicants whose sum gives the required minimized function given by.

$$F = x'y'z + w'xz' + wx' + xyz$$

simplify the following Boolean function by prime implicant method.

$$F(a,b,c,d) = \Sigma(0,1,2,8,10,14,15)$$

Soln Here the function consists of four variables. Hence representing the minterms interms of 4 bit binary equivalent we get

0	0000
1	0001
2	0010
8	1000
10	1010
11	1011
14	1110
15	1111

(0,1)	000—	X
(0,2)	00—0	✓
(0,8)	—000	
(2,10)	—010	
(8,10)	10—0	✓
(10,11)	101—	✓
(10,14)	1—10	✓
(11,15)	1—11	✓
(14,15)	111—	✓

(0,2,8,10)	—0—0	X
(10,11,14,15)	1—1—	X
(10,14,11,15)	1—1—	
(0,8,2,10)	—0—0	

Here the prime implicants are $a'b'c'$, $b'd'$ & ac

Hence the simplified function is

$$F(a,b,c,d) = a'b'c' + b'd' + ac$$

Scanned by CamScanner

using k-map

ab \ cd	c'd'	c'd	cd	cd'
a'b'	1	1		1
a'b				
ab			1	1
ab'	1		1	1

$$F(a,b,c,d) = ac + b'd' + a'b'c'$$

Simplify the following Boolean function.

$$F(w,x,y,z) = \Sigma(0,3,6,7,8,10,11,12,13)$$

Sol^n Representing the minterms interms of 4-bit binary representation we have

0	0000	0	0000 ✓	(0,8)	-000	x
3	0011	8	1000 ✓	(8,10)	10-0	x
6	0110	3	0011	(8,12)	1-00	x
7	0111	6	0110	(3,7)	0-11	x
8	1000	10	1010 ✓	(3,11)	-011	x
10	1010	12	1100 ✓	(6,7)	011-	x
11	1011	07	0111 ✓	(10,11)	101-	x
12	1100	11	1011	(12,13)	110-	x
13	1101	13	1101			

Now collecting the prime implicants we get

$$F(w,x,y,z) = x'y'z' + wx'z' + wy'z' + w'yz + x'yz + w'xy + wx'y + wxy'$$

By K-map we get

wx \ yz	y'z'	y'z	yz	yz'
w'x'	1		1	
w'x			1	1
wx	1	1		
wx'	1		1	1

wx \ yz	y'z'	y'z	yz	yz'
w'x'	1		1	
w'x			1	1
wx	1	1		
wx'	1		1	1

$$F = wx'y + w'xy + wxy' + w'yz + wy'z' + \cancel{x'y'z'}$$

Here the prime implicants

Decimal	Binary	Terms
0, 8	–000	$x'y'z'$
8, 10	10–0	$wx'z'$
8, 12	1–00	$wy'z'$
3, 7	0–11	$w'yz$
3, 11	–011	$x'yz$
6, 7	011–	$w'xy$
10, 11	101–	$wx'y$
12, 13	110–	wxy'

Now the prime implicant table is as follow

			0	3	6	7	8	10	11	12	13
✓	$x'y'z'$	0,8	X				X				
	$wx'z'$	8,10					X	X			
	$wy'z'$	8,12					X			X	
	$w'yz$	3,7		X		X					
✓	$x'yz$	3,11		X					X		
✓	$w'xy$	6,7			X	X					
✓	$wx'y$	10,11						X	X		
✓	wxy'	12,13								X	X
			✓		✓	✓	✓			✓	✓

Here the essential prime implicants are selected. The minterms whose column have a single cross 0, 6, 13. minterm 0 is covered by $x'y'z'$, 6 is covered by $w'xy$ & 13 is covered by wxy' & guarantees that these minterms are included in the function.

Now we include essential prime implicants & a check mark is placed in the table next to essential prime implicants to indicate that they have been selected. Next we check each column whose minterm is covered by the selected essential prime implicants. eg. the selected prime implicant $x'y'z'$ covers the minterms 0 & 8. A check is inserted in the bottom of the columns. Ily prime implicant $w'xy$ covers the minterms 6 & 7 and wxy' covers the minterms 12 & 13. Inspection of prime implicant table shows that the selection of essential prime implicants cover all the minterms of the function except 3, 10 & 11. These three minterms must be included by the selection of one or more prime implicants. In this it is clear that

$x'yz$ covers the minterms 3 and 11 and
$wx'y$ covers the minterms 10 and 11 and

hence are to be selected. Hence we form the minimum set of prime implicants whose sum gives the required function given by

$$F(wxyz) = x'y'z' + w'xy + wxy' + x'yz + wx'y$$