XMUT 202
Digital Electronics

Felix Yan

School of Engineering and Computer Science
Victoria University of Wellington

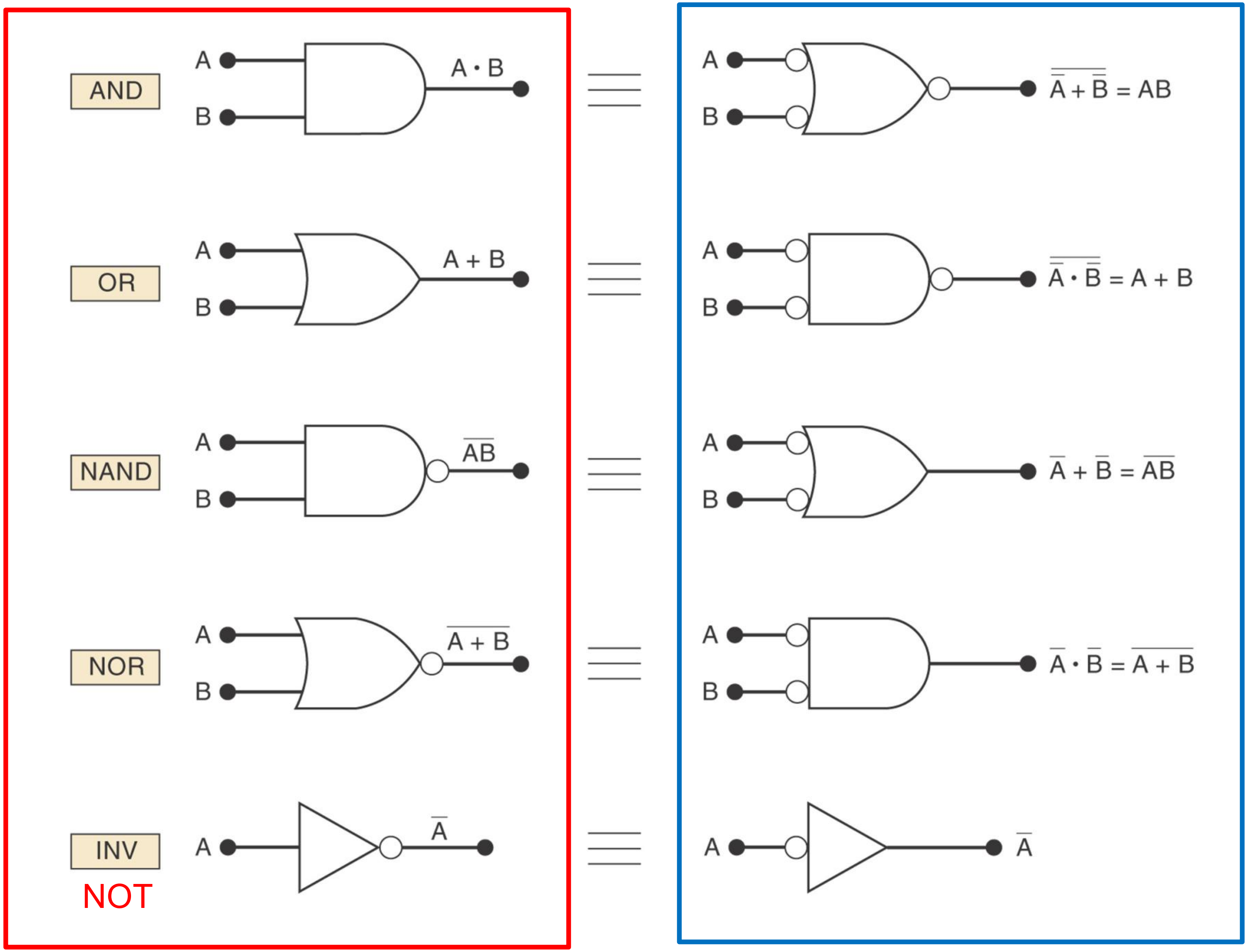
Lecture Content

- Alternate Logic Representation
- Combinatorial Logic

Alternate Logic-Gate Representations

- To convert a standard symbol to an alternate:
  - Invert each input and output (add an inversion bubble where there are none on the standard symbol, and remove bubbles where they exist on the standard symbol.
  - Change a standard OR gate to an AND gate, or an AND gate to an OR gate.

Standard and alternate symbols for various logic gates and inverter.



Alternate Logic-Gate Representations

- Active high – an input or output has no inversion bubble.
- Active low – an input or output has an inversion bubble.

Alternate Logic-Gate Representations

- Active high – an input or output has no inversion bubble.
- Active low – an input or output has an inversion bubble.
- An AND gate will produce an active output when all inputs are in their active states.
- An OR gate will produce an active output when any input is in an active state.

Which Gate Representation to Use?

- Using alternate and standard logic gate symbols together can make circuit operation clearer.
- When possible choose gate symbols so that bubble outputs are connected to bubble input and non-bubble outputs are connected to non-bubble inputs.

IEEE/ANSI standard Logic Symbols

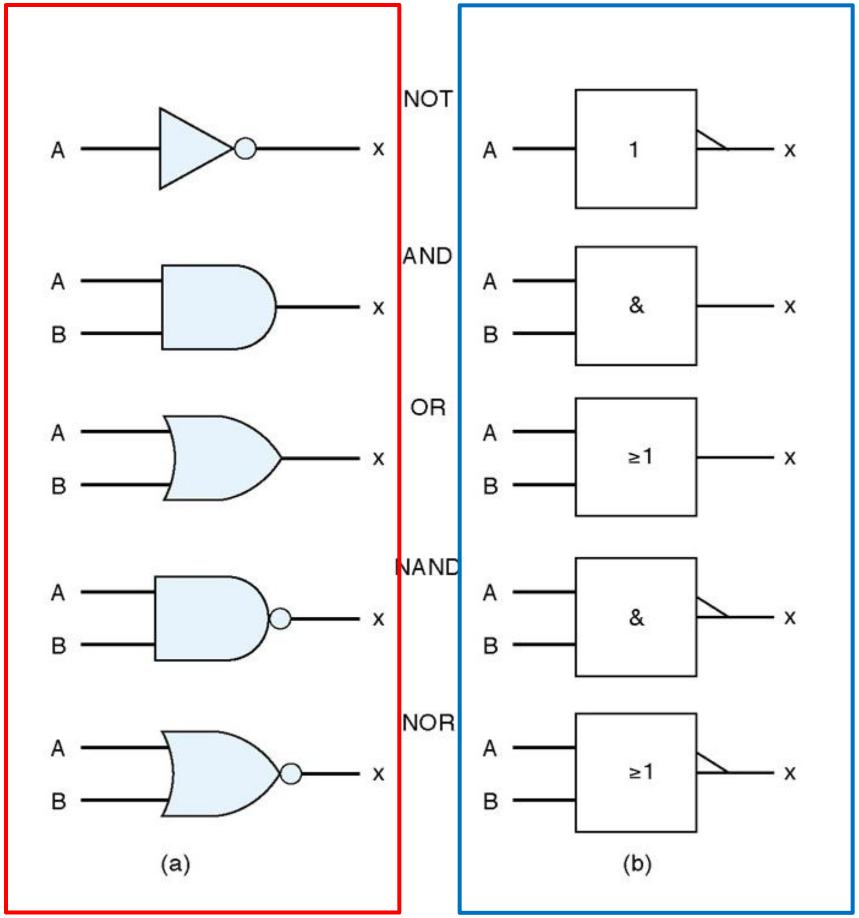
- Rectangular symbols represent logic gates and circuits.
- Dependency notation inside symbols show how output depends on inputs.
- A small triangle replaces the inversion bubble.

IEEE/ANSI standard Logic Symbols

- Compare the IEEE/ANSI symbols to traditional symbols.
- These symbols are not widely accepted but may appear in some schematics.

Lecture Notes

XMUT students' textbook



Combinatorial Logic

- Basic logic gate functions will be combined.
- At any time the output depends only on the combination of logic levels at the inputs to the circuit.
- Simplification of logic circuits will be done using Boolean algebra and a mapping technique.

Simplifying logic circuits

- Logic circuit simplification and design requires the logic expression to be in a sum-of-products (SOP) form.

- This expression will appear as two or more AND terms ORed together.

