Digital Circuit
&
Logic Design
(CSDC-0101)

Gate Level Minimization

The Map Method

In a digital circuit with multiple digital inputs and multiple digital outputs, the outputs depend on the current value of inputs.

A Boolean function can be represented in the form of **sum of minterms** or **the product of maxterms**, which enable the designer to make a truth table more easily.

In a truth table, an output of one represents minterms, and an output of zero represent maxterms.

Also, Boolean functions can be simplified using **Karnaugh map (K-map)** without using Boolean theorems, by transferring a function to K-map and reading simplified function from K-map.

Minterm

x	y	minterm	designation
0	0	x'y'	m_0
0	1	x'y	m_1
1	0	xy'	m_2
1	1	xy	m_3

A minterm is associated with each combination of variables in a function.

If a function has n variables, then it has 2^n minterms.

It is simple to generate a truth table from minterms and vice versa.

Consider two Boolean variables, X and Y.

There are four different possible combinations that can be generated from X AND Y, and they are
X' Y', X' Y, X Y', and XY.

These four combinations are called the minterms for X AND Y .

E.g. F(X,Y) = X Y' + X' Y

This may also be rewritten as F (X, Y) = Σ(1,2) where F is sum of minterms (or where F = 1):.

This function can be represented as F(X,Y) = m1 + m2 or each minterm that represents a one in the truth table.

Each minterm is obtained from an AND term of the n variables, with each variable being primed if the corresponding bit of the binary number is 0 and unprimed if a 1.

$F(X,Y,Z) = X'Y'Z + X'YZ + XYZ$

X Y Z	Minterms	Designation
0 0 0	$X'Y'Z'$	m_0
0 0 1	$X'Y'Z$	m_1
0 1 0	$X'YZ'$	m_2
0 1 1	$X'YZ$	m_3
1 0 0	$XY'Z'$	m_4
1 0 1	$XY'Z$	m_5
1 1 0	XYZ'	m_6
1 1 1	XYZ	m_7

X Y Z	F
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	0
1 1 1	1

Maxterm

Maxterm is complement of a minterm.

If the minterm m_0 is X Y Z, then the maxterm M_0 is

(X’ Y’ Z’)’ = X + Y + Z

x	y	minterm	designation
0	0	x+y	M_0
0	1	x+y’	M_1
1	0	x’+y	M_2
1	1	x’+y’	M_3

In a truth table, an output of one represents minterms, and an output of zero represent maxterms.

The function F can be expressed as the product of maxterms.

$F(X,Y,Z) = M_0 M_2 M_4 M_5 M_6$, or it can represented by

$F(X,Y,Z) = \pi(0,2,4,5,6)$

Each maxterm is obtained from an OR term of the n variables, with each variable being unprimed if the corresponding bit is a 0 and primed if a 1.

F(X,Y,Z) = (M0,M2,M4,M5,M6)

X Y Z	Maxterm	Designation
0 0 0	$X + Y + Z$	M_0
0 0 1	$X + Y + Z'$	M_1
0 1 0	$X + Y' + Z$	M_2
0 1 1	$X + Y' + Z'$	M_3
1 0 0	$X' + Y + Z$	M_4
1 0 1	$X' + Y + Z'$	M_5
1 1 0	$X' + Y' + Z$	M_6
1 1 1	$X' + Y' + Z'$	M_7

X Y Z	F
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	0
1 1 1	1

				Minterms		*Maxterms*
X	*Y*	*Z*		*Product Terms*		*Sum Terms*
0	*0*	*0*		$m_0 = \overline{X} \cdot \overline{Y} \cdot \overline{Z} = \min(\overline{X}, \overline{Y}, \overline{Z})$		$M_0 = X + Y + Z = \max(X, Y, Z)$
0	*0*	*1*		$m_1 = \overline{X} \cdot \overline{Y} \cdot Z = \min(\overline{X}, \overline{Y}, Z)$		$M_1 = X + Y + \overline{Z} = \max(X, Y, \overline{Z})$
0	*1*	*0*		$m_2 = \overline{X} \cdot Y \cdot \overline{Z} = \min(\overline{X}, Y, \overline{Z})$		$M_2 = X + \overline{Y} + Z = \max(X, \overline{Y}, Z)$
0	*1*	*1*		$m_3 = \overline{X} \cdot Y \cdot Z = \min(\overline{X}, Y, Z)$		$M_3 = X + \overline{Y} + \overline{Z} = \max(X, \overline{Y}, \overline{Z})$
1	*0*	*0*		$m_4 = X \cdot \overline{Y} \cdot \overline{Z} = \min(X, \overline{Y}, \overline{Z})$		$M_4 = \overline{X} + Y + Z = \max(\overline{X}, Y, Z)$
1	*0*	*1*		$m_5 = X \cdot \overline{Y} \cdot Z = \min(X, \overline{Y}, Z)$		$M_5 = \overline{X} + Y + \overline{Z} = \max(\overline{X}, Y, \overline{Z})$
1	*1*	*0*		$m_6 = X \cdot Y \cdot \overline{Z} = \min(X, Y, \overline{Z})$		$M_6 = \overline{X} + \overline{Y} + Z = \max(\overline{X}, \overline{Y}, Z)$
1	*1*	*1*		$m_7 = X \cdot Y \cdot Z = \min(X, Y, Z)$		$M_7 = \overline{X} + \overline{Y} + \overline{Z} = \max(\overline{X}, \overline{Y}, \overline{Z})$

