Boolean Algebra and Logic Simplification

BY UNSA SHAKIR

Boolean Algebra

- Boolean algebra is a mathematical system for the manipulation of variables that can have one of two values.
  - In formal logic, these values are “true” and “false.”
  - In digital systems, these values are “on” and “off,” 1 and 0, or “high” and “low.”

- When we learned numbers like 1, 2, 3, we also then learned how to add, multiply, etc. with them. Boolean Algebra covers operations that we can do with 0’s and 1’s. Computers do these operations ALL THE TIME and they are basic building blocks of computation inside your computer program.

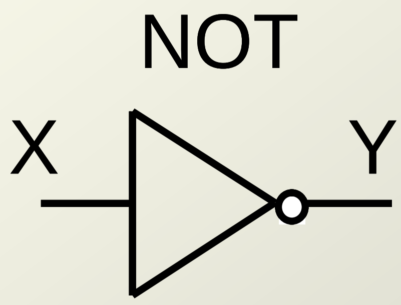
How does Boolean Algebra fit into the big picture?

- It is part of the Combinational Logic topics (memoryless)
  - Different from the Sequential logic topics (can store information)

- Learning Axioms and theorems of Boolean algebra
  - Allows you to design logic functions
  - Allows you to know how to combine different logic gates
  - Allows you to simplify or optimize on the complex operations

Basic Logic Gates

NOT Gate -- Inverter



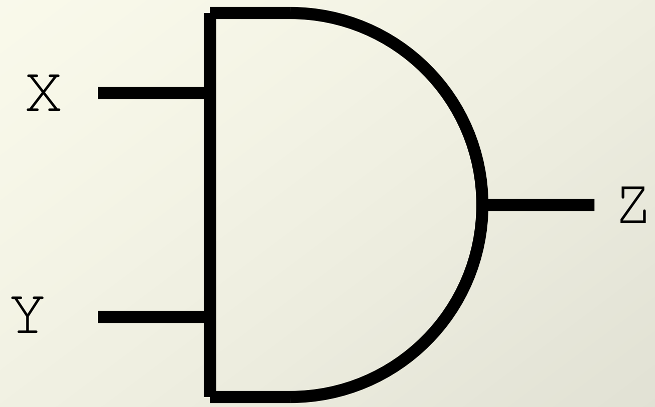
Y = ~X

X	Y
0	1
1	0

- The NOT operation is most often designated by an overbar. It is sometimes indicated by a prime mark (‘) or an “elbow” (¬).

AND Gate

AND (Multiplication)

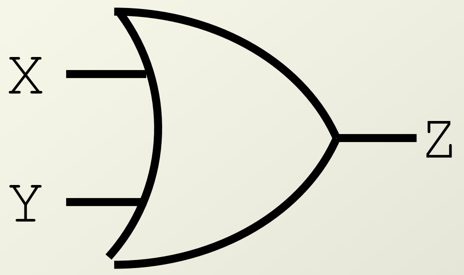


Z = X . Y

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

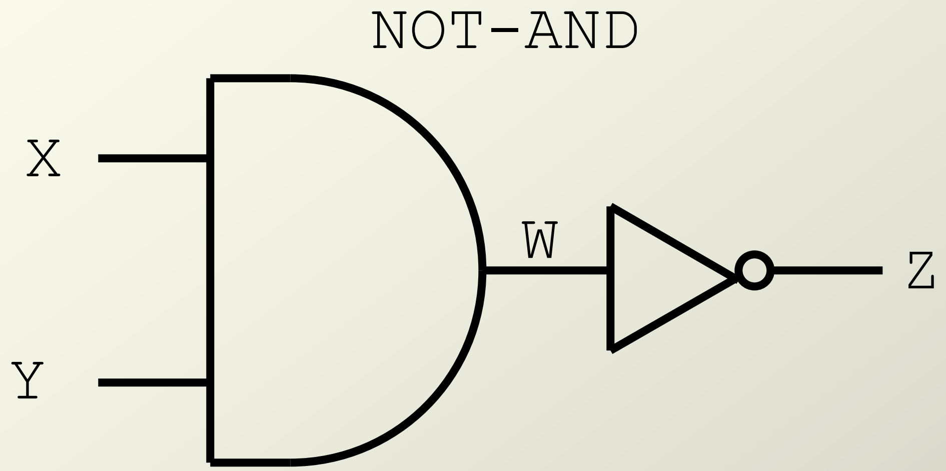
OR (Addition)



Z = X + Y

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

NAND Gate

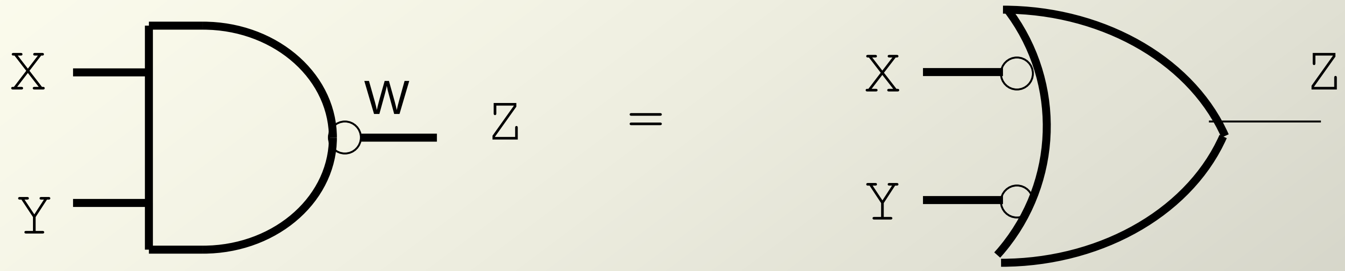


X	Y	W	Z
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

W = X . Y

Z = ~W = ~(X . Y)

NAND Gate



Z = ~(X . Y)

