
PGT104 – Digital Electronics

Part 3 – Boolean Algebra

Disclaimer:

- Most of the contents (if not all) are extracted from resources available for Digital Fundamentals 10th Edition

Boolean Addition

In Boolean algebra, a **variable** is a symbol used to represent an action, a condition, or data. A single variable can only have a value of 1 or 0.

The **complement** represents the inverse of a variable and is indicated with an overbar. Thus, the complement of A is \bar{A} .

A **literal** is a variable or its complement.

Addition is equivalent to the OR operation. The sum term is 1 if one or more of the literals are 1. The sum term is zero only if each literal is 0.



Example

Determine the values of A , B , and C that make the sum term of the expression $A + B + C = 0$?

Each literal must = 0; therefore $A = 1$, $B = 0$ and $C = 1$.

Boolean Multiplication

In Boolean algebra, multiplication is equivalent to the AND operation. The product of literals forms a product term. The product term will be 1 only if all of the literals are 1.

Example

What are the values of the A , B and C if the product term of $A \cdot \overline{B} \cdot \overline{C} = 1$?

Each literal must = 1; therefore $A = 1$, $B = 0$ and $C = 0$.

Commutative Laws

The **commutative laws** are applied to addition and multiplication. For addition, the commutative law states
In terms of the result, the order in which variables are ORed makes no difference.

$$A + B = B + A$$

For multiplication, the commutative law states
In terms of the result, the order in which variables are ANDed makes no difference.

$$AB = BA$$

Associative Laws

The **associative laws** are also applied to addition and multiplication. For addition, the associative law states

When ORing more than two variables, the result is the same regardless of the grouping of the variables.

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

For multiplication, the associative law states

When ANDing more than two variables, the result is the same regardless of the grouping of the variables.

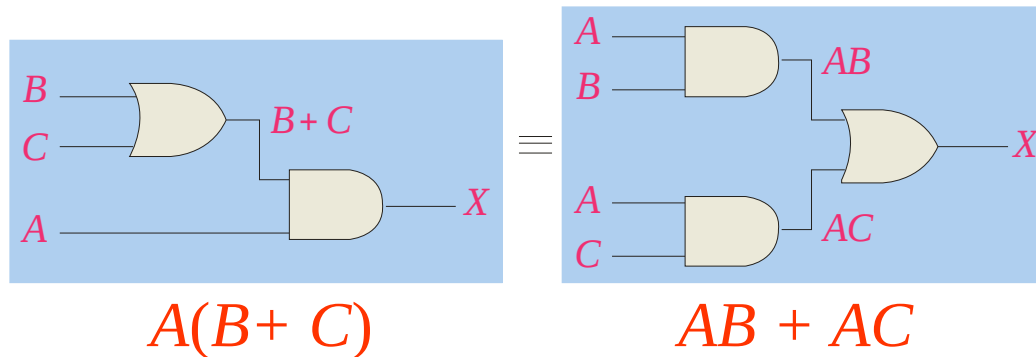
$$A(BC) = (AB)C$$

Distributive Law

The **distributive law** is the factoring law. A common variable can be factored from an expression just as in ordinary algebra. That is

$$AB + AC = A(B + C)$$

The distributive law can be illustrated with equivalent circuits:



Rules of Boolean Algebra

1. $A + 0 = A$

2. $A + 1 = 1$

3. $A \cdot 0 = 0$

4. $A \cdot 1 = A$

5. $A + A = A$

6. $A + \bar{A} = 1$

7. $A \cdot A = A$

8. $A \cdot \bar{A} = 0$

9. $\bar{\bar{A}} = A$

10. $A + AB = A$

11. $A + \bar{A}B = A + B$

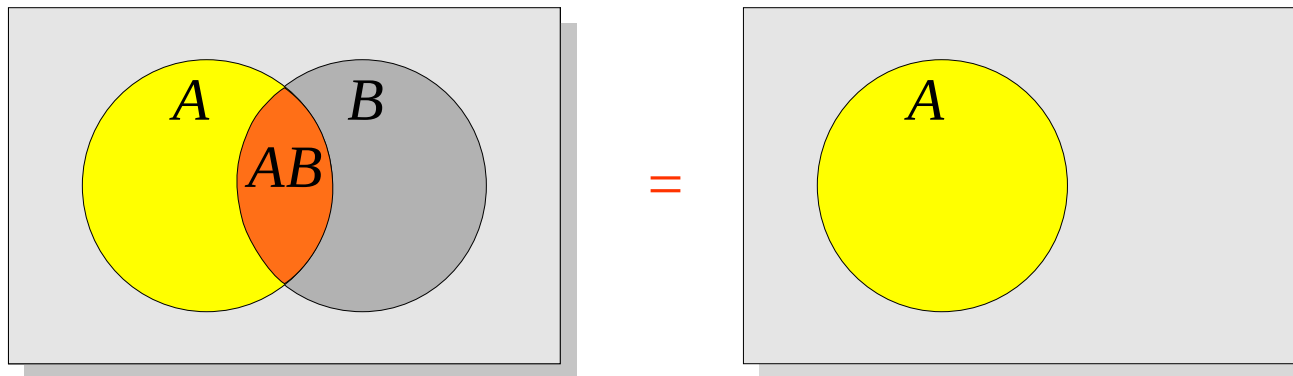
12. $(A + B)(A + C) = A + BC$

Rules of Boolean Algebra

Rules of Boolean algebra can be illustrated with *Venn* diagrams. The variable A is shown as an area.

The rule $A + AB = A$ can be illustrated easily with a diagram. Add an overlapping area to represent the variable B .

The overlap region between A and B represents AB .



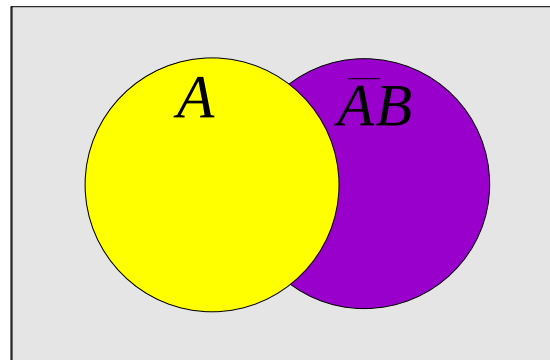
The diagram visually shows that $A + AB = A$. Other rules can be illustrated with the diagrams as well.

Rules of Boolean Algebra

Example

Illustrate the rule $A + \bar{A}B = A + B$ with a Venn diagram.

This time, \bar{A} is represented by the blue area and B again by the red circle. The intersection represents $\bar{A}B$. Notice that $A + \bar{A}B = A + B$



Rules of Boolean Algebra

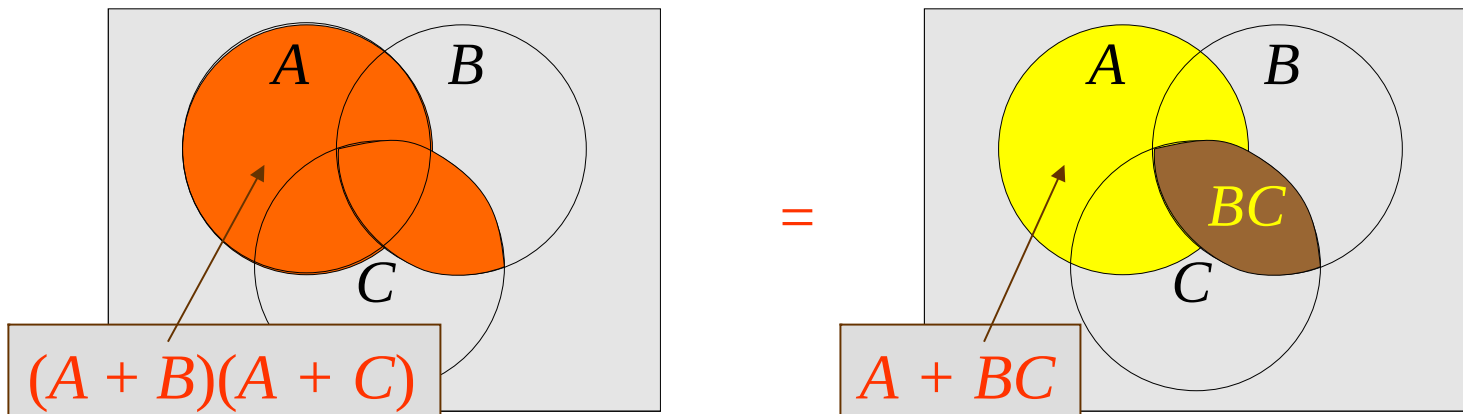
Rule 12, which states that $(A + B)(A + C) = A + BC$, can be proven by applying earlier rules as follows:

$$\begin{aligned}(A + B)(A + C) &= AA + AC + AB + BC \\ &= A + AC + AB + BC \\ &= A(1 + C + B) + BC \\ &= A \cdot 1 + BC \\ &= A + BC\end{aligned}$$

This rule is a little more complicated, but it can also be shown with a Venn diagram, as given on the following slide...

Three areas represent the variables A , B , and C .
The area representing $A + B$ is shown in yellow.
The area representing $A + C$ is shown in red.
The overlap of red and yellow is shown in orange.

The overlapping area between B and C represents BC .
ORing with A gives the same area as before.



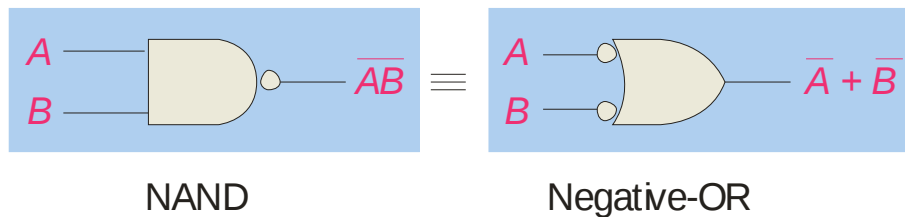
DeMorgan's Theorem

DeMorgan's 1st Theorem

The complement of a product of variables is equal to the sum of the complemented variables.

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

Applying DeMorgan's first theorem to gates:



Inputs		Output	
A	B	\overline{AB}	$\overline{A} + \overline{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

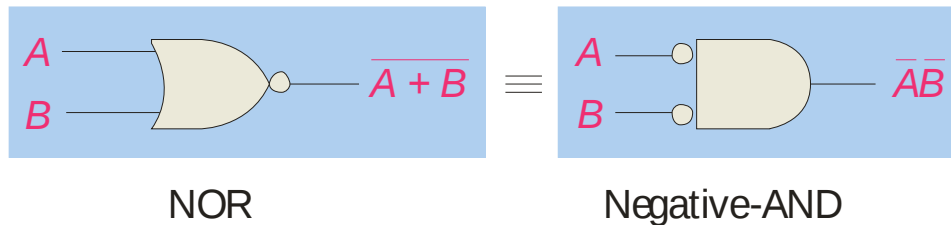
DeMorgan's Theorem

DeMorgan's 2nd Theorem

The complement of a sum of variables is equal to the product of the complemented variables.

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

Applying DeMorgan's second theorem to gates:



DeMorgan's Theorem



Example

Apply DeMorgan's theorem to remove the overbar covering both terms from the expression $X = \overline{C + D}$.

To apply DeMorgan's theorem to the expression, you can break the overbar covering both terms and change the sign between the terms. This results in $X = \overline{\overline{C}} \cdot \overline{\overline{D}}$. Deleting the double bar gives $X = C \cdot \overline{D}$.

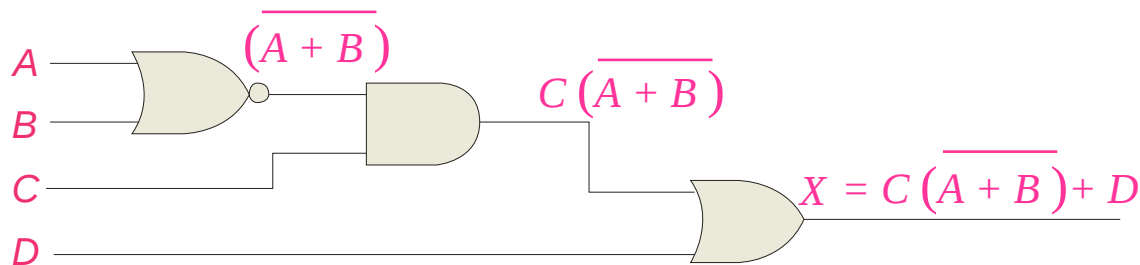
Boolean Analysis of Logic Circuits

Combinational logic circuits can be analyzed by writing the expression for each gate and combining the expressions according to the rules for Boolean algebra.