Karnaugh Map (K-Map)

Karnaugh maps are used to simplify a Boolean function without using Boolean algebra theorems.

A K-map is also another way to represent the truth table of a function.

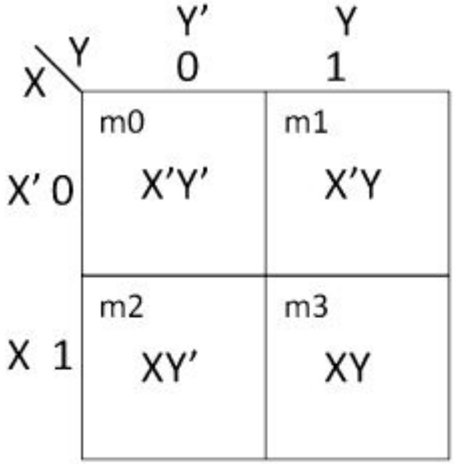
K-maps are made of cells where each cell represents a minterm. Cells marked with a one will be the minterms used for the sum of the minterms representation of a function.

Conversely, cells marked with a zero will be used for the product of the maxterms representation.

Adjacent Cell

Two cells are adjacent if they differ on only one variable. The cells X’Y’ and X’Y are adjacent because their only difference is Y’ and Y. Adjacent cells can be combined in order to simplify a K-map’s function.

X Y	Minterms	Designation
0 0	X′Y′	m_0
0 1	X′Y	m_1
1 0	XY′	m_2
1 1	XY	m_3

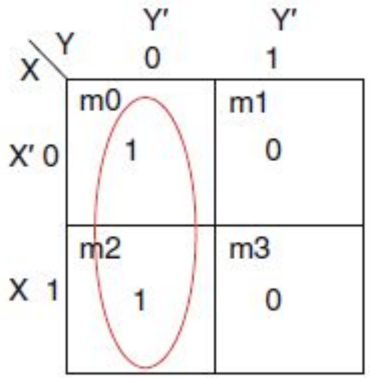


Example:

F(X,Y) = XY’ + X’Y’ = m2 + m0.

The truth table for the function and its mapping to a K-map is done as:

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



Simplify the following function:

$$F(X,Y)=X'Y+XY'+XY$$

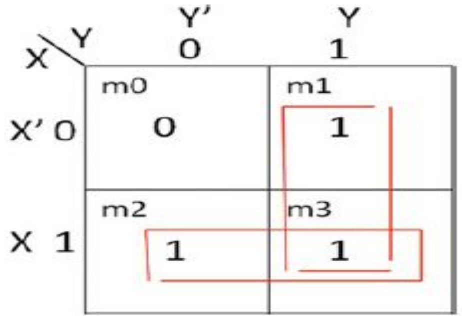
or

$$F(X,Y) = m1+m2+m3$$

The cells m2 and m3 are adjacent, so they can be combined. Likewise, the cells m1 and m3 can be combined. By reading the map, you will have the simplified function.

Cells m2 and m3 are the entire row X, and cells m1 and m3 are the entire column Y, with the other cell being zero. Therefore,

$$F(X,Y) = X+Y$$



Three-Variable Map

A three-variable K-map contains eight cells, and each cell represents a minterm.

(a) At row 0, all four cells contain X’; therefore this row is labeled X’.