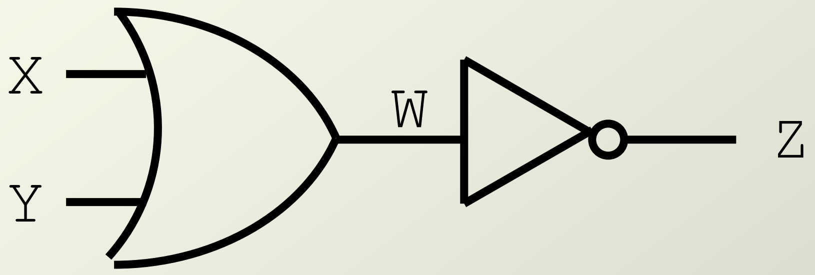
Z = ~X + ~Y

X	Y	W	Z
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

X	Y	~X	~Y	Z
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

NOR Gate

NOT-OR

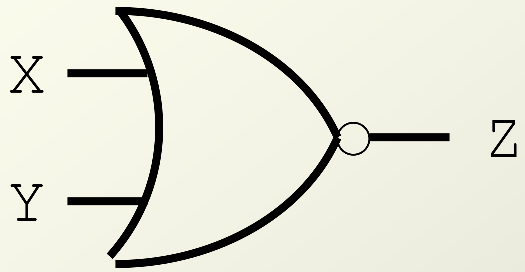


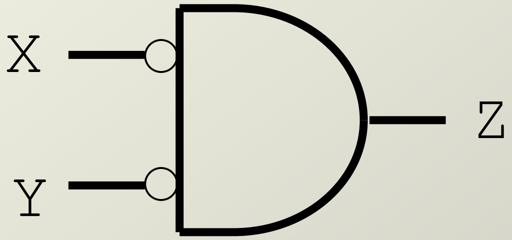
X	Y	W	Z
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

W = X + Y

Z = ~W = ~(X + Y)

NOR Gate





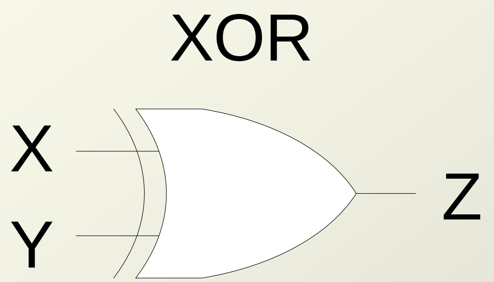
Z = ~(X + Y)

Z = ~X . ~Y

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

X	Y	~X	~Y	Z
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Exclusive-OR Gate



`xor(Z,X,Y)`

$$A.\overline{B}+\overline{A}.B=Y$$

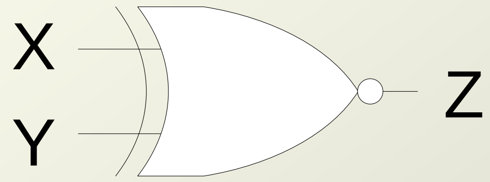
$$A\oplus B=Y$$

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

- The output of an XOR gate is **true (1)** only when exactly one of its inputs is **true (1)**. If both of an XOR gate's inputs are **false (0)**, or if both of its inputs are **true (1)**, then the output of the XOR gate is **false (0)**.

Exclusive-NOR Gate

XNOR



xnor(Z,X,Y)

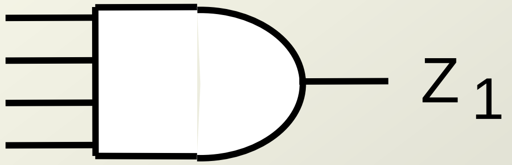
$$\overline{A}.\overline{B} + A.B = Y$$

$$\overline{A \oplus B} = Y$$

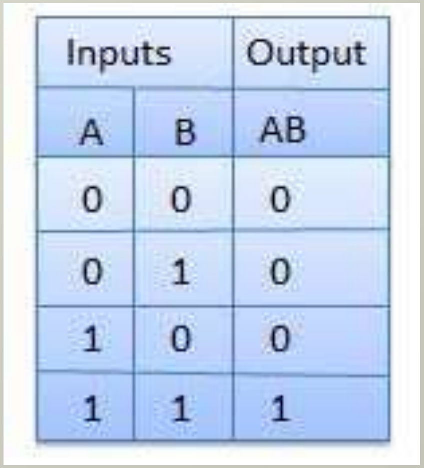
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

- The output of an XNOR gate is **true (1)** when all of its inputs are **true (1)** or when all of its inputs are **false (0)**. If some of its inputs are **true (1)** and others are **false (0)**, then the output of the XNOR gate is **false(0)**.

Multiple-input AND Gate

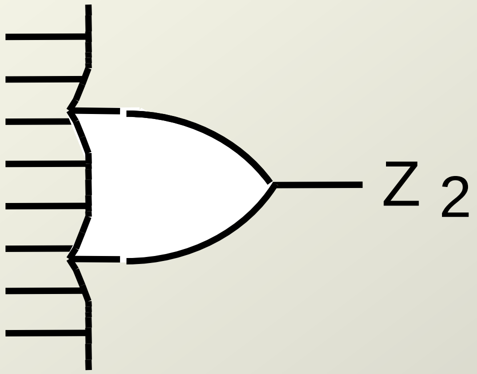


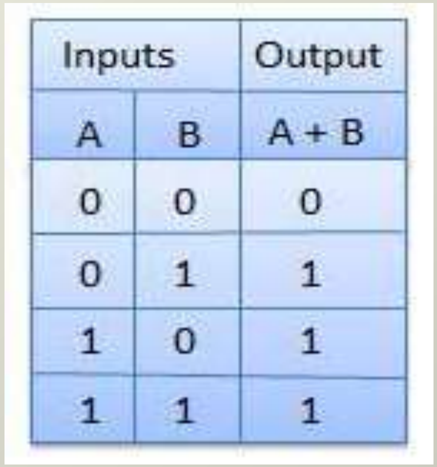
Output Z_1 is HIGH only if all inputs are HIGH