Examples: 1) $ABC + \overline{A}B\overline{C}$

2) $AB + \overline{A}B\overline{C} + \overline{C}\,\overline{D} + D$

Simplifying Logic Circuits

Two methods to simplify logic circuits:

(1) Boolean algebra

(2) Karnaugh mapping

Algebraic Simplification

Steps:

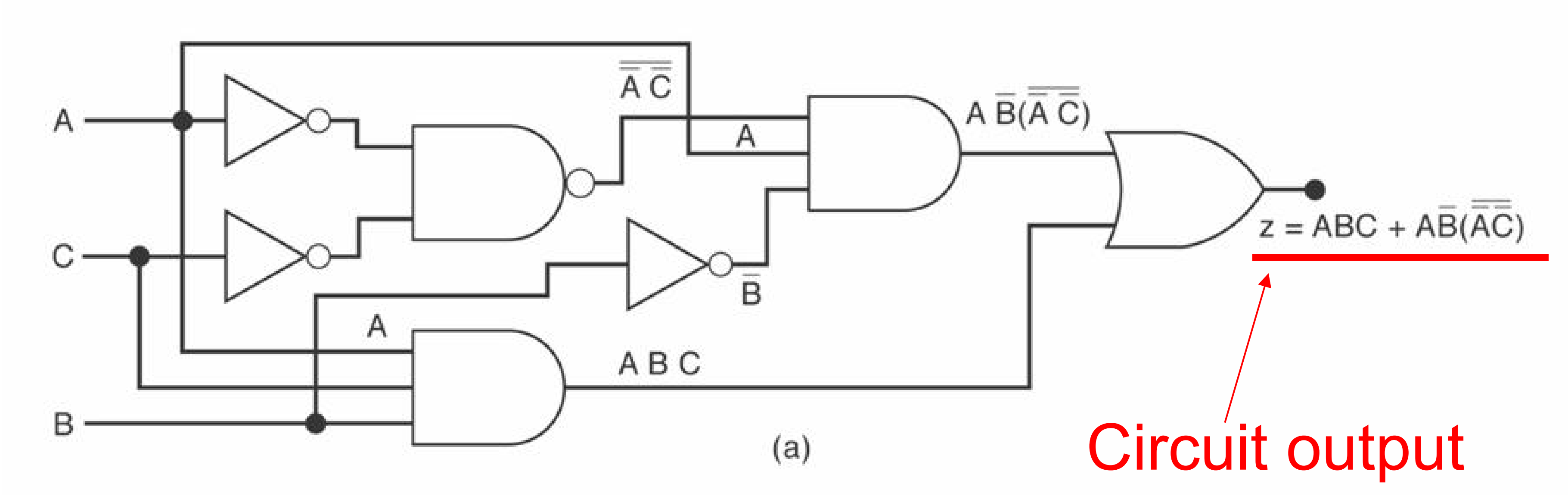
1. Place the expression in Sum-of-Products (SOP) form by applying DeMorgan's theorems and multiplying terms.
2. Check the SOP form for common factors and perform factoring where possible.

Note that this process may involve some trial and error to obtain the simplest result. Simplification gives a more efficient implementation.

De Morgan's Theorem

$$\overline{(x + y)} = \overline{x}.\overline{y} \qquad \overline{(x.y)} = \overline{x} + \overline{y}$$

Simplify the following logic circuit.



Simplification: $Z = ABC + A\overline{B}.(\overline{\overline{A}\,\overline{C}})$

Simplification: $Z = ABC + A\overline{B}.(\overline{\overline{A}\,\overline{C}})$

$$= ABC + A\overline{B}.(\overline{\overline{A}} + \overline{\overline{C}})$$

Steps:

1. Place the expression in Sum-of-Products (SOP) form by applying De Morgan's theorems and multiplying terms.

De Morgan's Theorems

$$\overline{(x + y)} = \overline{x}.\overline{y}$$

$$\overline{(x.y)} = \overline{x} + \overline{y}$$

Simplification:

$$Z = ABC + A\overline{B}.(\overline{\overline{A}\,\overline{C}})$$

$$= ABC + A\overline{B}.(\overline{\overline{A}} + \overline{\overline{C}})$$

$$= ABC + A\overline{B}.(A + C)$$

Simplification: $Z = ABC + A\bar{B}.(\overline{\bar{A}\bar{C}})$

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

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

$$= ABC + \boxed{A\bar{B}A + A\bar{B}C}$$

Simplification: $Z = ABC + A\overline{B}.(\overline{\overline{A}\,\overline{C}})$

$$= ABC + A\overline{B}.(\overline{\overline{A}} + \overline{\overline{C}})$$

$$= ABC + A\overline{B}.(A + C)$$

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

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

$$= AC(\overline{B} + B) + A\overline{B}$$

Simplification: $Z = ABC + A\overline{B}.(\overline{\overline{A}\,\overline{C}})$

$$= ABC + A\overline{B}.(\overline{\overline{A}} + \overline{\overline{C}})$$

$$= ABC + A\overline{B}.(A + C)$$

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

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

$$= AC(\overline{B} + B) + A\overline{B}$$

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

Simplification: $Z = ABC + A\bar{B}.(\overline{\bar{A}\bar{C}})$

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

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

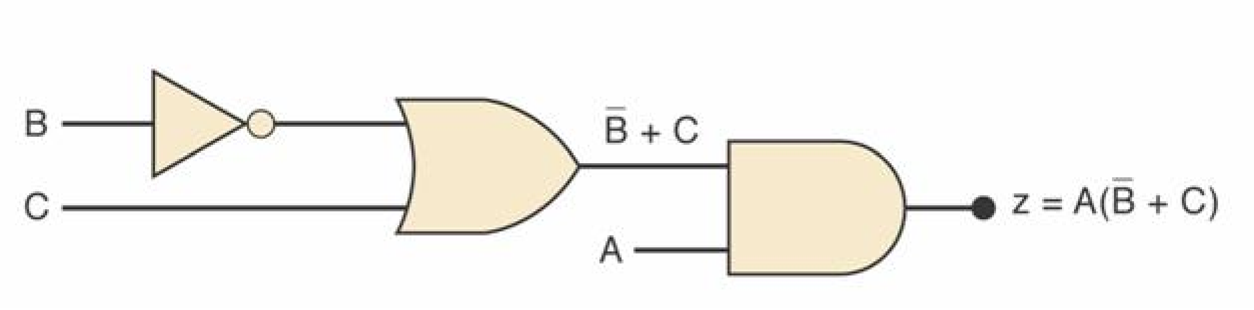
$$= ABC + A\bar{B}A + A\bar{B}C$$

$$= ABC + A\bar{B} + A\bar{B}C$$

$$= AC(\bar{B} + B) + A\bar{B}$$

$$= AC + A\bar{B}$$

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

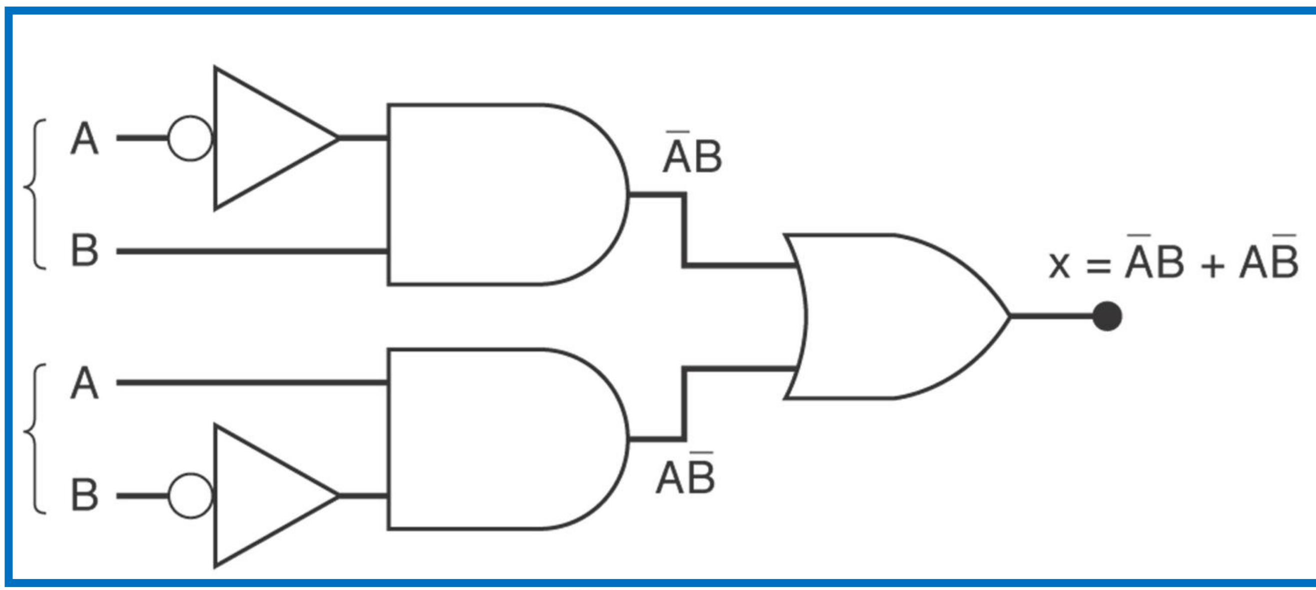


Designing Combinatorial Logic Circuits

If we know the design conditions (truth table) we want to design the logic circuit and then implement the circuit with AND, OR and NOT gates.

