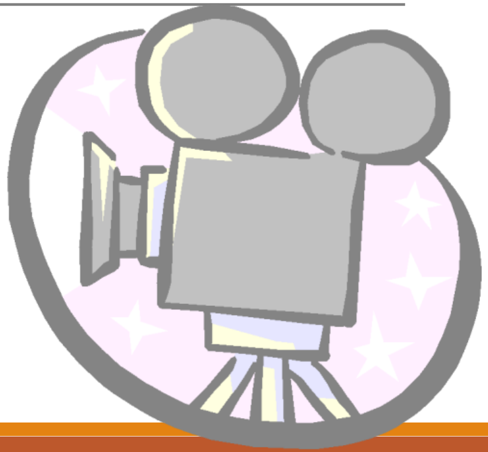
Digital Logic and Circuit
Paper Code: CS-102



Outline

- Canonical and standard form
- Conversion between Canonical Forms
- Simplification of Boolean Functions using k-map.

Conversion between Canonical Forms

- The complement of a function expressed as the sum of minterms equals the sum of minterms missing from the original function.

- This is because the original function is expressed by those minterms that make the function equal to 1, whereas its complement is a 1 for those minterms that the function is a 0.

$F(A, B, C) = \sum (1, 4, 5, 6, 7)$

This has a complement that can be expressed as
$F'(A , B, C) = \sum(0, 2, 3) = m_0 + m_2 + m_3$

Now, if we take the complement of F' by DeMorgan's theorem, we obtain F in a different form:

$$F = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' = M_0 M_2 M_3 = \Pi(0, 2, 3)$$

The last conversion follows from the definition of minterms and maxterms

$$m_j' = M_j$$

That is, the maxterm with subscript j is a complement of the minterm with the same subscript j, and vice versa.

Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

A Boolean function can be converted from an algebraic expression to a product of maxterms by using a truth table and the canonical conversion procedure.

Consider, for example, the Boolean expression

$$F = xy + x'z$$

The function expressed in sum of minterms is

$$F(x, y, z) = \Sigma(1, 3, 6, 7)$$

Since there are a total of eight minterms or maxterms in a function of three variable, we determine the missing terms to be 0, 2, 4, and 5. The function expressed in product of maxterm is

$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

Truth Table for $F = xy + x'z$

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	1
1	1	1	1

Standard Forms

- The two canonical forms of Boolean algebra are basic forms that one obtains from reading a function from the truth table.
- These forms are very seldom the ones with the least number of literals, because each minterm or maxterm must contain, by definition, all the variables either complemented or uncomplemented.
- Another way to express Boolean functions is in standard form. In this configuration, the terms that form the function may contain one, two, or any number of literals.
- There are two types of standard forms: **the sum of products** and **product of sums**.

Sum of products

The sum ofproducts is a Boolean expression containing AND terms, called product terms, of one or more literals each.

The sum denotes the ORing of these terms. An example of a function expressed in sum of products is

$$F_1 = y' + xy + x'yz'$$

The expression has three product terms of one, two, and three literals each, respectively. Their sum is in effect an OR operation.

Product of sums

A product of sums is a Boolean expression containing OR terms, called sum terms.

Each term may have any number of literals. The product denotes the ANDing of these terms.

An example of a function expressed in product of sums is

$$F_2 = x(y' + z)(x' + y + z' + w)$$

This expression has three sum terms of one, two, and four literals each, respectively.

Note: The product is an AND operation.

The use of the words product and sum stems from the similarity of the AND operation to the arithmetic product (multiplication) and the similarity of the OR operation to the arithmetic sum (addition).

Non standard form

A Boolean function may be expressed in a nonstandard form. For example, the function

$$F_3 = (AB + CD)(A'B' + C'D')$$

is neither in sum of products nor in product of sums.

It can be changed to a standard form by using the distributive law to remove the parentheses:

$$F_3 = A'B'CD + ABC'D'$$

Example

Simplify the following Boolean functions to a minimum number of literals.

1. $x + x'y$

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

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

2. $x(x' + y)$

$= xx' + xy$

$= 0 + xy = xy$

3. $x'y'z + x'yz + xy'$

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

4. $xy + x'z + yz$

$= xy + x'z + yz(x + x')$
$= xy + x'z + xyz + x'yz$
$= xy(1 + z) + x'z(1 + y)$
$= xy + x'z$

THE MAP METHOD

The truth table representation of a function is unique, expressed algebraically, it can appear in many different forms.