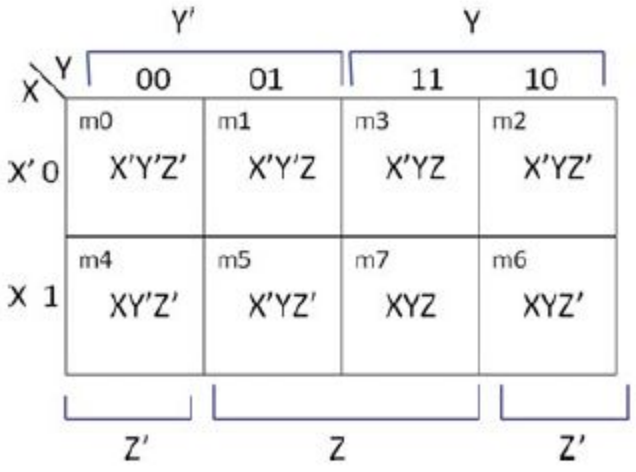
(b) At row 1, all four cells contain X; therefore this row is labeled X.

(c) At the columns 11 and 10, all four cells contain Y; therefore these columns are labeled Y.

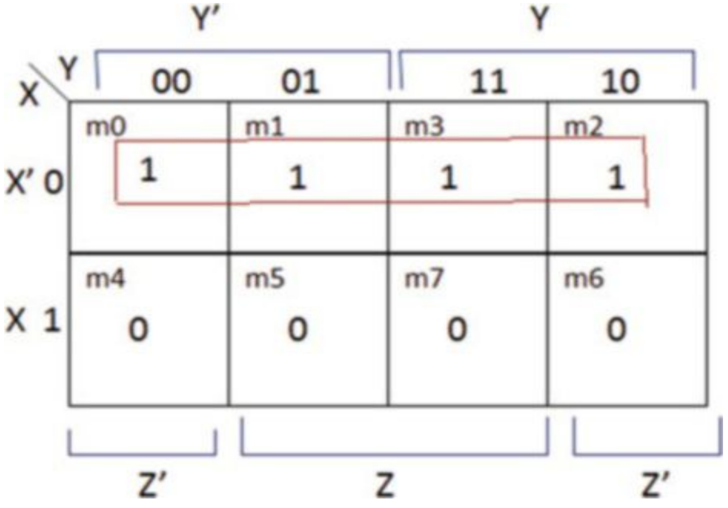
(d) At the columns 00 and 01, all four cells contain Y’; therefore these columns are labeled Y’.

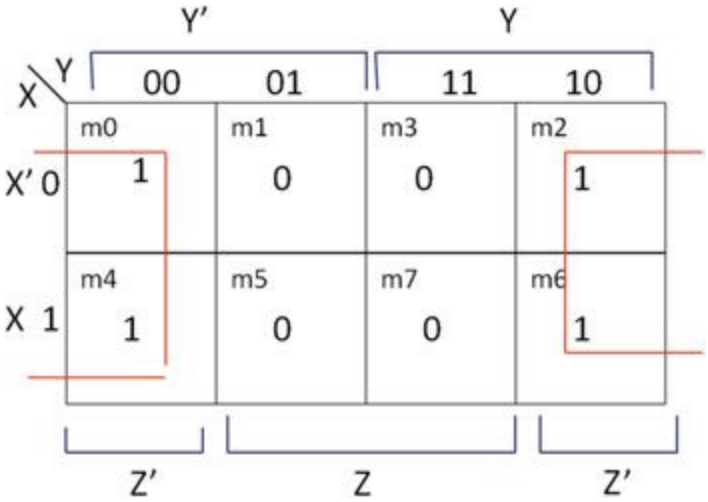
(e) At the columns 01 and 11, all four cells contain Z; therefore these columns are labeled Z.

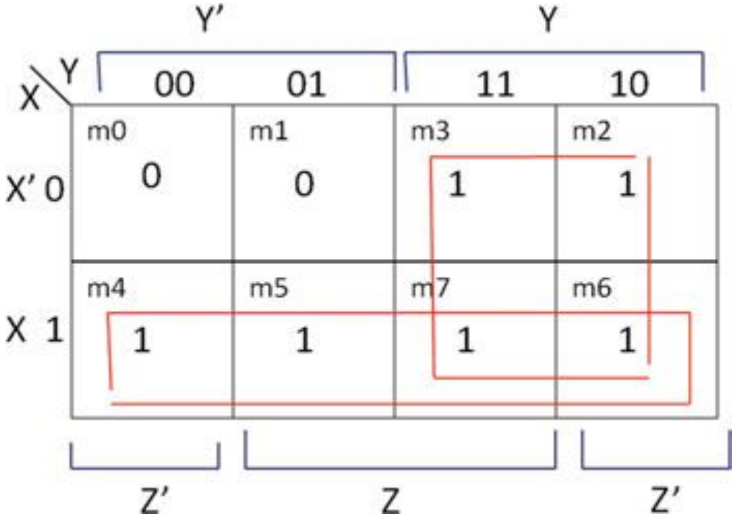
(f) At the columns 00 and 10, all four cells contain Z’; therefore these columns are labeled Z’.