Example

Apply Boolean algebra to derive the expression for X .

Write the expression for each gate:



Applying DeMorgan's theorem and the distribution law:

$$X = C \overline{(A + B)} + D = \overline{A} \overline{B} C + D$$

SOP and POS forms

Boolean expressions can be written in the **sum-of-products** form (**SOP**) or in the **product-of-sums** form (**POS**). These forms can simplify the implementation of combinational logic, particularly with PLDs. In both forms, an overbar cannot extend over more than one variable.

An expression is in SOP form when two or more product terms are summed as in the following examples:

$$\overline{A} \overline{B} \overline{C} + A B$$

$$A B \overline{C} + \overline{C} \overline{D}$$

$$C D + \overline{E}$$

An expression is in POS form when two or more sum terms are multiplied as in the following examples:

$$(A + B)(\overline{A} + C)$$

$$(A + B + \overline{C})(B + D)$$

$$(\overline{A} + B)C$$

SOP Standard form

In **SOP standard form**, every variable in the domain must appear in each term. This form is useful for constructing truth tables or for implementing logic in PLDs.

You can expand a nonstandard term to standard form by multiplying the term by a term consisting of the sum of the missing variable and its complement.



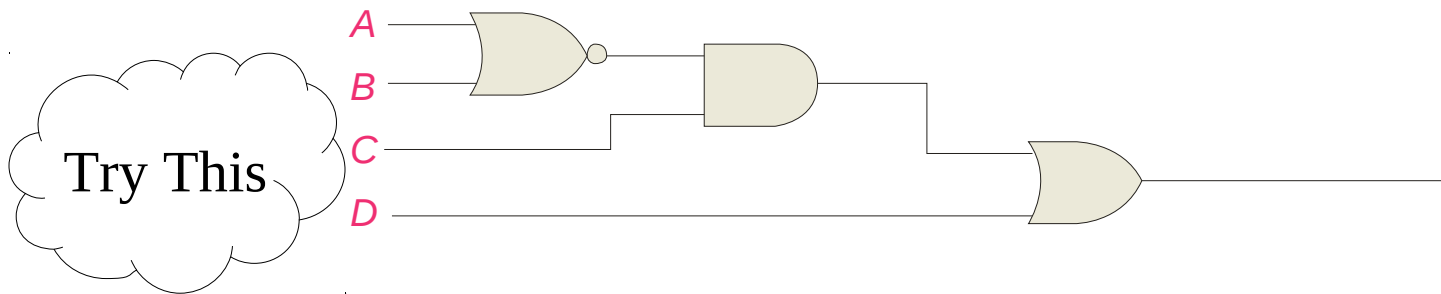
Example

Convert $X = \bar{A}\bar{B} + ABC$ to standard form.

The first term does not include the variable C . Therefore, multiply it by the $(C + \bar{C})$, which = 1:

$$\begin{aligned} X &= \bar{A}\bar{B}(C + \bar{C}) + ABC \\ &= \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + ABC \end{aligned}$$

SOP Standard form



Convert the circuit above into standard SOP form.

POS Standard form

In **POS standard form**, every variable in the domain must appear in each sum term of the expression.

You can expand a nonstandard POS expression to standard form by adding the product of the missing variable and its complement and applying rule 12, which states that $(A + B)(A + C) = A + BC$.

Convert $X = (\bar{A} + \bar{B})(A + B + C)$ to standard form.