Boolean functions may be simplified by algebraic means. This procedure of minimization is awkward because it lacks specific rules to predict each succeeding step in the manipulative process.

The map method provides a simple straightforward procedure for minimizing Boolean functions.

The map method, first proposed by Veitch and modified by Karnaugh, is also known as the “Veitch diagram’' or the “Karnaugh map.”

Description of K-maps and Terminology

A K-map is a matrix consisting of rows and columns that represent the output values of a Boolean function.

The output values placed in each cell are derived from the minterms of a Boolean function.

A *minterm* is a product term that contains all of the function’s variables exactly once, either complemented or not complemented.

Description of K-maps and Terminology

For example, the minterms for a function having the inputs *x* and *y* are: $\bar{x}\bar{y}$, $\bar{x}y$, $x\bar{y}$, and xy

Consider the Boolean function, $F(x,y) = xy + x\bar{y}$

Its minterms are:

Minterm	X	Y
$\bar{X}\bar{Y}$	0	0
$\bar{X}Y$	0	1
$X\bar{Y}$	1	0
XY	1	1

Description of K-maps and Terminology

Similarly, a function having three inputs, has the minterms that are shown in this diagram.

Minterm	X	Y	Z
$\overline{X}\overline{Y}\overline{Z}$	0	0	0
$\overline{X}\overline{Y}Z$	0	0	1
$\overline{X}Y\overline{Z}$	0	1	0
$\overline{X}YZ$	0	1	1
$X\overline{Y}\overline{Z}$	1	0	0
$X\overline{Y}Z$	1	0	1
$XY\overline{Z}$	1	1	0
XYZ	1	1	1

Description of K-maps and Terminology

A K-map has a cell for each minterm.

This means that it has a cell for each line for the truth table of a function.

The truth table for the function *F(x,y) = xy* is shown at the right along with its corresponding K-map.

F(X,Y) = XY

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X \ Y	0	1
0	0	0
1	0	1

Description of K-maps and Terminology

As another example, we give the truth table and K-Map for the function, *F(x,y)* = *x* + *y* at the right.

This function is equivalent to the OR of all of the minterms that have a value of 1. Thus:

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

F(X,Y) = X+Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

X \ Y	0	1
0	0	1
1	1	1

K-map Simplification for Two Variables

Of course, the minterm function that we derived from our K-map was not in simplest terms.

- That's what we started with in this example.

We can, however, reduce our complicated expression to its simplest terms by finding adjacent 1s in the K-map that can be collected into groups that are powers of two.

- In our example, we have two such groups.
    - Can you find them?

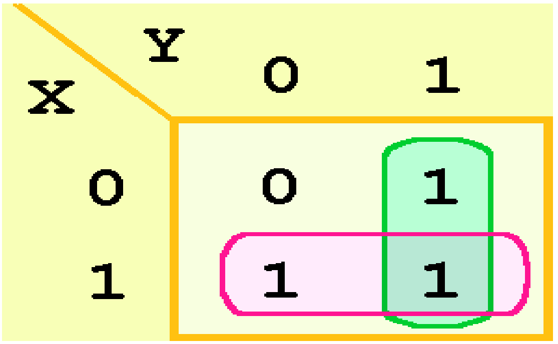
X \ Y	0	1
0	0	1
1	1	1

K-map Simplification for Two Variables

The best way of selecting two groups of 1's form our simple K-map is shown below.

We see that both groups are powers of two and that the groups overlap.

The next slide gives guidance for selecting Kmap groups.



K-map Simplification for Two Variables

The rules of Kmap simplification are:

- Groupings can contain only 1’s; no 0’s.
- Groups can be formed only at right angles; diagonal groups are not allowed.
- The number of 1’s in a group must be a power of 2 – even if it contains a single 1.
- The groups must be made as large as possible.
- Groups can overlap and wrap around the sides of the Kmap.

K-map Simplification for Three Variables

A Kmap for three variables is constructed as shown in the diagram below.

We have placed each minterm in the cell that will hold its value.

- Notice that the values for the *yz* combination at the top of the matrix form a pattern that is not a normal binary sequence.

x \ yz	00	01	11	10
0	$\overline{x}\,\overline{y}\,\overline{z}$	$\overline{x}\,\overline{y}z$	$\overline{x}yz$	$\overline{x}y\overline{z}$
1	$x\overline{y}\,\overline{z}$	$x\overline{y}z$	xyz	$xy\overline{z}$

K-map Simplification for Three Variables

Thus, the first row of the Kmap contains all minterms where *x* has a value of zero.