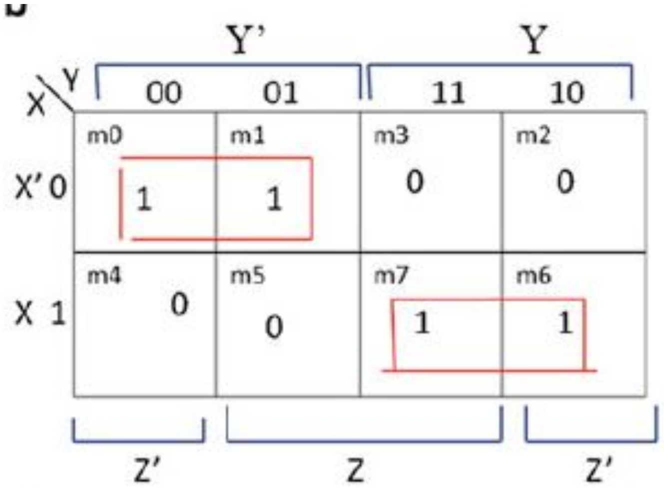


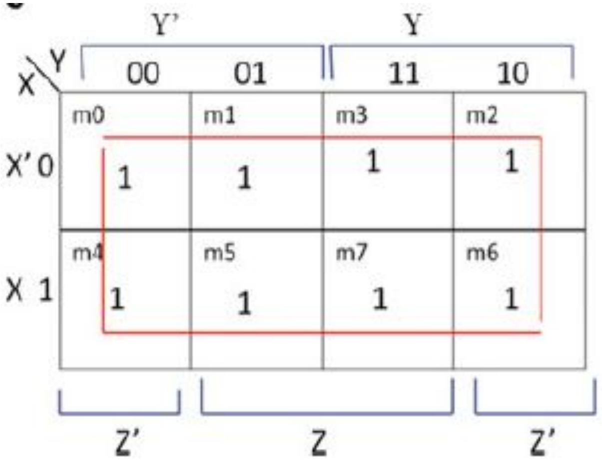
Adjacent cells can be grouped together in a K-map; in a K-map it can combine 2 cells, 4 cells, 8 cells, and 16 cells.


Y'
Y
X
Y
00
01
11
10
m0
m1
m3
m2
X' 0
1
1
1
1
m4
m5
m7
m6
X 1
0
0
0
0
Z'
Z
Z'


Y'
Y
X
Y
00
01
11
10
m0
m1
m3
m2
X' 0
1
0
0
1
m4
m5
m7
m6
X 1
1
0
0
1
Z'
Z
Z'


Y'
Y
X
Y
00
01
11
10
m0
m1
m3
m2
X' 0
0
0
1
1
m4
m5
m7
m6
X 1
1
1
1
1
Z'
Z
Z'


Y'
Y
X
Y
00
01
11
10
m0
m1
m3
m2
X' 0
1
1
0
0
m4
m5
m7
m6
X 1
0
0
1
1
Z'
Z
Z'


Y'
Y
X
Y
00
01
11
10
m0
m1
m3
m2
X' 0
1
1
1
1
m4
m5
m7
m6
X 1
1
1
1
1
Z'
Z
Z'

(a) All adjacent cells in columns Z' and columns Y are one.

F (X,Y, Z) = Z' + Y'

(b) The cells in row X', columns Y' are adjacent, as are the cells in row X, columns Y.