Example

The first sum term does not include the variable C . Therefore, add $C\bar{C}$ and expand the result by rule 12.

$$\begin{aligned} X &= (\bar{A} + \bar{B} + C\bar{C})(A + B + C) \\ &= (\bar{A} + \bar{B} + C)(\bar{A} + \bar{B} + \bar{C})(A + B + C) \end{aligned}$$

Minterms

- Associated with SOP
- For a 2-input XOR:
 - truth table row with 1's
 - $F_{XOR}(A,B) = m_1 + m_2$

- A compact notation:

$$F_{XOR}(A,B) = \sum (1,2)$$

- Notice the small letter m is used

A	B	minter m
0	0	$m_0 = \bar{A} \bar{B}$
0	1	$m_1 = \bar{A} B$
1	0	$m_2 = A \bar{B}$
1	1	$m_3 = AB$

Maxterms

- Associated with SOP
- For a 2-input XOR:
 - truth table row with 0's
 - $F_{XOR}(A,B) = M_0 \cdot M_3$
- A compact notation:

$$F_{XOR}(A,B) = \prod (0,3)$$
- Notice the capital letter M

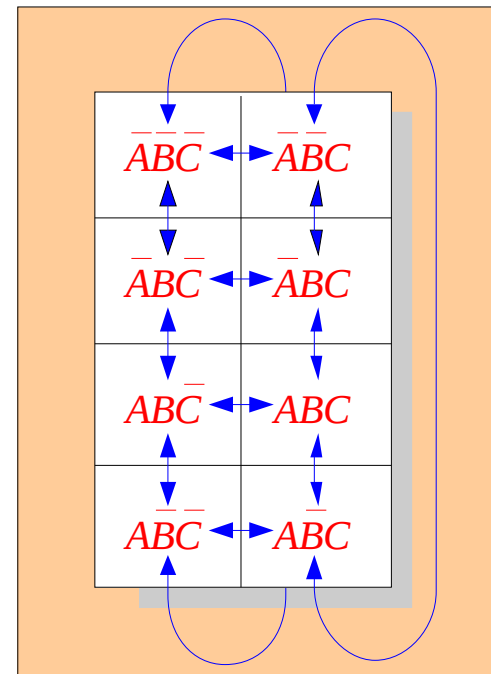
A	B	maxterm
0	0	$M_0 = \bar{A} + \bar{B}$
0	1	$M_1 = \bar{A} + B$
1	0	$M_2 = A + \bar{B}$
1	1	$M_3 = A + B$

Karnaugh maps

The Karnaugh map (K-map) is a tool for simplifying combinational logic with 3 or 4 variables. For 3 variables, 8 cells are required (2^3).

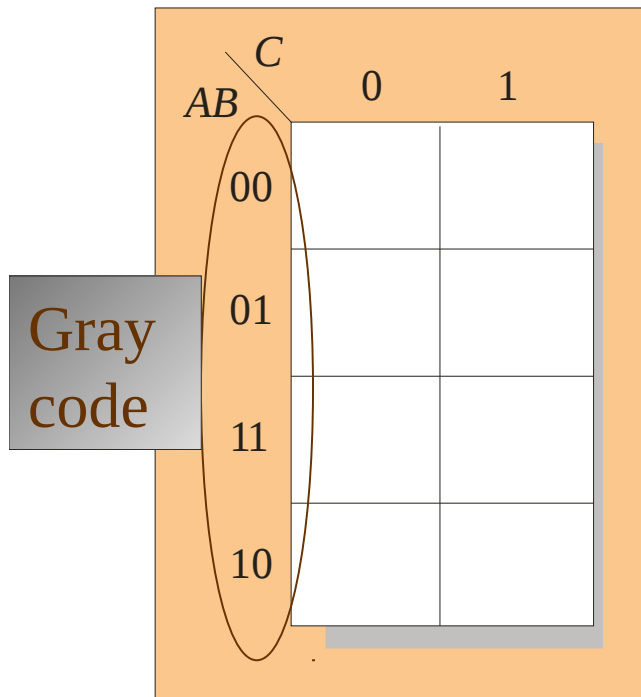
The map shown is for three variables labeled A , B , and C . Each cell represents one possible product term.

Each cell differs from an adjacent cell by only one variable.