The first column contains all minterms where *y* and *z* both have a value of zero.

x \ yz	00	01	11	10
0	$\overline{x}\overline{y}\overline{z}$	$\overline{x}\overline{y}z$	$\overline{x}yz$	$\overline{x}y\overline{z}$
1	$x\overline{y}\overline{z}$	$x\overline{y}z$	xyz	$xy\overline{z}$

K-map Simplification for Three Variables

Consider the function:

$$F(X,Y) = \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}Z + XYZ$$

Its Kmap is given below.

- What is the largest group of 1s that is a power of 2?

X \ YZ	00	01	11	10
0	0	1	1	0
1	0	1	1	0

K-map Simplification for Three Variables

This grouping tells us that changes in the variables *x* and *y* have no influence upon the value of the function:

They are irrelevant. This means that the function.

$$F(X,Y) = \overline{X}\,\overline{Y}Z + \overline{X}YZ + X\overline{Y}Z + XYZ$$

reduces to *F(x) = z.*

You could verify this reduction with identities or a truth table.

X \ YZ	00	01	11	10
0	0	1	1	0
1	0	1	1	0

K-map Simplification for Three Variables

Now for a more complicated Kmap. Consider the function:

$$F(X,Y,Z) = \overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}\,\overline{Y}Z + \overline{X}YZ + \overline{X}Y\overline{Z} + X\overline{Y}\,\overline{Z} + XY\overline{Z}$$

Its Kmap is shown below. There are (only) two groupings of 1s.

- Can you find them?

X \ YZ	00	01	11	10
0	1	1	1	1
1	1	0	0	1

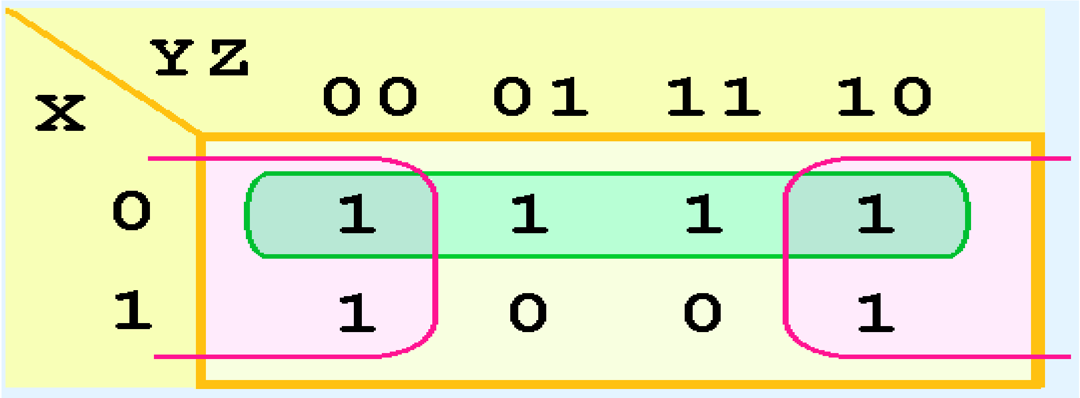
K-map Simplification for Three Variables

In this Kmap, we see an example of a group that wraps around the sides of a Kmap.

This group tells us that the values of *x* and *y* are not relevant to the term of the function that is encompassed by the group.

- What does this tell us about this term of the function?

What about the green group in the top row?



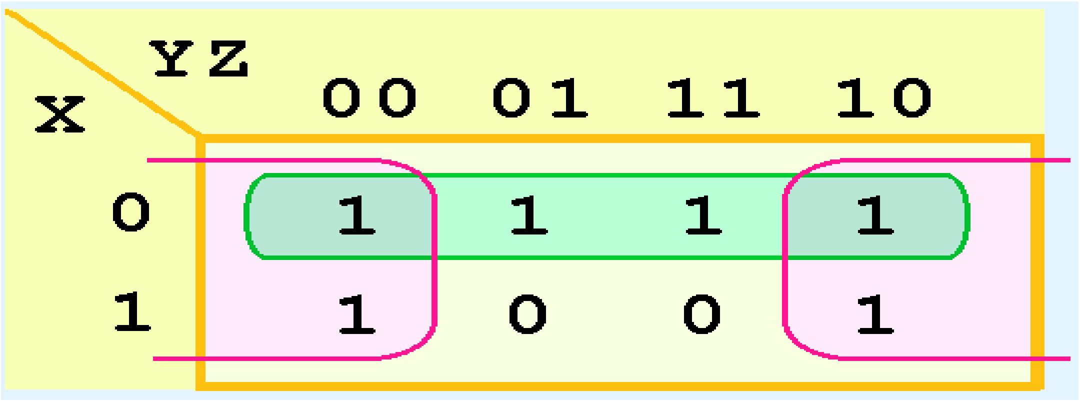
K-map Simplification for Three Variables