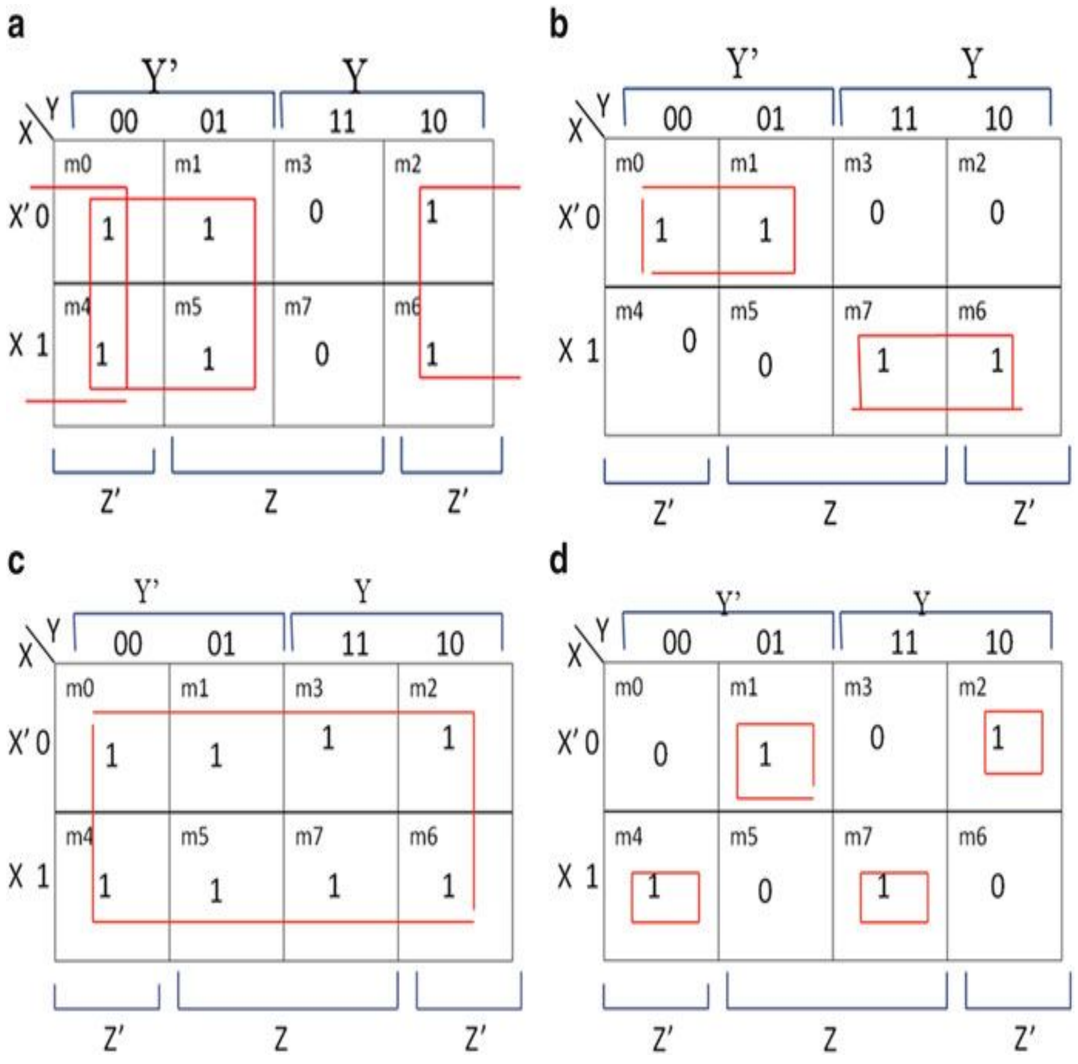
F (X,Y, Z) = X'Y'+XY

(c) All cells are one, so the function is always equal to one.

F (X,Y, Z) = 1

(d) Without adjacent cells to simplify the terms, the function equals the ones.

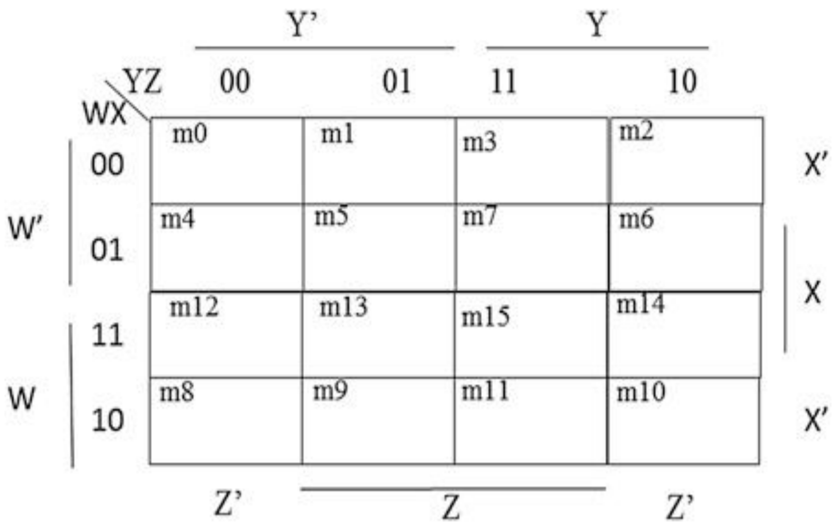
F (X,Y, Z) = X' Y'Z + X'YZ' + XY'Z' + XYZ