Inputs		Output
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

Multiple-input OR Gate

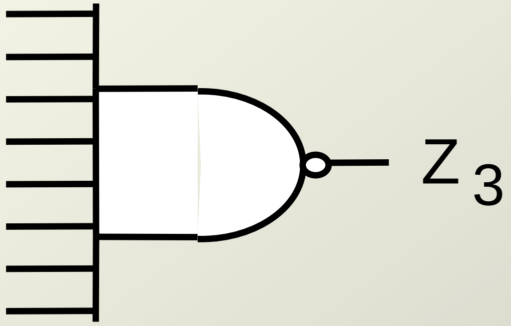


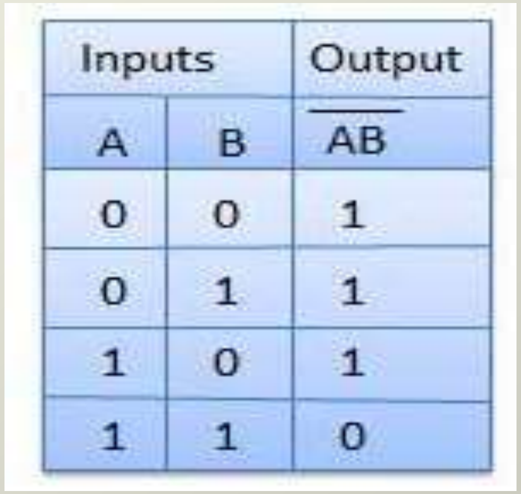


Inputs		Output
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

Output Z_2 is LOW only if all inputs are LOW

Multiple-input NAND Gate

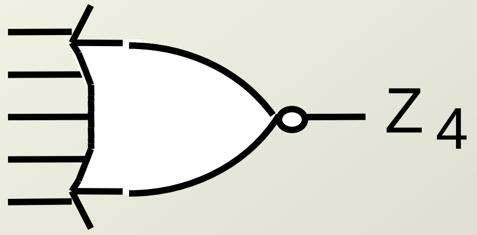




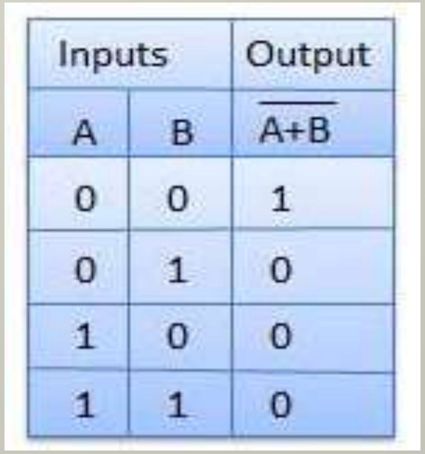
Inputs		Output
A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

Output Z_3 is LOW only if all inputs are HIGH

Multiple-input NOR Gate



Output Z_4 is HIGH only if all inputs are LOW



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

Some notation

- Priorities: $\overline{A} \bullet B + C = ((\overline{A}) \bullet B) + C$

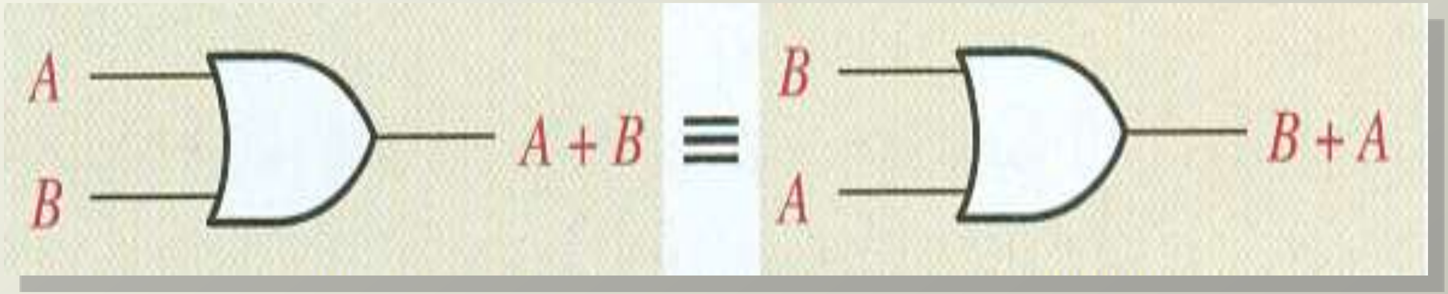
1. Parentheses
2. NOT > AND > XOR > OR

- Variables and their complements are sometimes called **literals**

Laws of Boolean Algebra

- Commutative Law of Addition:

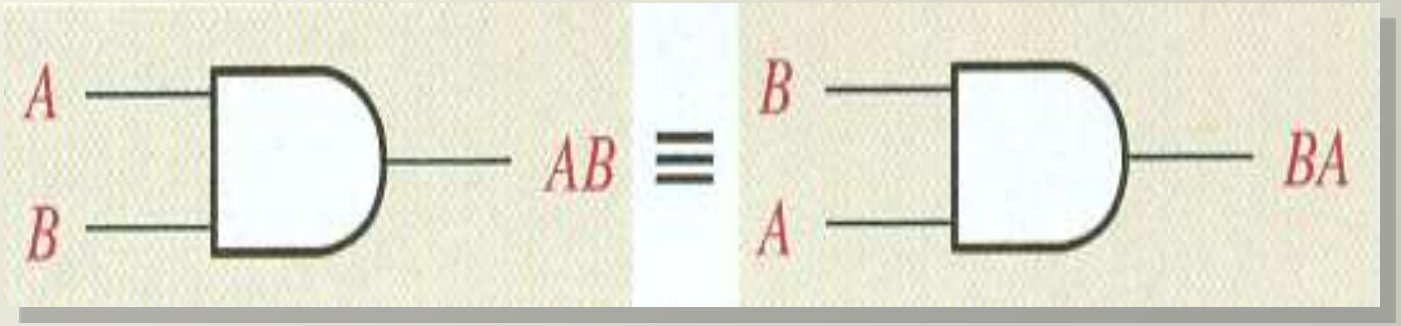
$$A + B = B + A$$



Laws of Boolean Algebra

- Commutative Law of Multiplication:

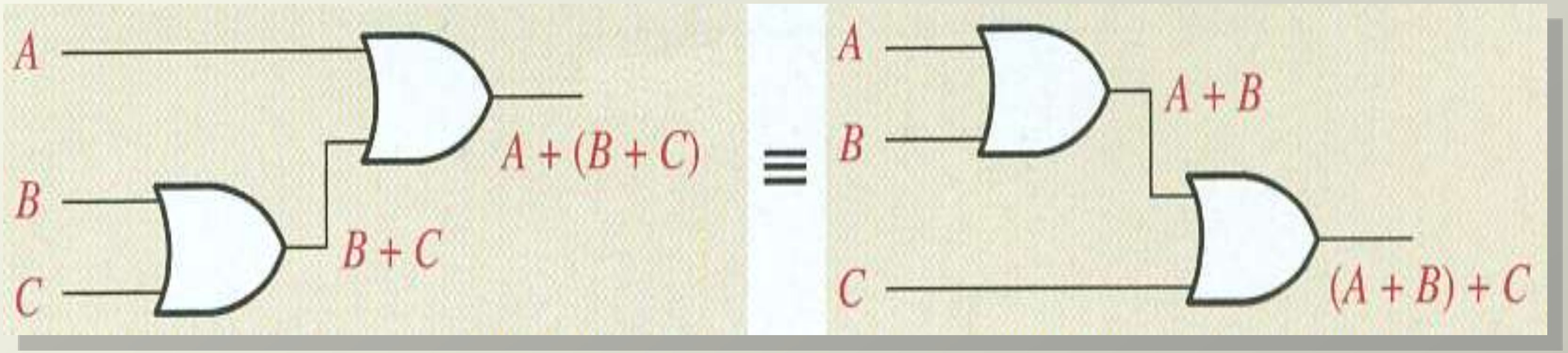
A * B = B * A



Laws of Boolean Algebra

- Associative Law of Addition:

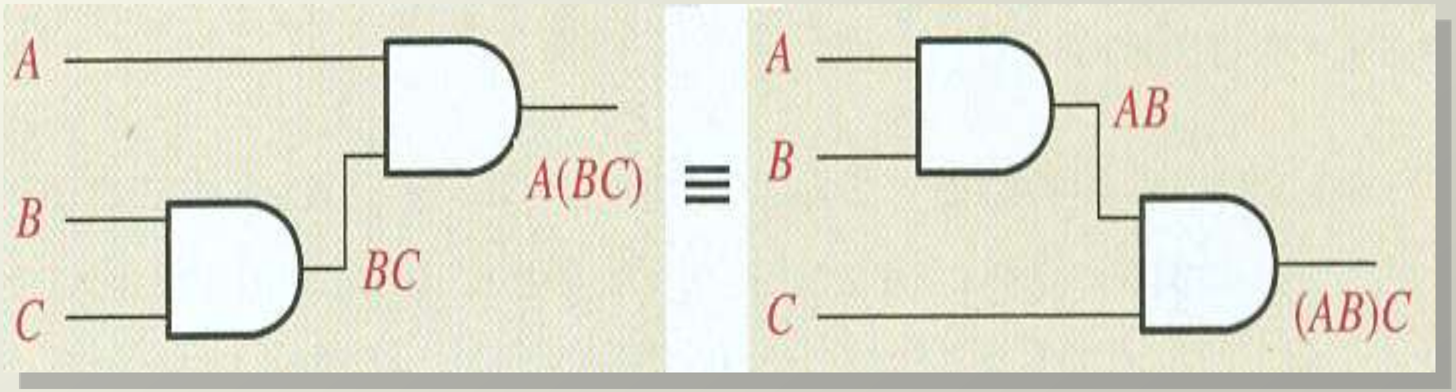
$$A + (B + C) = (A + B) + C$$



Laws of Boolean Algebra

- Associative Law of Multiplication:

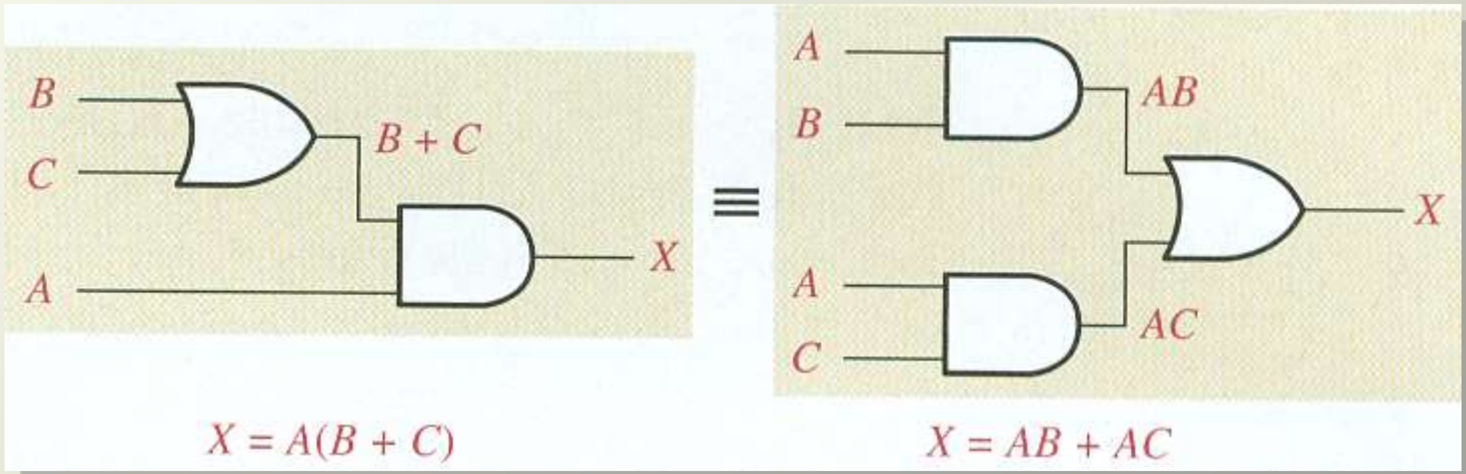
A * (B * C) = (A * B) * C



Laws of Boolean Algebra

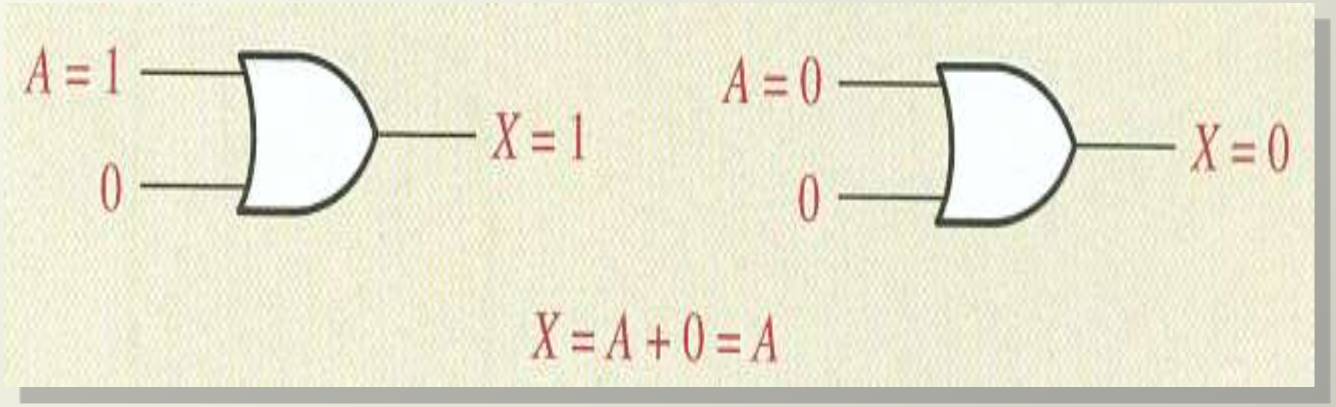
- Distributive Law:

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



Rules of Boolean Algebra

- Rule 1   IDENTITY w.r.t ADDITION



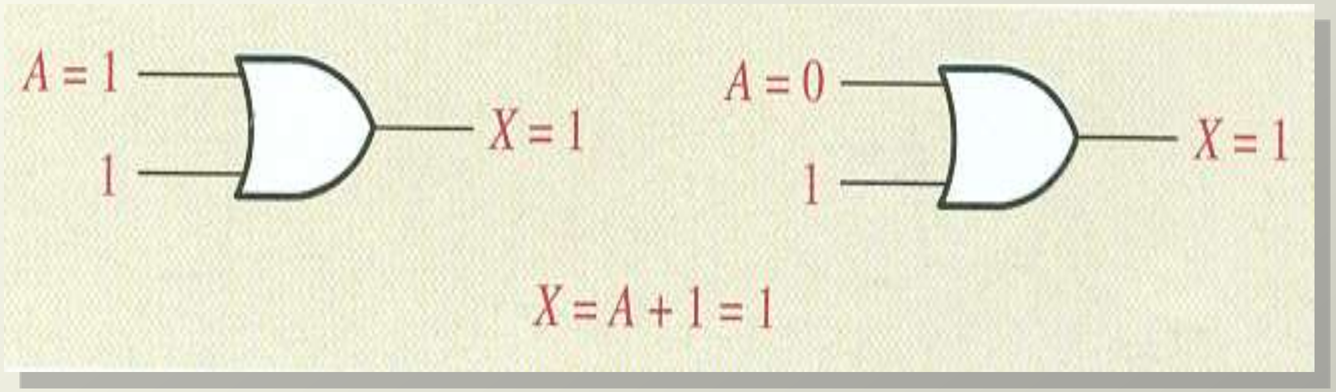
$X = A + 0 = A$

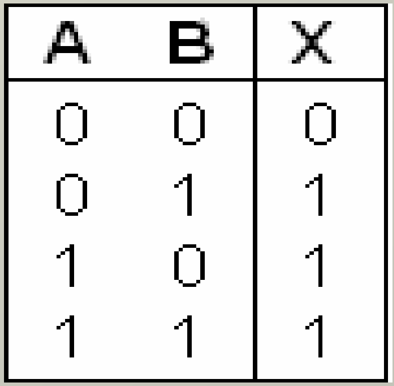
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Rules of Boolean Algebra

- Rule 2   NULL w.r.t ADDITION



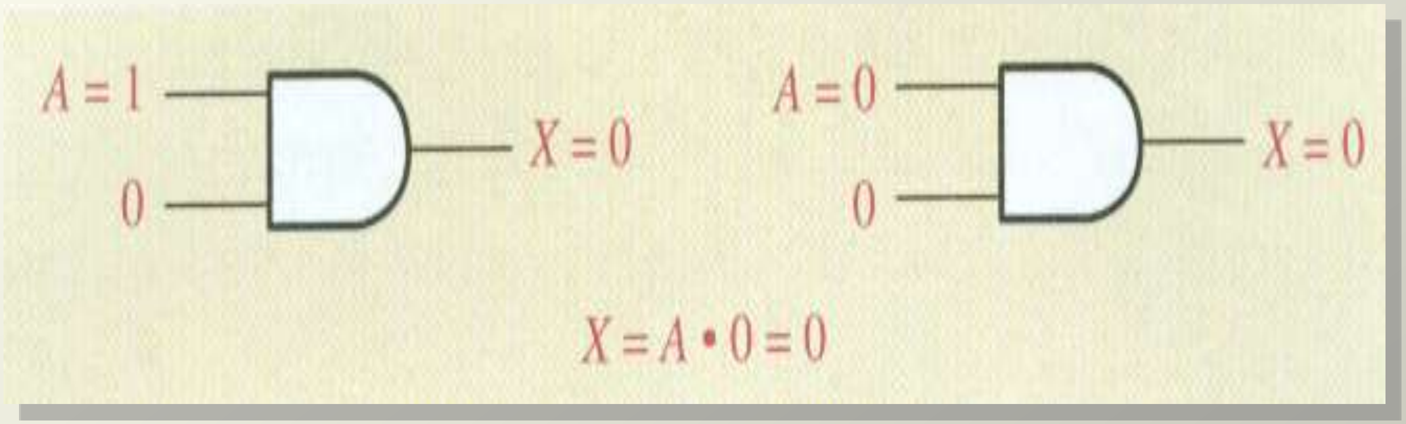


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Rules of Boolean Algebra

- Rule 3 NULL w.r.t MULTIPLICATION

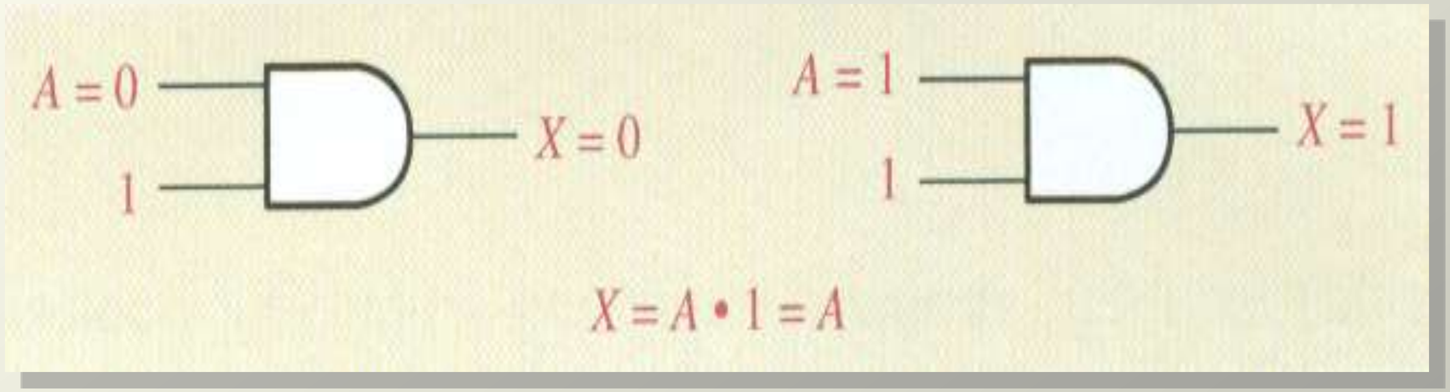


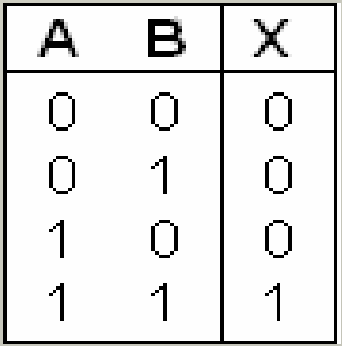
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Rules of Boolean Algebra