The green group in the top row tells us that only the value of *x* is significant in that group.

We see that it is complemented in that row, so the other term of the reduced function is $\overline{X}$.

Our reduced function is: $F(X,Y,Z) = \overline{X} + \overline{Z}$

Recall that we had six minterms in our original function!



X \ YZ	00	01	11	10
0	1	1	1	1
1	1	0	0	1

K-map Simplification for Four Variables

Our model can be extended to accommodate the 16 minterms that are produced by a four-input function.

This is the format for a 16-minterm Kmap.

WX \ YZ	00	01	11	10
00	$\bar{W}\bar{X}\bar{Y}\bar{Z}$	$\bar{W}\bar{X}\bar{Y}Z$	$\bar{W}\bar{X}YZ$	$\bar{W}\bar{X}Y\bar{Z}$
01	$\bar{W}X\bar{Y}\bar{Z}$	$\bar{W}X\bar{Y}Z$	$\bar{W}XYZ$	$\bar{W}XY\bar{Z}$
11	$WX\bar{Y}\bar{Z}$	$WX\bar{Y}Z$	$WXYZ$	$WXY\bar{Z}$
10	$W\bar{X}\bar{Y}\bar{Z}$	$W\bar{X}\bar{Y}Z$	$W\bar{X}YZ$	$W\bar{X}Y\bar{Z}$

K-map Simplification for Four Variables

We have populated the Kmap shown below with the nonzero minterms from the function:

$$F(W,X,Y,Z) = \overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z} + \overline{W}\,\overline{X}\,\overline{Y}Z + \overline{W}\,\overline{X}Y\overline{Z} + \overline{W}XY\overline{Z} + W\overline{X}\,\overline{Y}\,\overline{Z} + W\overline{X}\,\overline{Y}Z + W\overline{X}Y\overline{Z}$$

- Can you identify (only) three groups in this Kmap?

Recall that groups can overlap.

WX \ YZ	00	01	11	10
00	1	1		1
01				1
11				
10	1	1		1

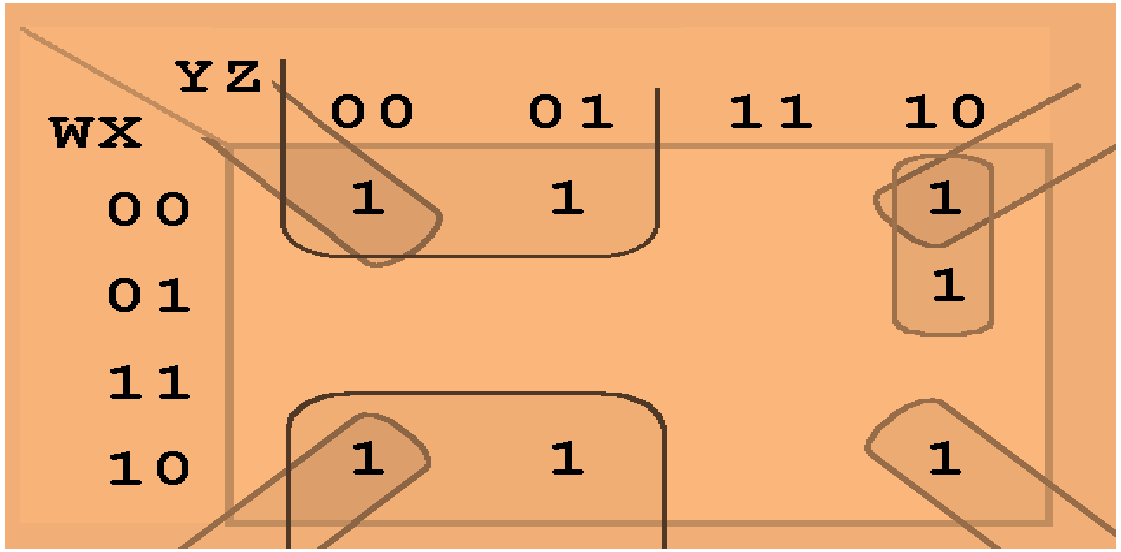
K-map Simplification for Four Variables

Our three groups consist of:

- A purple group entirely within the Kmap at the right.
- A pink group that wraps the top and bottom.
- A green group that spans the corners.