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

(a)



(b)

Designing Combinational Logic Circuits

Design Procedure:

1. Set up truth table
2. Write AND term for each case where the output is HI
3. Write the SOP expression for the output
4. Simplify the expression
5. Implement the circuit

Designing Combinational Logic Circuits

- Design a logic circuit that has three inputs, *A*, *B*, and *C*, whose output will be HIGH only when a majority of the inputs are HIGH.

Step 1: Set up truth table

3 Inputs: A, B and C

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

Step 1: Set up truth table

3 Inputs: A, B and C

A	B	C	x
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

Output will be HIGH only when a majority of inputs are HIGH

Step 1: Set up truth table

Step 2: Write AND term for each case where the output is HI

A	B	C	x	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\rightarrow \bar{A}BC$
1	0	0	0	
1	0	1	1	$\rightarrow A\bar{B}C$
1	1	0	1	$\rightarrow AB\bar{C}$
1	1	1	1	$\rightarrow ABC$

Step 3: Write the SOP expression for the output

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Step 3: Write the SOP expression for the output

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Step 4: Simplify the expression

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Step 3: Write the SOP expression for the output

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Step 4: Simplify the expression (trick: copy product terms!)

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

$$= \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$

Step 3: Write the SOP expression for the output

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Step 4: Simplify the expression

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

$$= \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$

$$= BC(\overline{A} + A) + AC(\overline{B} + B) + AB(\overline{C} + C)$$

Step 3: Write the SOP expression for the output

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Step 4: Simplify the expression

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

$$= \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$

$$= BC(\overline{A} + A) + AC(\overline{B} + B) + AB(\overline{C} + C)$$

$$= BC + AC + AB$$

Step 3: Write the SOP expression for the output

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Step 4: Simplify the expression

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

$$= \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$

$$= BC(\overline{A} + A) + AC(\overline{B} + B) + AB(\overline{C} + C)$$

$$= BC + AC + AB$$

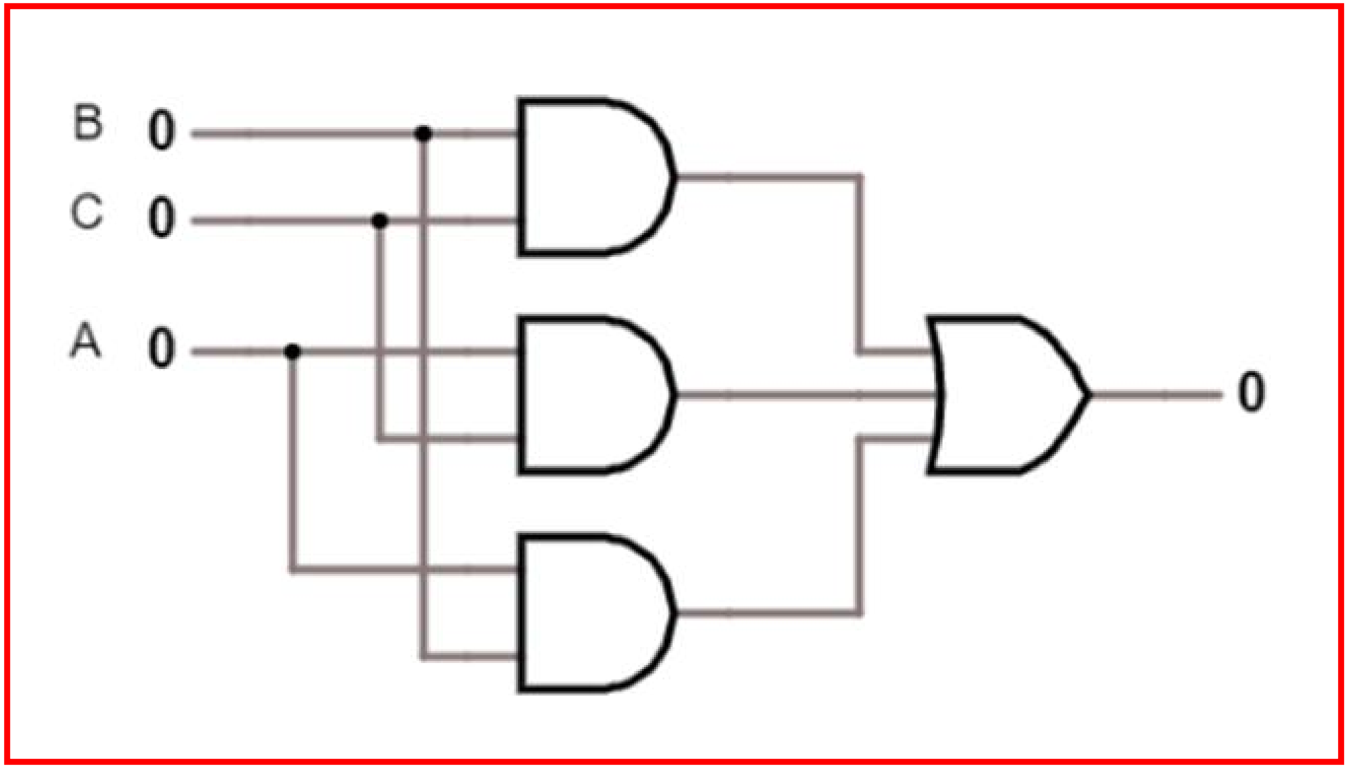
Step 5: Implement Circuit

You have **5 minutes** to implement (ie **draw**) the circuit for

$$x = BC + AC + AB$$

Step 5: Implement Circuit

x = BC + AC + AB



Exercises

1. Use Boolean Algebra to simplify the following:

a) $AB + A(B + C) + B(B + C)$

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

c) $[A\overline{B}(C + BD) + \overline{A}\,\overline{B}]C$

d) $\overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + ABC$

2. Draw the logic circuits for the Boolean expression shown above and the simplified circuit derived from your solutions.

Boolean Algebra - Basic Rules

1. $A + 0 = A$
2. $A + 1 = 1$
3. $A \bullet 0 = 0$
4. $A \bullet 1 = A$
5. $A + A = A$
6. $A + \overline{A} = 1$

Boolean Algebra - Basic Rules

1. $A + 0 = A$
2. $A + 1 = 1$
3. $A \bullet 0 = 0$
4. $A \bullet 1 = A$
5. $A + A = A$
6. $A + \overline{A} = 1$
7. $A \bullet A = A$
8. $A \bullet \overline{A} = 0$
9. $\overline{\overline{A}} = A$
10. $A + AB = A$
11. $A + \overline{A}B = A + B$
12. $(A + B)(A + C) = A + BC$

Exercises

Use Boolean Algebra to simplify the following:

a) $AB + A(B + C) + B(B + C)$

$= AB + AB + AC + BB + BC$ Distributive law

Exercises

Use Boolean Algebra to simplify the following:

a) AB + A(B + C) + B(B + C)

= AB + AB + AC + BB + BC — Distributive law

= AB + AC+ BB + BC — Rule 5 (AB + AB = AB)

Exercises

Use Boolean Algebra to simplify the following:

a) AB + A(B + C) + B(B + C)

= AB + AB + AC + BB + BC — Distributive law

= AB + AC+ BB + BC — Rule 5 (AB + AB = AB)

= AB + AC + B + BC — Rule 7 (BB = B)

Exercises

Use Boolean Algebra to simplify the following:

a) AB + A(B + C) + B(B + C)