- Rule 4 IDENTITY w.r.t MULTIPLICATION



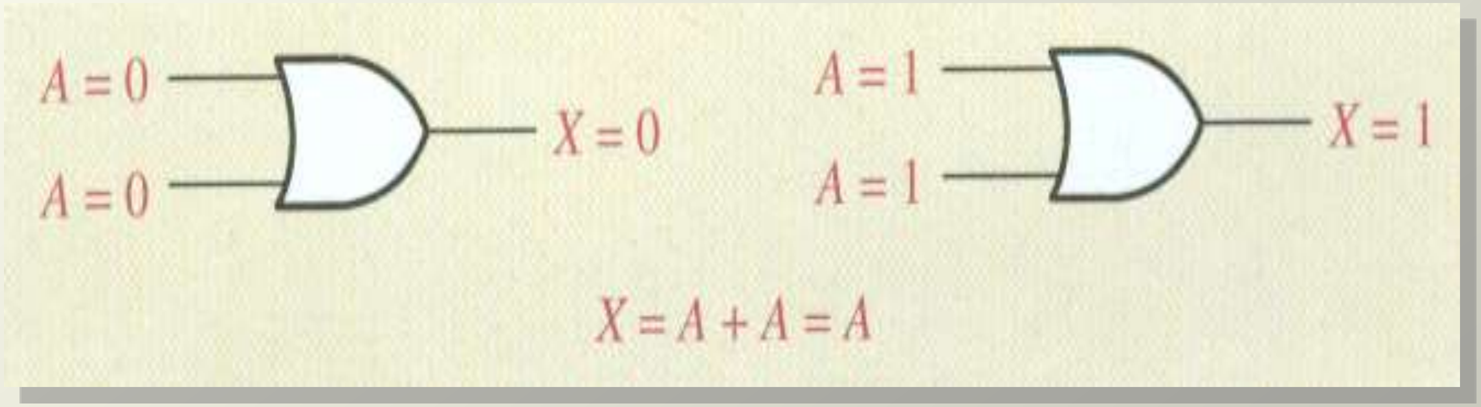


A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Rules of Boolean Algebra

- Rule 5 IDEMPOTENT w.r.t ADDITION

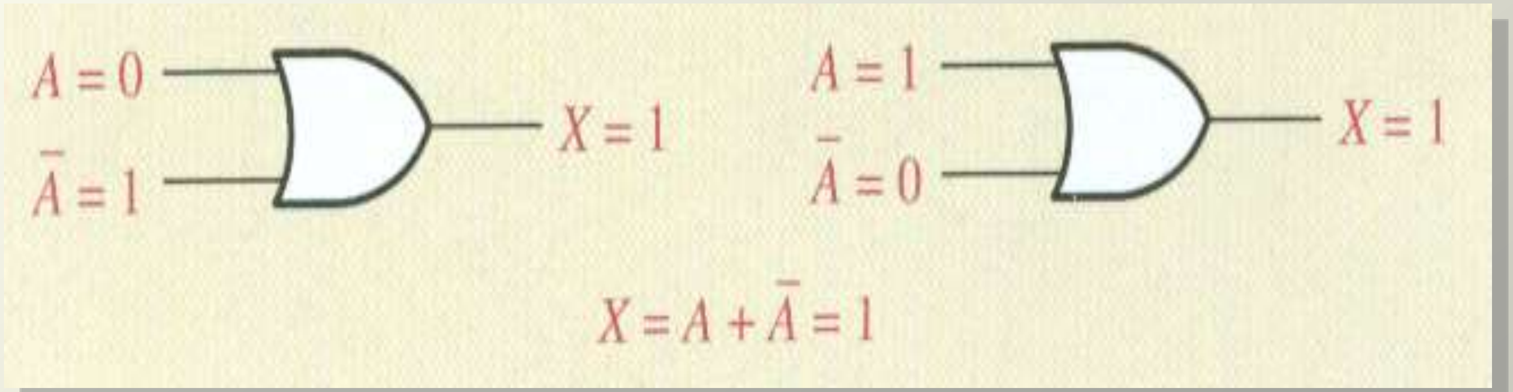


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Rules of Boolean Algebra

- Rule 6 COMPLEMENTARITY w.r.t ADDITION



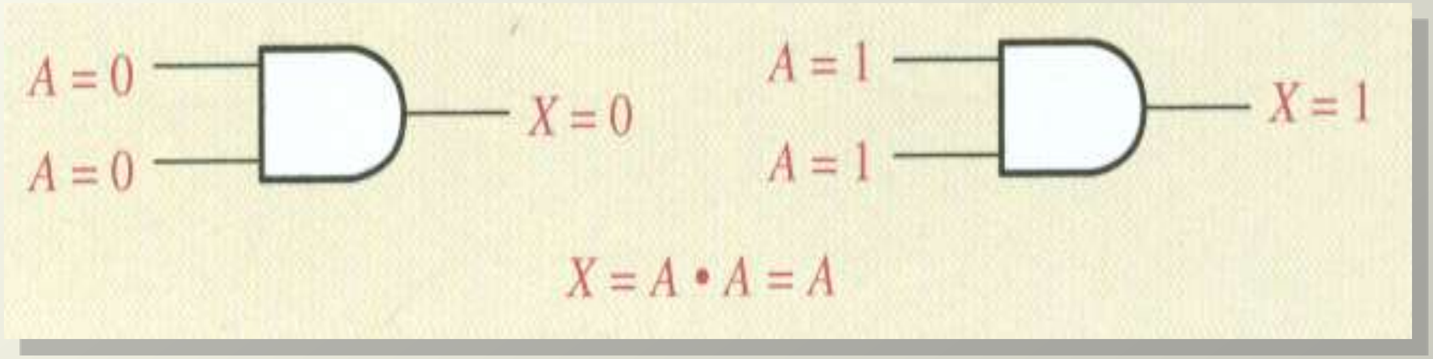
$X = A + \bar{A} = 1$

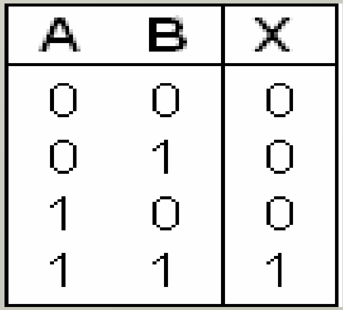
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Rules of Boolean Algebra