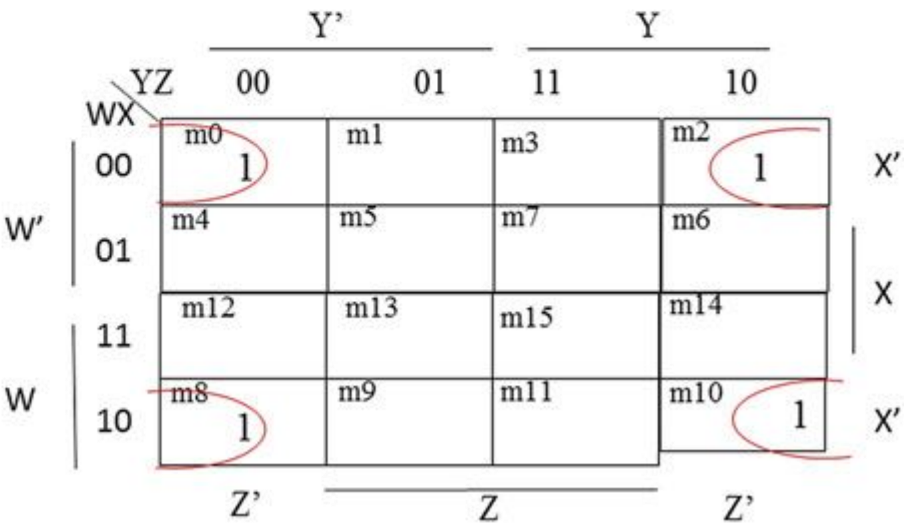
Four-Variable Map

Four-variable K-maps contain 16 cells.

- W covers rows 11 and 10.
- W' covers rows 00 and 01.
- X covers rows 01 and 11.
- X' covers rows 00 and 10.
- Y covers columns 11 and 10.
- Y' covers columns 00 and 01.
- Z covers columns 01 and 11.
- Z' covers columns 00 and 10.