= AB + AB + AC + BB + BC	Distributive law
= AB + AC+ BB + BC	Rule 5 (AB + AB = AB)
= AB + AC + B + BC	Rule 7 (BB = B)
= AB + AC + B	Rule 10 (B + BC = B)

Exercises

Use Boolean Algebra to simplify the following:

a) $AB + A(B + C) + B(B + C)$

$= AB + AB + AC + BB + BC$	Distributive law
$= AB + AC + BB + BC$	Rule 5 ($AB + AB = AB$)
$= AB + AC + B + BC$	Rule 7 ($BB = B$)
$= AB + AC + B$	Rule 10 ($B + BC = B$)
$= AB + B + AC$	Rule 10 ($B + BC = B$)
$= B + AC$	

Exercises

Use Boolean Algebra to simplify the following:

Original expression | Simplified expression

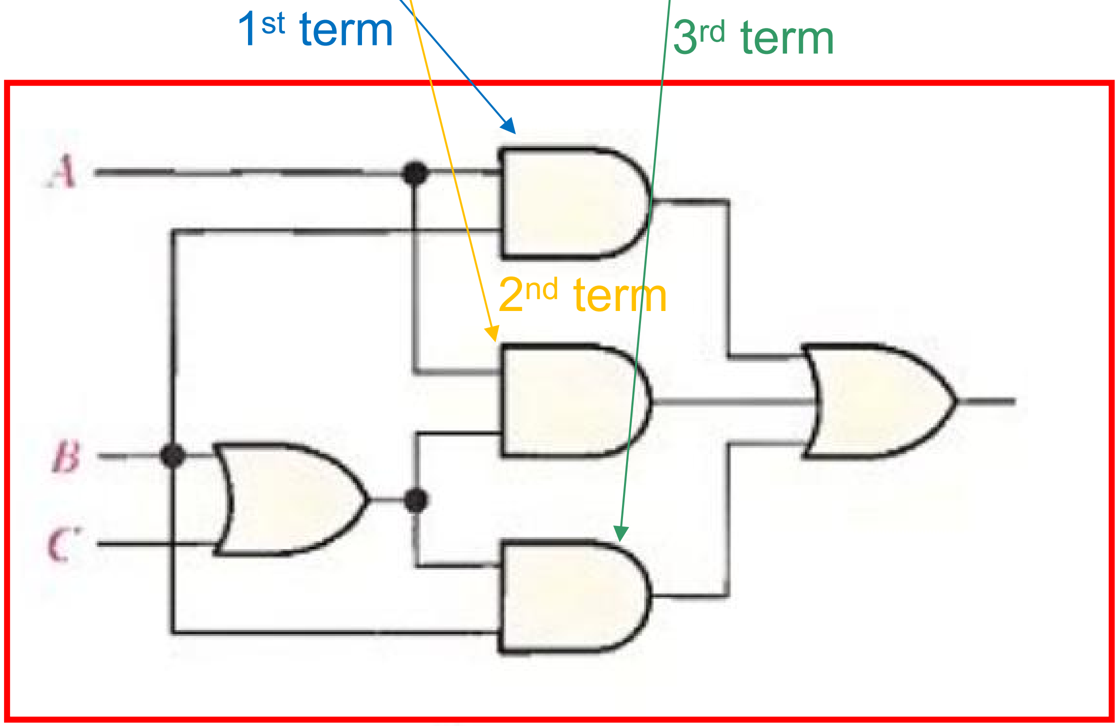
a) AB + A(B + C) + B(B + C) = B + AC

Draw the logic circuits for the Boolean expression shown above and the simplified circuit derived from your solutions.

Exercises

a) AB + A(B + C) + B(B + C)

Draw the logic circuits for the Boolean expression shown above and the simplified circuit B + AC

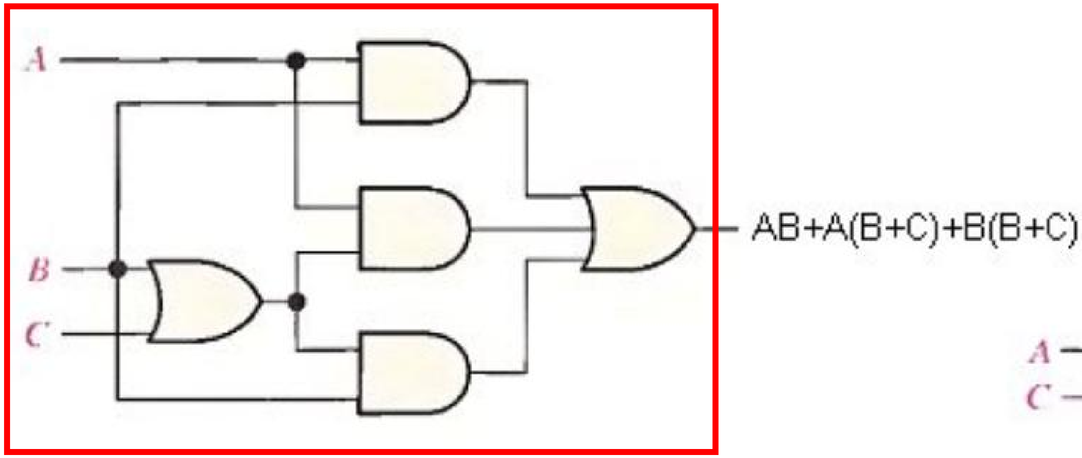


(a)

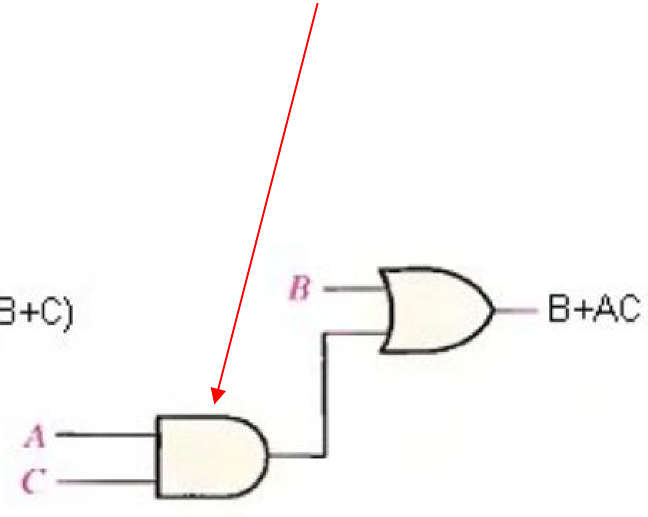
Exercises

a) AB + A(B + C) + B(B + C)

Draw the logic circuits for the Boolean expression shown above and the simplified circuit B + AC



(a)



(b)

Exercises

Use Boolean Algebra to simplify the following:

a) $AB + A(B + C) + B(B + C)$

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

c) $[A\overline{B}(C + BD) + \overline{A}\,\overline{B}]C$

d) $\overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + ABC$

Draw the logic circuits for the Boolean expression shown above and the simplified circuit derived from your solutions.

Exercises

Use Boolean Algebra to simplify the following:

a) $AB + A(B + C) + B(B + C)$

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

c) $[A\overline{B}(C + BD) + \overline{A}\,\overline{B}]C$

d) $\overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + ABC$

b) $A\overline{B}$

c) $\overline{B}C$

d) $\overline{B}\,\overline{C} + BC + AC$

Karnaugh Map Method (K-Map)

- A graphical method of simplifying logic equations or truth tables. Also called a K map.

Karnaugh Map Method (K-Map)

- A graphical method of simplifying logic equations or truth tables. Also called a K map.

- Theoretically can be used for any number of input variables, but practically limited to 4 variables. 5 or 6 variables can be done but gets complex !

Karnaugh Map Method

- The truth table values are placed in the K-map depending on their location.

Karnaugh Map Method

- The truth table values are placed in the K-map depending on their location.
- Adjacent K-map square differ in only one variable both horizontally and vertically.

Karnaugh Map Method

- The truth table values are placed in the K-map depending on their location.
- Adjacent K-map square differ in only one variable both horizontally and vertically.
- The pattern from top to bottom and left to right must be in the form $\bar{A}\bar{B}, \bar{A}B, AB, A\bar{B}$

LEFT → RIGHT

TOP TO BOTTOM ↓

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$				
$\bar{A}B$				
AB				
$A\bar{B}$				

Karnaugh Map Method

- The truth table values are placed in the K map as shown in on next slide.
- Adjacent K map square differ in only one variable both horizontally and vertically.
- The pattern from top to bottom and left to right must be in the form $\overline{A}\,\overline{B}, \overline{A}B, AB, A\overline{B}$
- A SOP expression can be obtained by OR-ing all squares that contain a 1.