- Rule 7 IDEMPOTENT w.r.t MULTIPLICATION



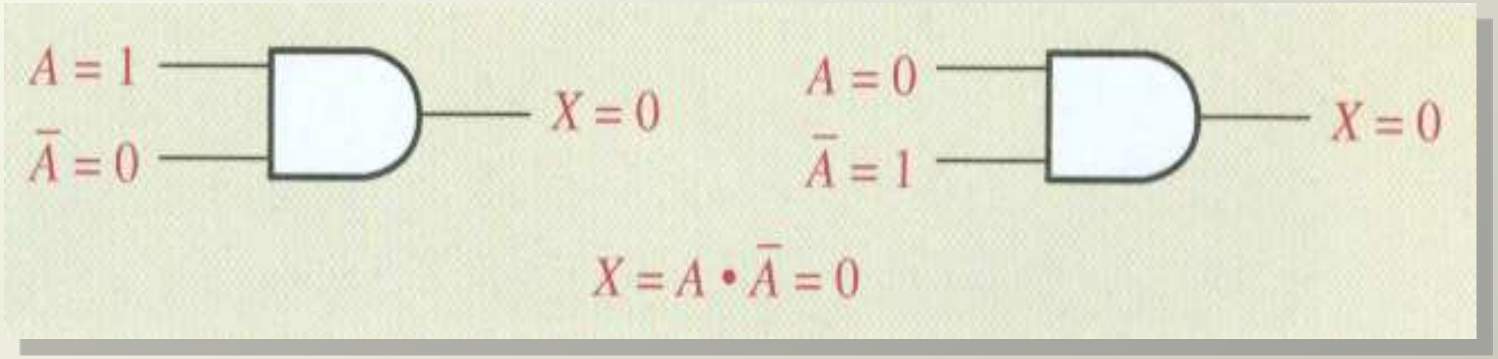


A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Rules of Boolean Algebra

- Rule 8 COMPLEMENTARITY w.r.t MULTIPLICATION

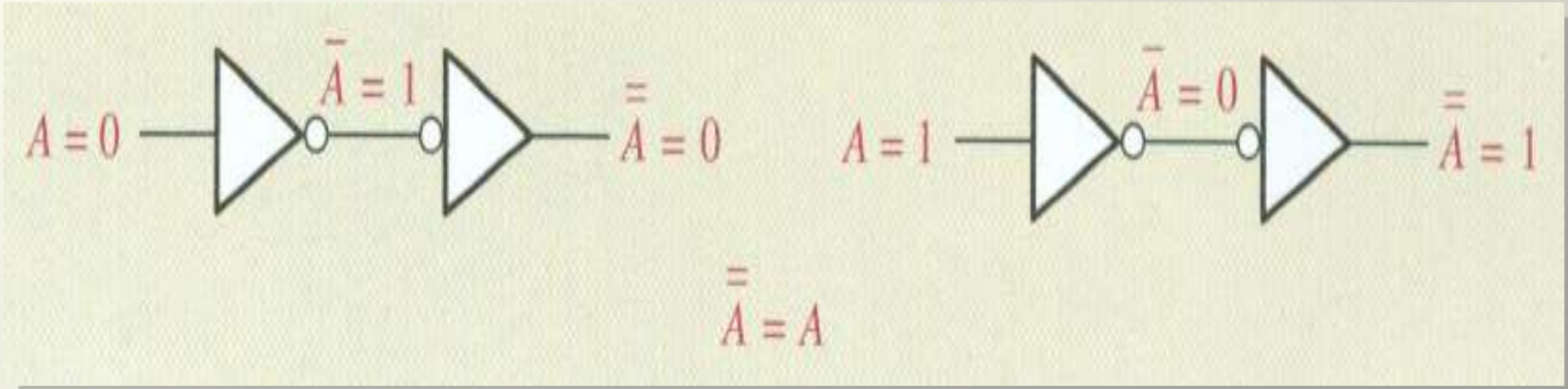


A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

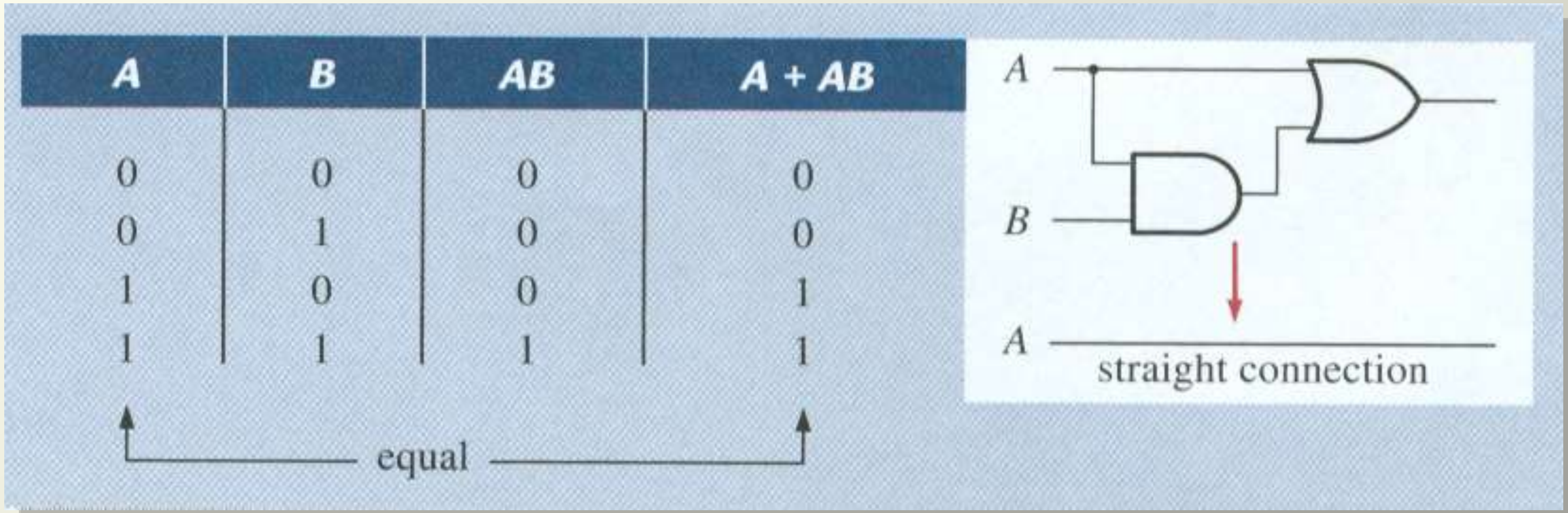
Rules of Boolean Algebra

- Rule 9 INVOLUTION

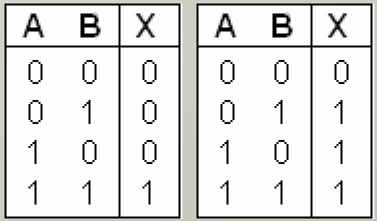


Rules of Boolean Algebra

- Rule 10: A + AB = A



A	*B*	*AB*	*A + AB*
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

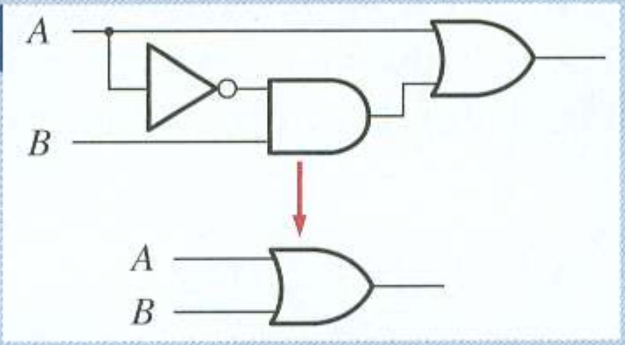
OR Truth Table

Rules of Boolean Algebra

- Rule 11: $A + \overline{A}B = A + B$

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

equal



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

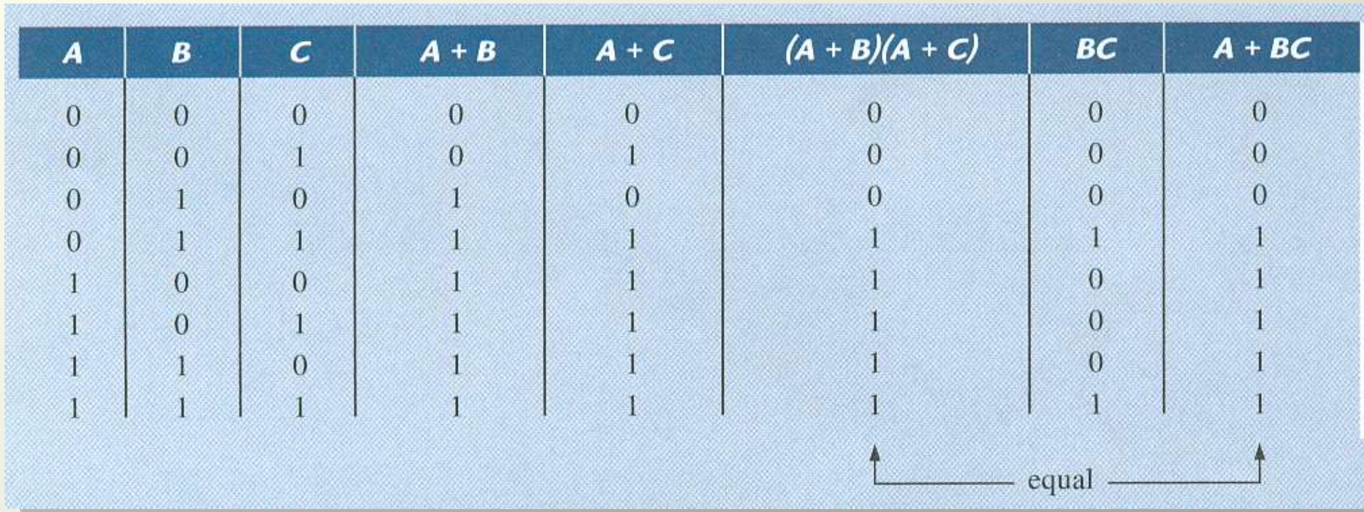
AND Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