K-map for F(W,X,Y,Z) = m0 + m2 + m8 + m10

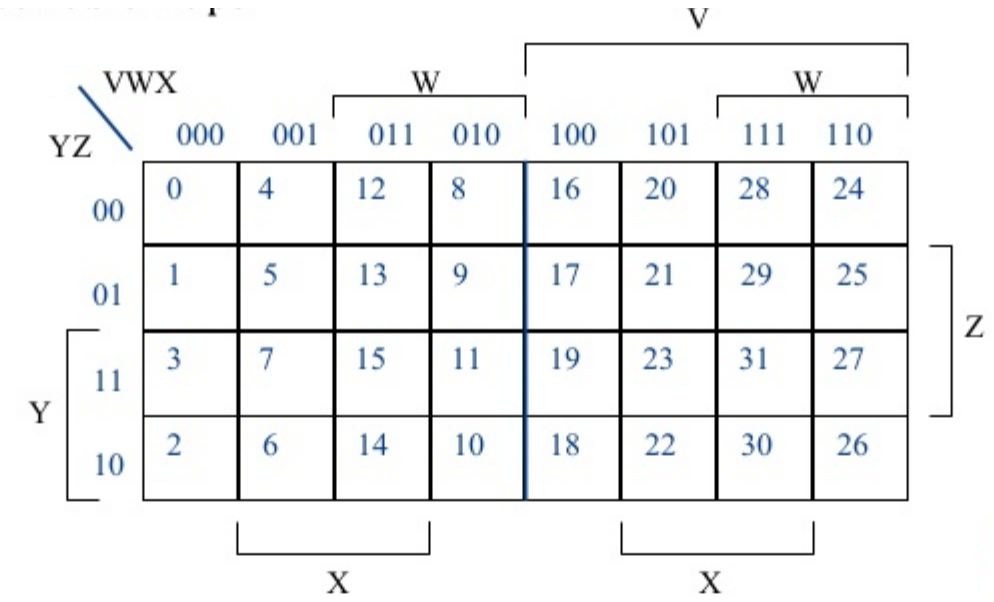


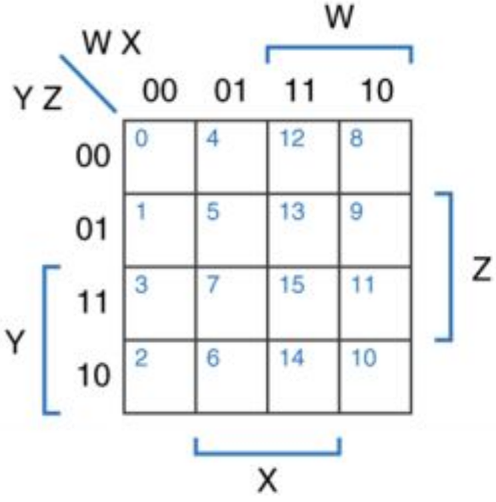
The simplified function is F(W,X,Y,Z) = X’Z’.

Five-Variable Map

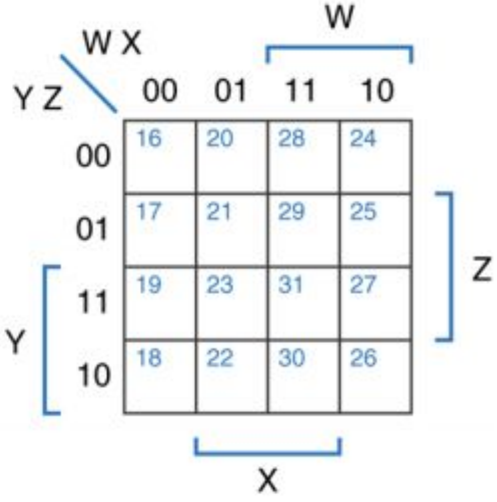
A 5-variable Boolean function can have a maximum of 32 minterms. Hence, 5-variable K-Map contains 32 cells.

A 5-variable K-map is formed using two connected 4-variable map.





V=0



V=1

Five-Variable Map

$$F(A,B,C,D,E) = \sum(0,2,4,6,9,13,21,23,25,29,31)$$

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

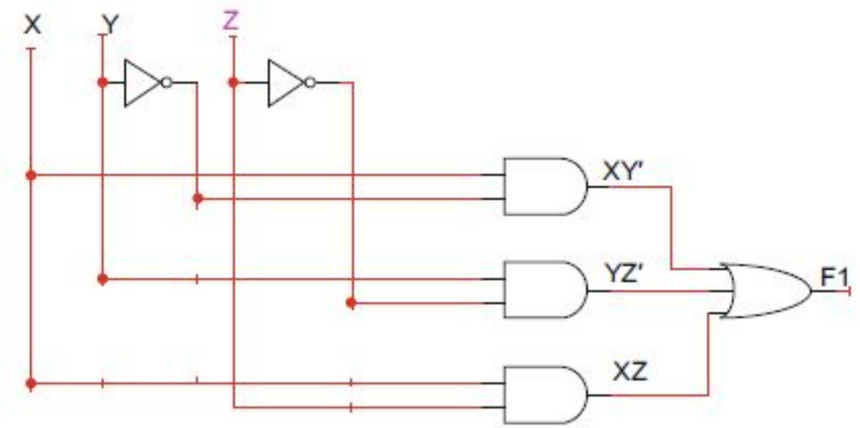
$$F(A,B,C,D,E) = \sum(0,2,4,7,8,10,12,16,18,23,24,25,26,27,28)$$

$$F(A,B,C,D,E) = \bar{D}\bar{E} + \bar{B}CDE + \bar{C}\bar{E} + AB\bar{C}$$

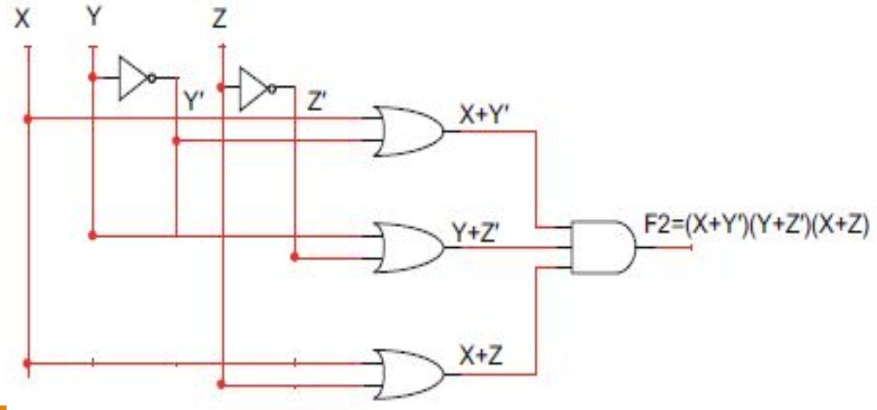
Sum of Products (SOP) and Product of Sums (POS)