From truth table to K-Map

2-inputs Truth table

K-Map

A	B	X	
0	0	1	→ $\overline{A}\,\overline{B}$
0	1	0	
1	0	0	
1	1	1	→ AB

From truth table to K-Map

2-inputs Truth table

A	B	X	
0	0	1	→ $\overline{A}\,\overline{B}$
0	1	0	
1	0	0	
1	1	1	→ AB

$$\left\{ x = \overline{A}\,\overline{B} + AB \right\}$$

(a)

K-Map

	$\overline{B}$	B
$\overline{A}$	1	0
A	0	1

From truth table to K-Map

K-Map

3-inputs Truth table

A	B	C	X	
0	0	0	1	→ $\bar{A}\bar{B}\bar{C}$
0	0	1	1	→ $\bar{A}\bar{B}C$
0	1	0	1	→ $\bar{A}B\bar{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	→ $AB\bar{C}$
1	1	1	0	

From truth table to K-Map

3-inputs Truth table

A	B	C	X	
0	0	0	1	$\rightarrow \bar{A}\bar{B}\bar{C}$
0	0	1	1	$\rightarrow \bar{A}\bar{B}C$
0	1	0	1	$\rightarrow \bar{A}B\bar{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	$\rightarrow AB\bar{C}$
1	1	1	0	

Sum of Products (SOP) expression

$$X = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C}$$

K-Map

	$\bar{C}$	C
$\bar{A}\bar{B}$	1	1
$\bar{A}B$	1	0
AB	1	0
$A\bar{B}$	0	0

(b)

From truth table to K-Map

4-inputs Truth table

A	B	C	D	X	
0	0	0	0	0	
0	0	0	1	1	→ $\bar{A}\bar{B}\bar{C}D$
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	→ $\bar{A}B\bar{C}D$
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	0	
1	1	0	1	1	→ $AB\bar{C}D$
1	1	1	0	0	
1	1	1	1	1	→ $ABCD$

From truth table to K-Map

4-inputs Truth table

A	B	C	D	X	
0	0	0	0	0	
0	0	0	1	1	→ $\bar{A}\bar{B}\bar{C}D$
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	→ $\bar{A}B\bar{C}D$
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	0	
1	1	0	1	1	→ $AB\bar{C}D$
1	1	1	0	0	
1	1	1	1	1	→ $ABCD$

$$X = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + ABCD$$

K-Map

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	1	0
$A\bar{B}$	0	0	0	0

(c)

Karnaugh Map Method

- Looping adjacent groups of 2, 4, or 8 that contains 1's will result in further simplification.

A	B	C	D	X	
0	0	0	0	0	
0	0	0	1	1	$\rightarrow \bar{A}\bar{B}\bar{C}D$
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	$\rightarrow \bar{A}B\bar{C}D$
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	0	
1	1	0	1	1	$\rightarrow AB\bar{C}D$
1	1	1	0	0	
1	1	1	1	1	$\rightarrow ABCD$

$$X = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + ABCD$$

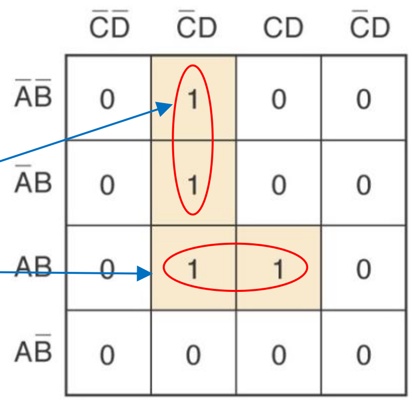
2 adjacent 1s

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	1	0
$A\bar{B}$	0	0	0	0

(c)

Karnaugh Map Method

- Looping adjacent groups of 2, 4, or 8 that contains 1's will result in further simplification.
- When the largest possible groups have been looped, only the common terms are placed in the final expression.

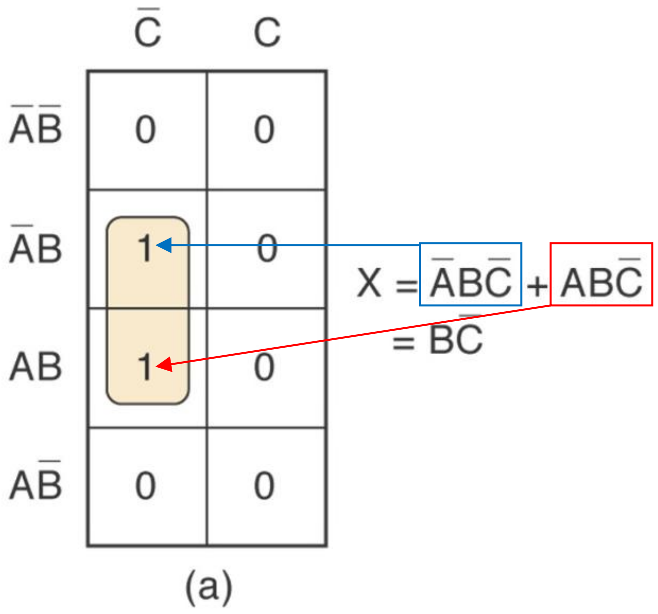


Final expression = $\bar{A}\bar{C}D + ABD$

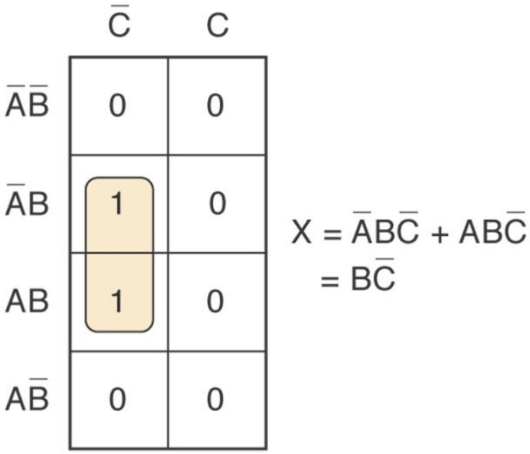
Karnaugh Map Method

- Looping adjacent groups of 2, 4, or 8 that contains 1's will result in further simplification.
- When the largest possible groups have been looped, only the common terms are placed in the final expression.
- Looping may also be wrapped between top, bottom, and sides.

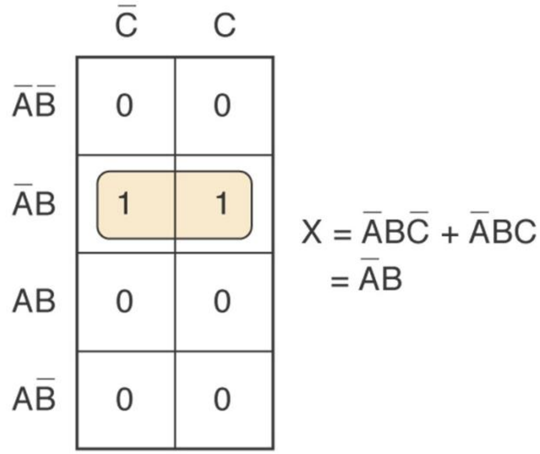
Looping pairs of adjacent 1's – One variable is eliminated.



Looping pairs of adjacent 1's – One variable is eliminated.