Karnaugh maps

Cells are usually labeled using 0's and 1's to represent the variable and its complement.



The numbers are entered in gray code, to force adjacent cells to be different by only one variable.

Ones are read as the true variable and zeros are read as the complemented variable.

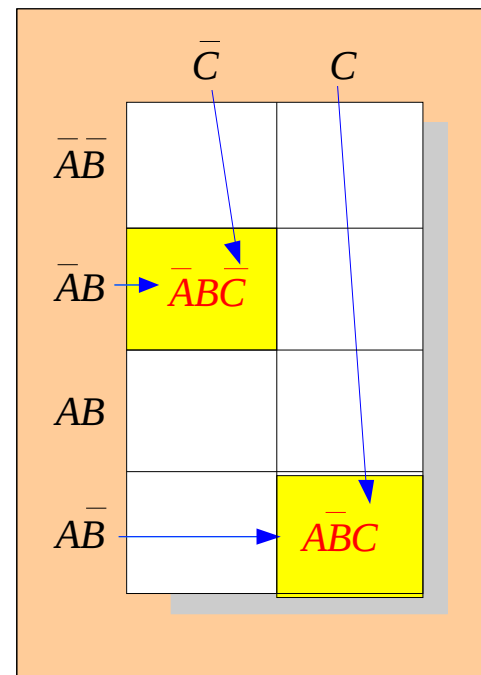
Karnaugh maps

Alternatively, cells can be labeled with the variable letters. This makes it simple to read, but it takes more time preparing the map.

Example

Read the terms for the yellow cells.

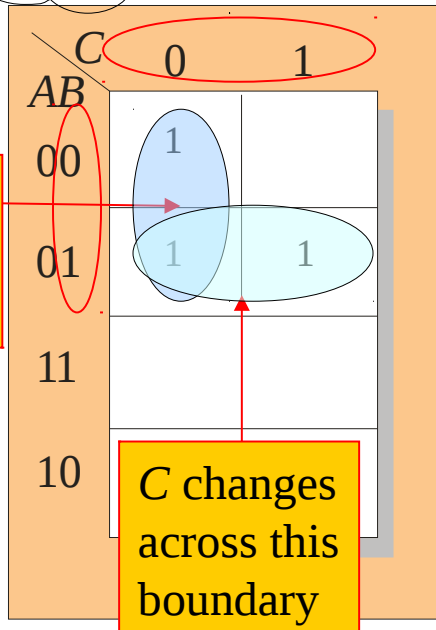
The cells are $\overline{A}B\overline{C}$ and $A\overline{B}C$.



Karnaugh maps

K-maps can simplify combinational logic by grouping cells and eliminating variables that change.

Example Group the 1's on the map and read the minimum logic.

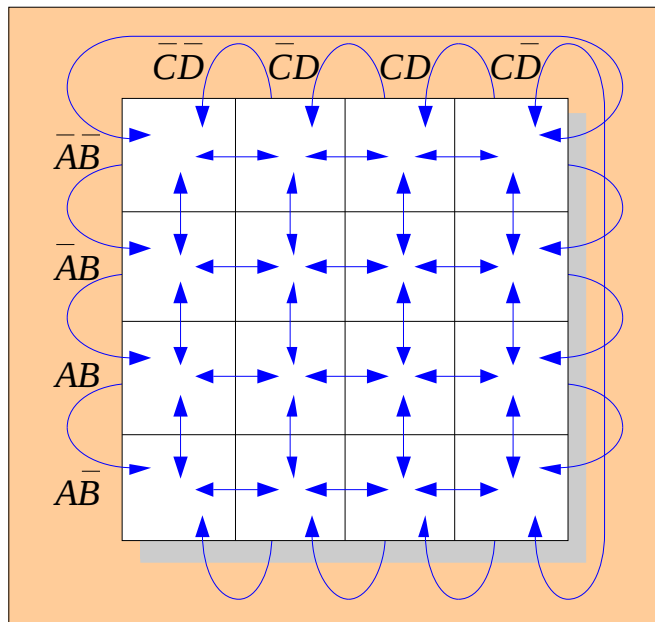


1. Group the 1's into two overlapping groups as indicated.
2. Read each group by eliminating any variable that changes across a boundary.
3. The vertical group is read $\overline{A}\overline{C}$.
4. The horizontal group is read $\overline{A}B$.

$$X = \overline{A}\overline{C} + \overline{A}B$$

Karnaugh maps

A 4-variable map has an adjacent cell on each of its four boundaries as shown.