The sum of products of a function is its simplified sum of minterms while product of sums of a function is its simplified product of maxterms.

Consider function F1(X,Y,Z) = XY’ + YZ’ + XZ, the logic circuit for function F1 which is made of AND-OR is:



Consider function F2(X,Y,Z) = (X + Y’)(Y + Z’)(X + Z) which is represented by the product of sums, the logic circuit for function F2 which is in the form OR-AND is:



Don’t Care Conditions

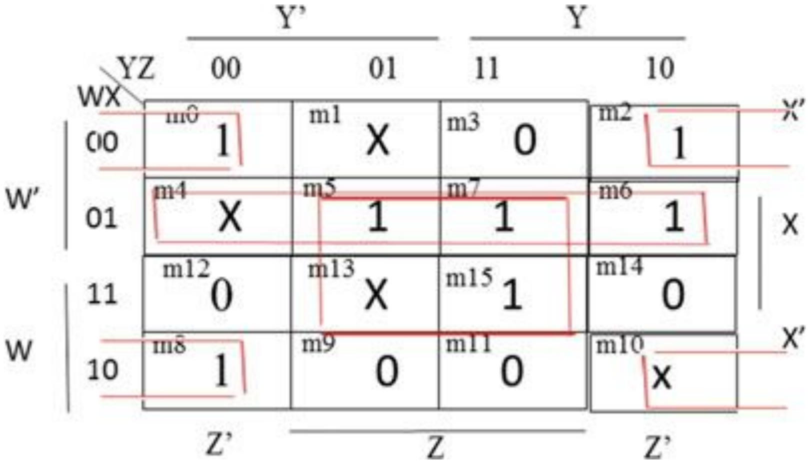
In a truth table, if certain combinations of the input variables are impossible, they are considered don’t care conditions.

These conditions are where the output of the function does not matter. For example, binary-coded decimal (BCD) is 4 bits and only 0000 to 1001 are used, so from 1010 to 1111 are not BCD; the truth table or K-map values are don’t cares.

A truth table or K-map cell marked with a “X” or “d” is a don’t care term, and output will not be affected whether it is a one or zero.

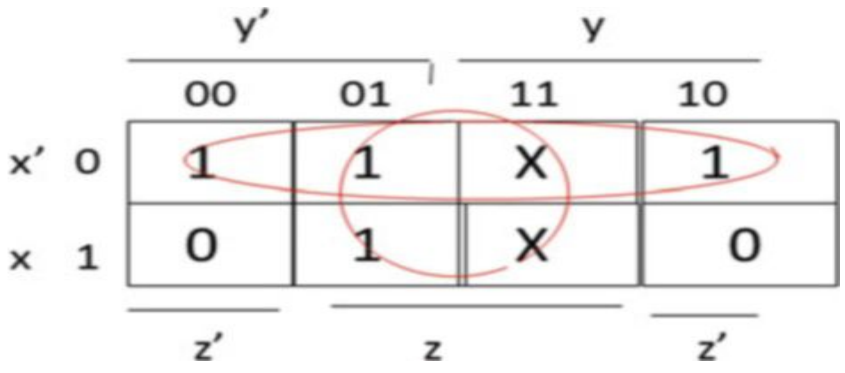
The don’t care can be used to expand the adjacency of cells in a K-map to further simplify a function, since their output does not matter.

Since a don’t care can output either a zero or one, we can assume it is a one in order to expand a grouping of adjacent cells.



K-map with don’t care minterms

F(W,X,Y,Z) = XZ + X’Z’ + XW ’



K-map for F(X,Y,Z) = m0 + m1 + m2 + m5 and D(X,Y,Z) = m3 + m7

Simplified: F(X,Y,Z) = X’+ Z.

When minterms for function F are don’t care terms, the don’t care function D is equal to the sum of the don’t care minterm(s).