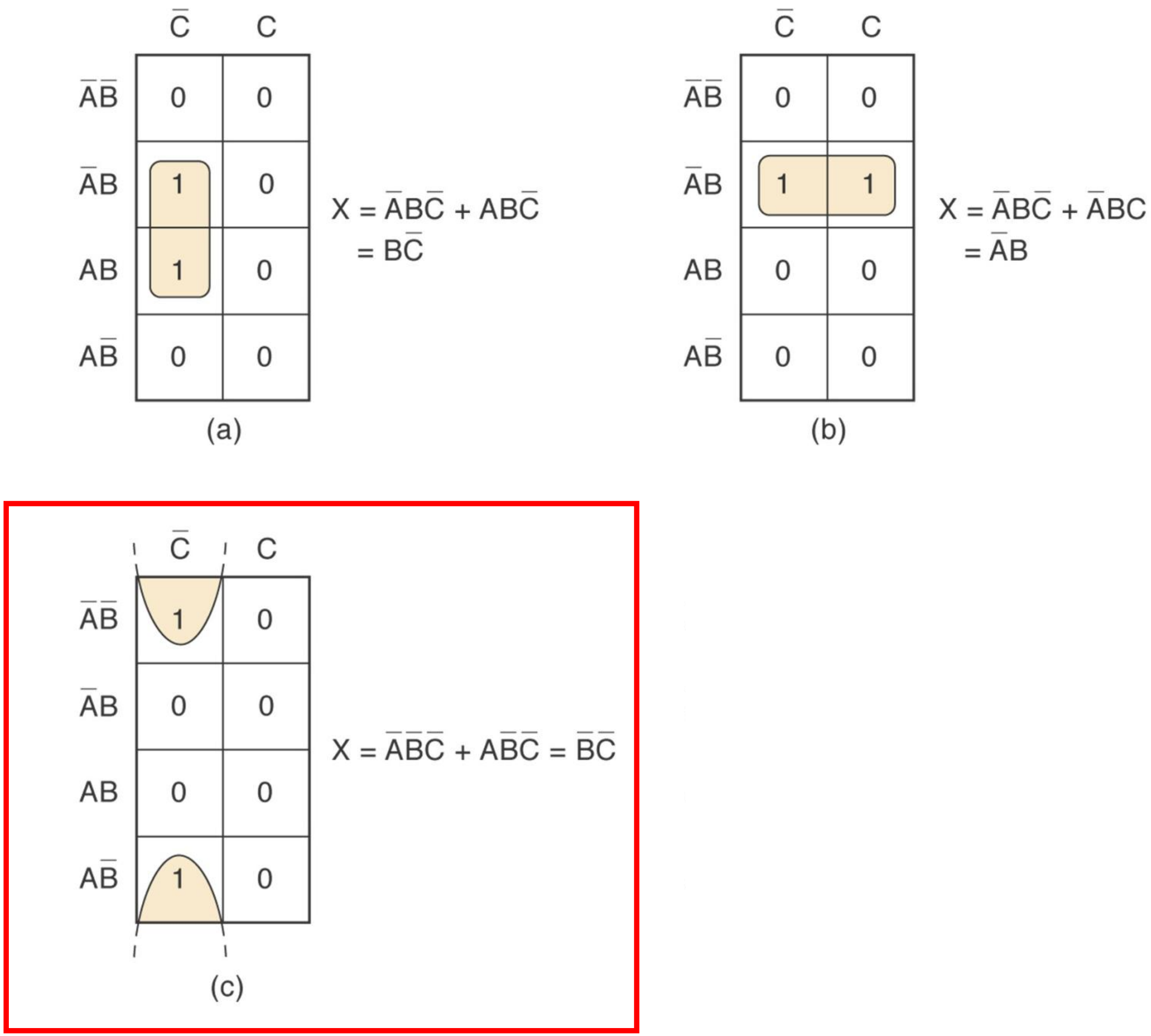


(a)

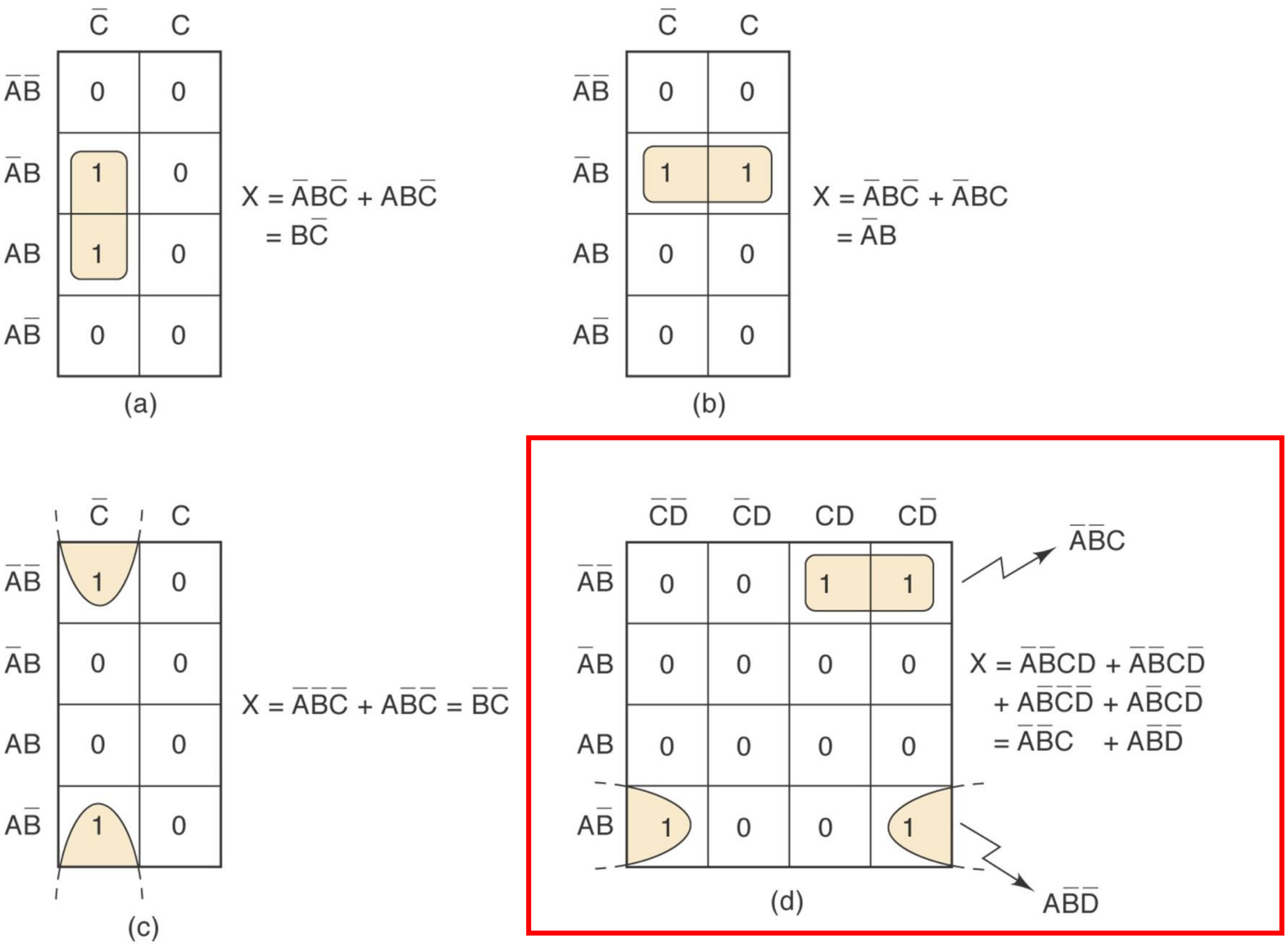


(b)

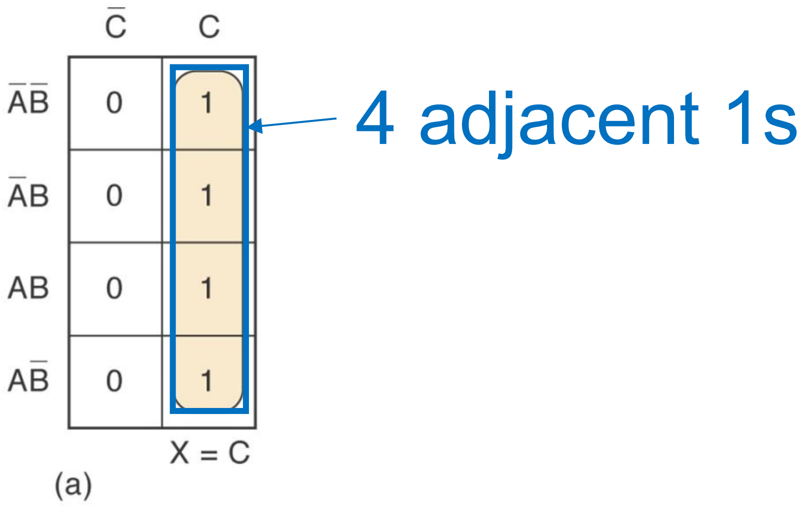
Looping pairs of adjacent 1's – One variable is eliminated.