Thus we have three terms in our final function:

$$F(W,X,Y,Z) = \bar{X}\bar{Y} + \bar{X}\bar{Z} + \bar{W}Y\bar{Z}$$

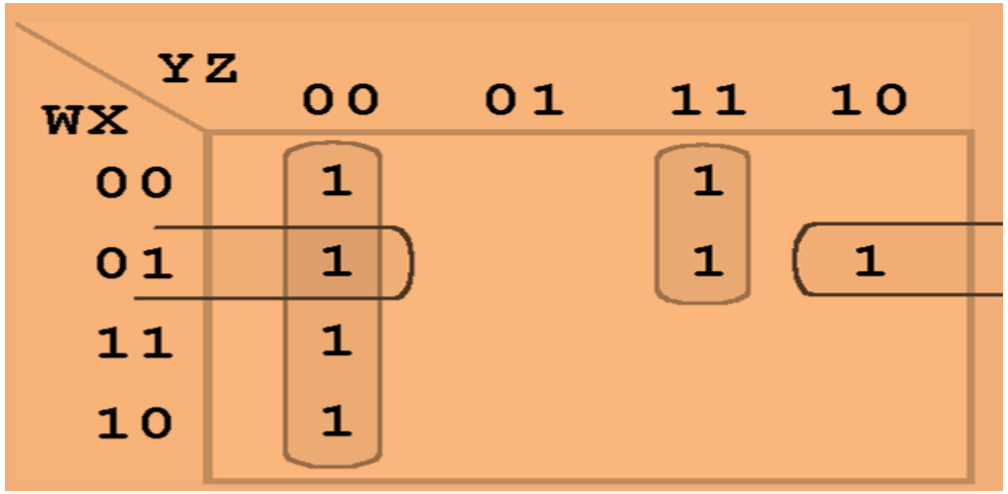


K-map Simplification for Four Variables

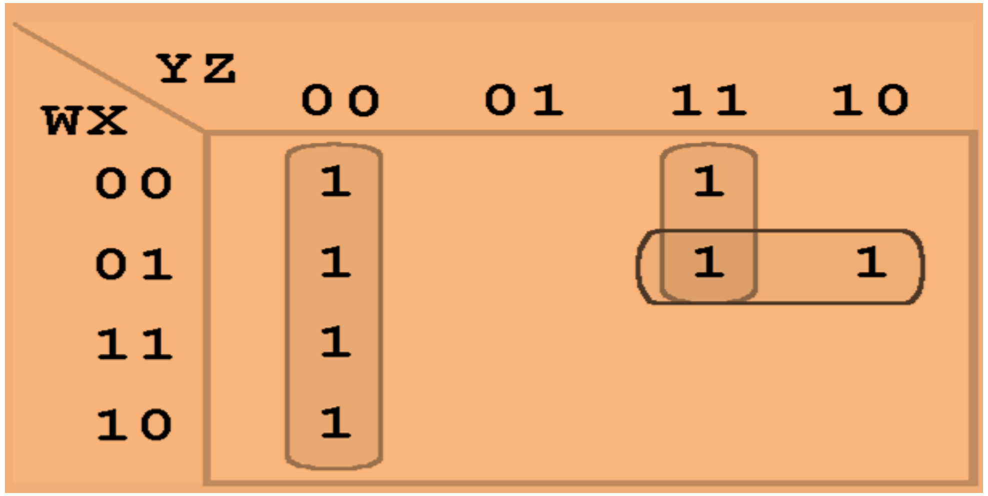
It is possible to have a choice as to how to pick groups within a Kmap, while keeping the groups as large as possible.

The (different) functions that result from the groupings below are logically equivalent.

y’z’ +yz++w’xz’



y’z’+yz+w’xy



Suggested Reading

- M. Morris Mano, Digital Logic and Computer Design, PHI.

Thank you