Each cell is different only by one variable from an adjacent cell.

Grouping follows the rules given in the text.

The following slide shows an example of reading a four variable map using binary numbers for the variables...

Karnaugh maps

Example

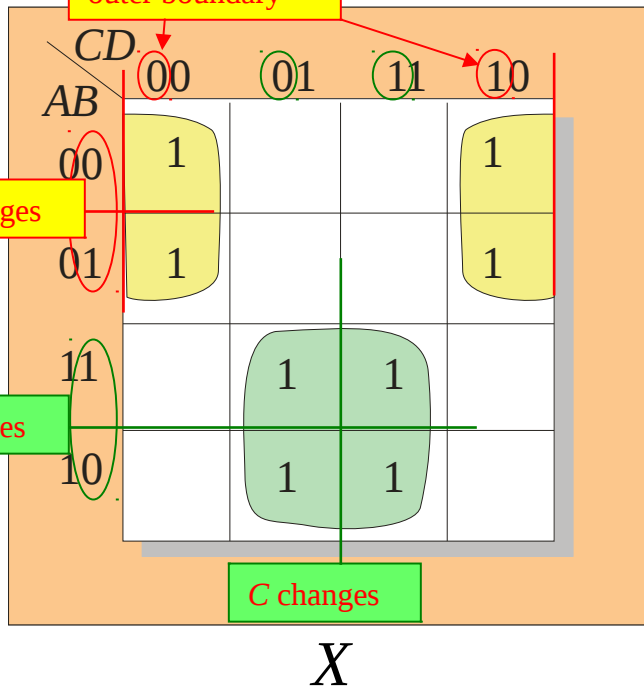
Group the 1's on the map and read the minimum logic.

C changes across outer boundary

B changes

B changes

C changes



1. Group the 1's into two separate groups as indicated.
2. Read each group by eliminating any variable that changes across a boundary.
3. The upper (yellow) group is read as $\overline{A}\overline{D}$.
4. The lower (green) group is read as AD .

$$X = \overline{A}\overline{D} + AD$$

Quiz

The associative law for addition is normally written as

a. $A + B = B + A$

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

c. $AB = BA$

d. $A + AB = A$

Quiz

The Boolean equation $AB + AC = A(B + C)$ illustrates

- a. the distribution law
- b. the commutative law
- c. the associative law
- d. DeMorgan's theorem

Quiz

The Boolean expression $A \cdot 1$ is equal to

- a. A
- b. B
- c. 0
- d. 1

Quiz

The Boolean expression $A + 1$ is equal to

- a. A
- b. B
- c. 0
- d. 1

Quiz

The Boolean equation $AB + AC = A(B + C)$ illustrates

- a. the distribution law
- b. the commutative law
- c. the associative law
- d. DeMorgan's theorem

Quiz

A Boolean expression that is in standard SOP form is

- a. the minimum logic expression
- b. contains only one product term
- c. has every variable in the domain in every term
- d. none of the above

Quiz

Adjacent cells on a Karnaugh map differ from each other by

- a. one variable
- b. two variables
- c. three variables
- d. answer depends on the size of the map

Quiz

The minimum expression that can be read from the Karnaugh map shown is

- a. $X = A$
- b. $X = \bar{A}$
- c. $X = B$
- d. $X = \bar{B}$

	\bar{C}	C
$\bar{A}\bar{B}$		
$\bar{A}B$		
AB	1	1
$A\bar{B}$	1	1

Quiz

The minimum expression that can be read from the Karnaugh map shown is

a. $X = A$

b. $X = \bar{A}$

c. $X = B$

d. $X = \bar{B}$

	\bar{C}	C
$\bar{A}\bar{B}$	1	1
$\bar{A}B$		
AB		
$A\bar{B}$	1	1