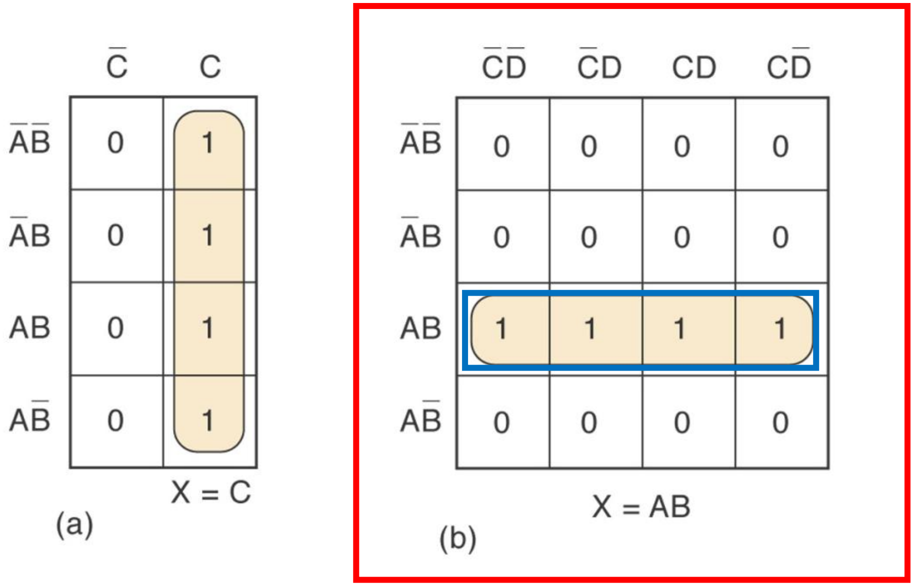
Looping pairs of adjacent 1's – One variable is eliminated.



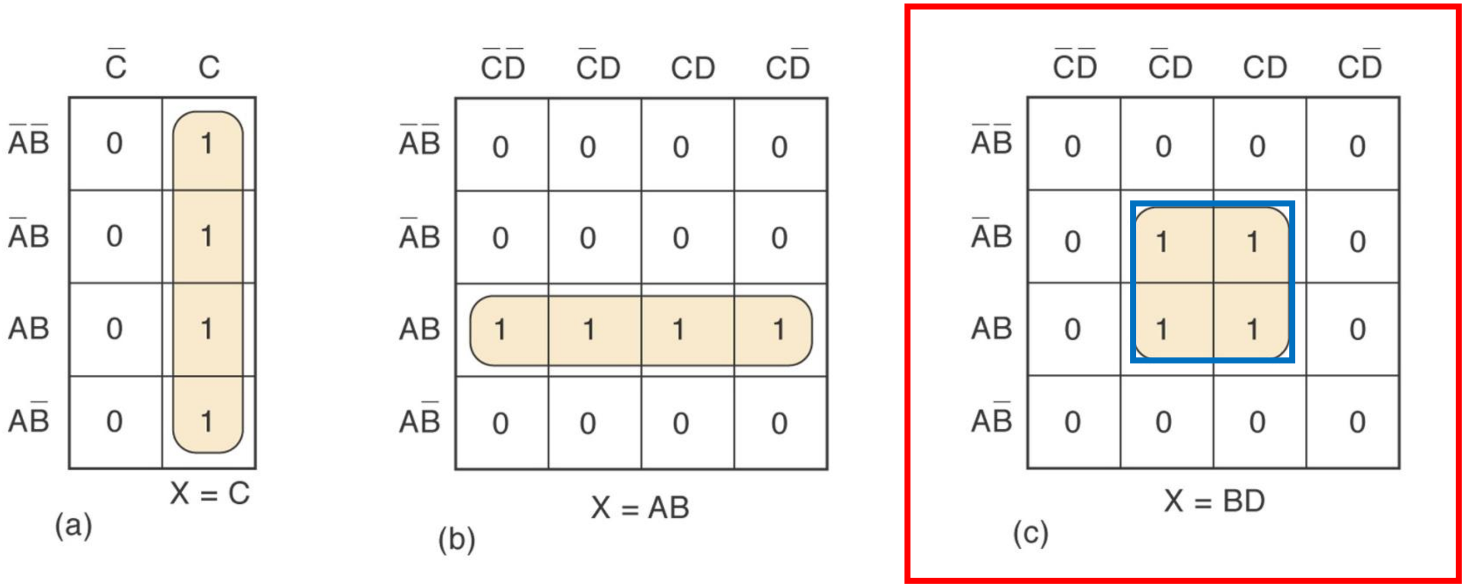
Looping groups of four adjacent 1's – Two variables are eliminated.



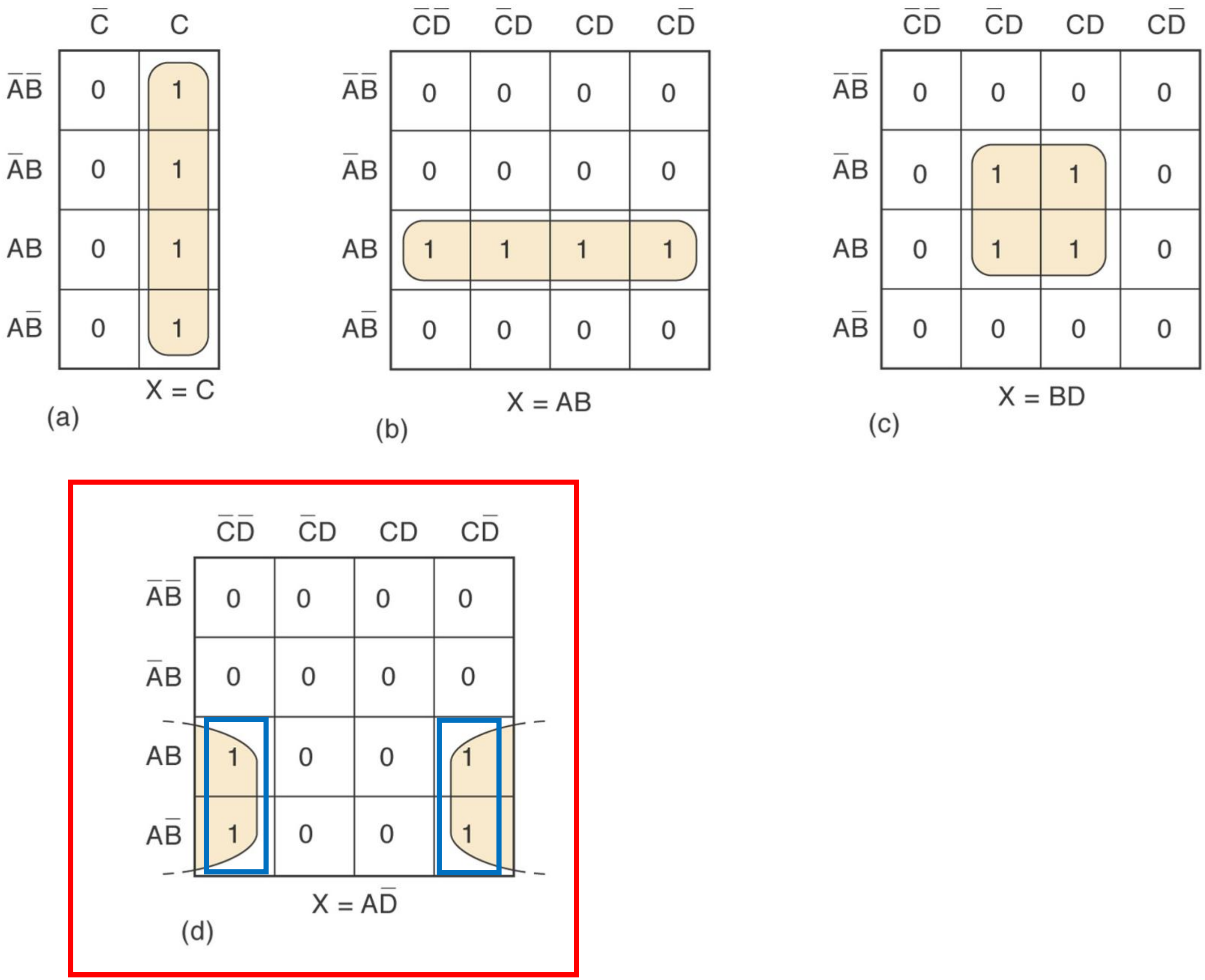
Looping groups of four adjacent 1's – Two variables are eliminated.



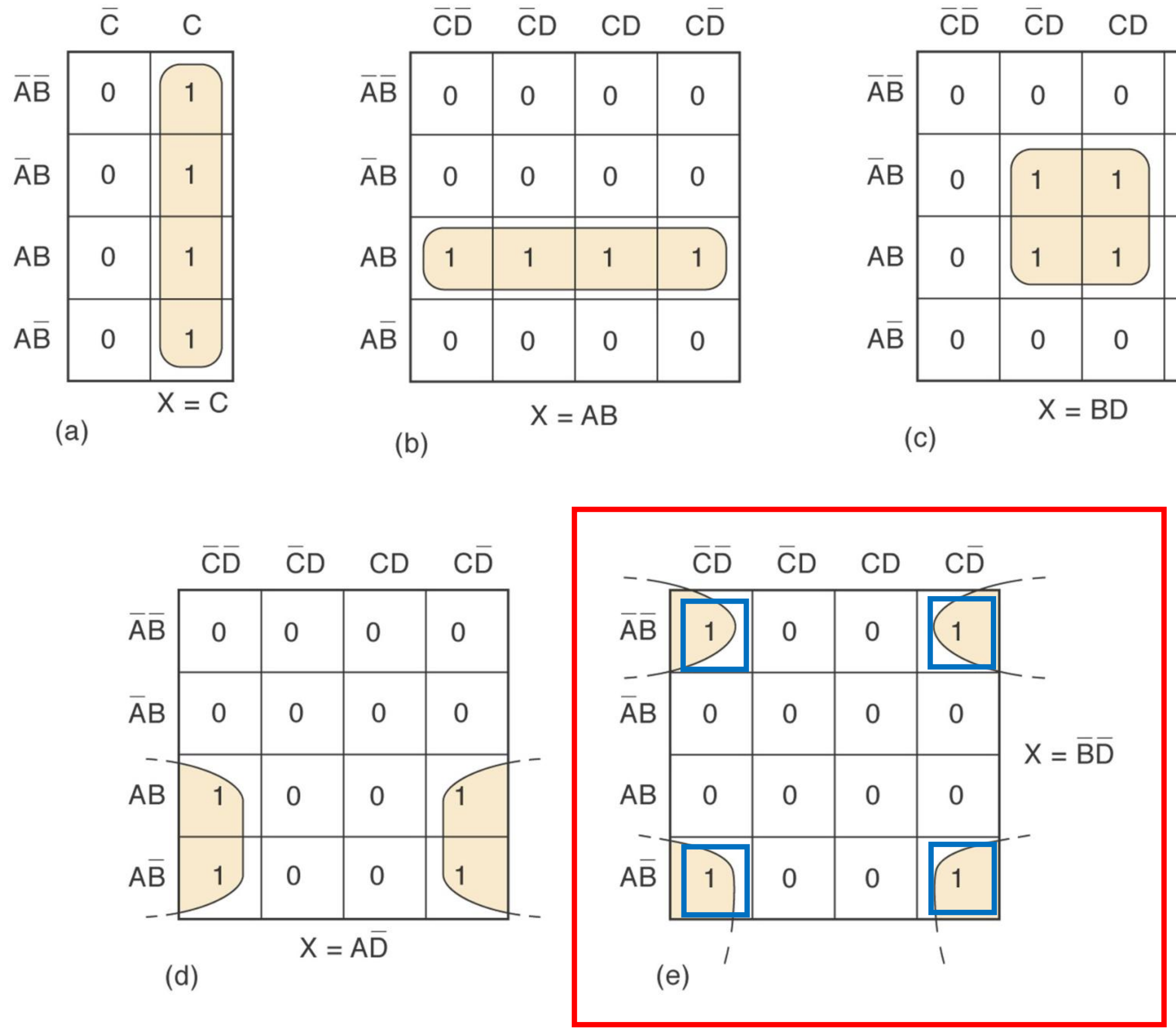
Looping groups of four adjacent 1's – Two variables are eliminated.



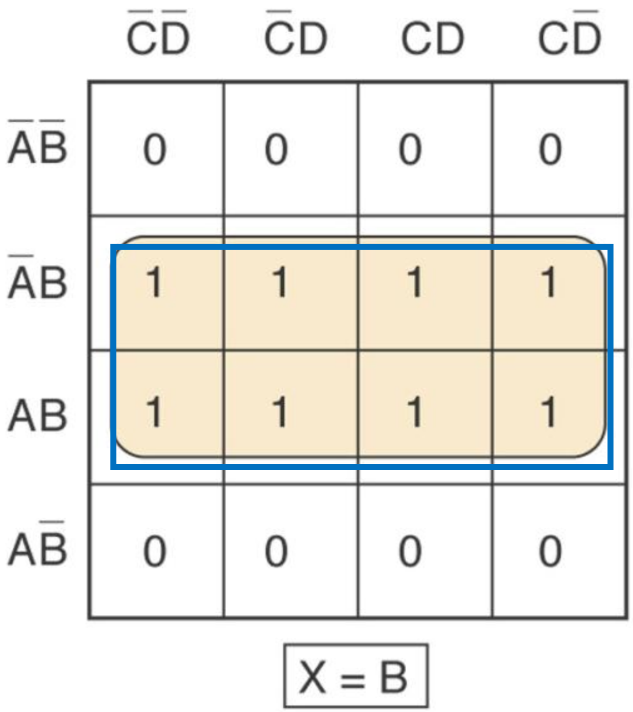
Looping groups of four adjacent 1's – Two variables are eliminated.



Looping groups of four adjacent 1's – Two variables are eliminated.

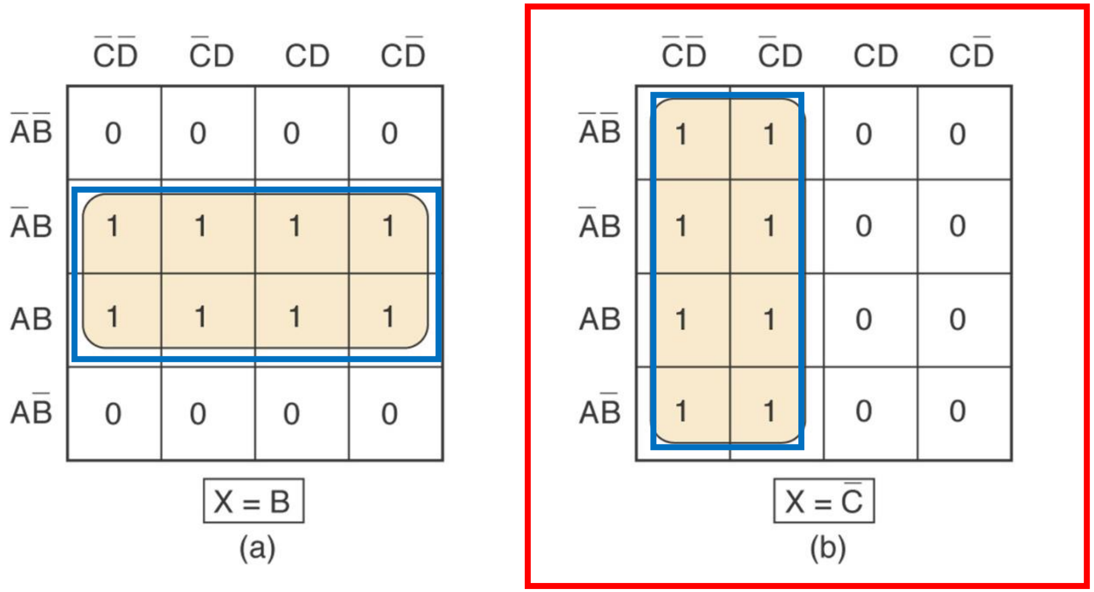


Looping eights of adjacent 1's – Three variables are eliminated.



(a)

Looping eights of adjacent 1's – Three variables are eliminated.



Looping eights of adjacent 1's – Three variables are eliminated.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	1	1	1	1
AB	1	1	1	1
$A\bar{B}$	0	0	0	0

$X = B$

(a)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	0
$\bar{A}B$	1	1	0	0
AB	1	1	0	0
$A\bar{B}$	1	1	0	0

$X = \bar{C}$

(b)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	1	1
$\bar{A}B$	0	0	0	0
AB	0	0	0	0
$A\bar{B}$	1	1	1	1

$X = \bar{B}$

(c)

Looping eights of adjacent 1's – Three variables are eliminated.