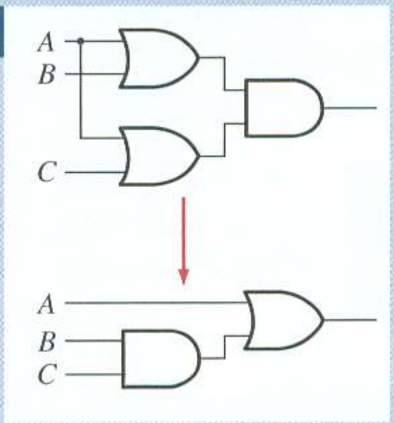
Rules of Boolean Algebra

- Rule 12: (A + B)(A + C) = A + BC



A	B	C	$A+B$	$A+C$	$(A+B)(A+C)$	BC	$A+BC$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

equal



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

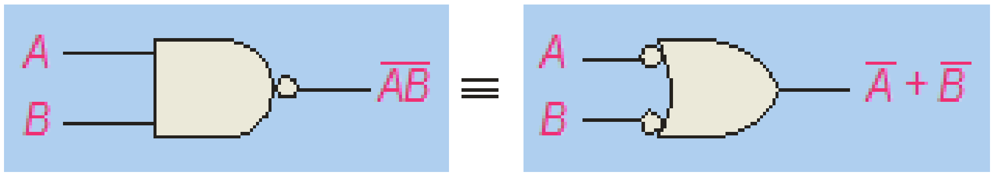
DeMorgan's Theorem

DeMorgan's 1st Theorem

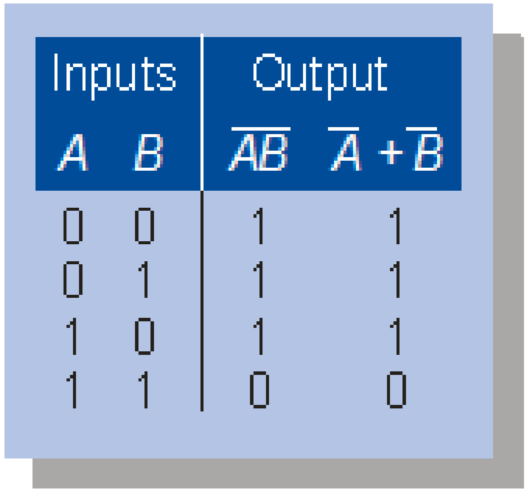
The complement of a product of variables is equal to the sum of the complemented variables.

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

Applying DeMorgan's first theorem to gates:



NAND Negative-OR



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

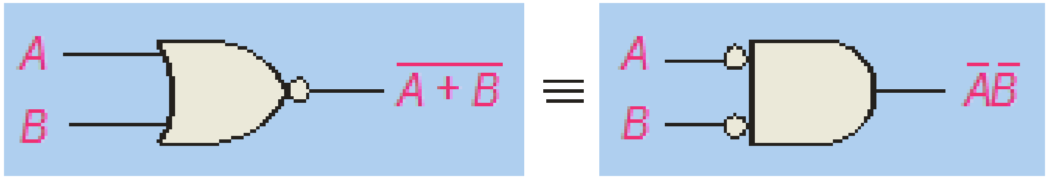
DeMorgan's Theorem

DeMorgan's 2nd Theorem

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

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

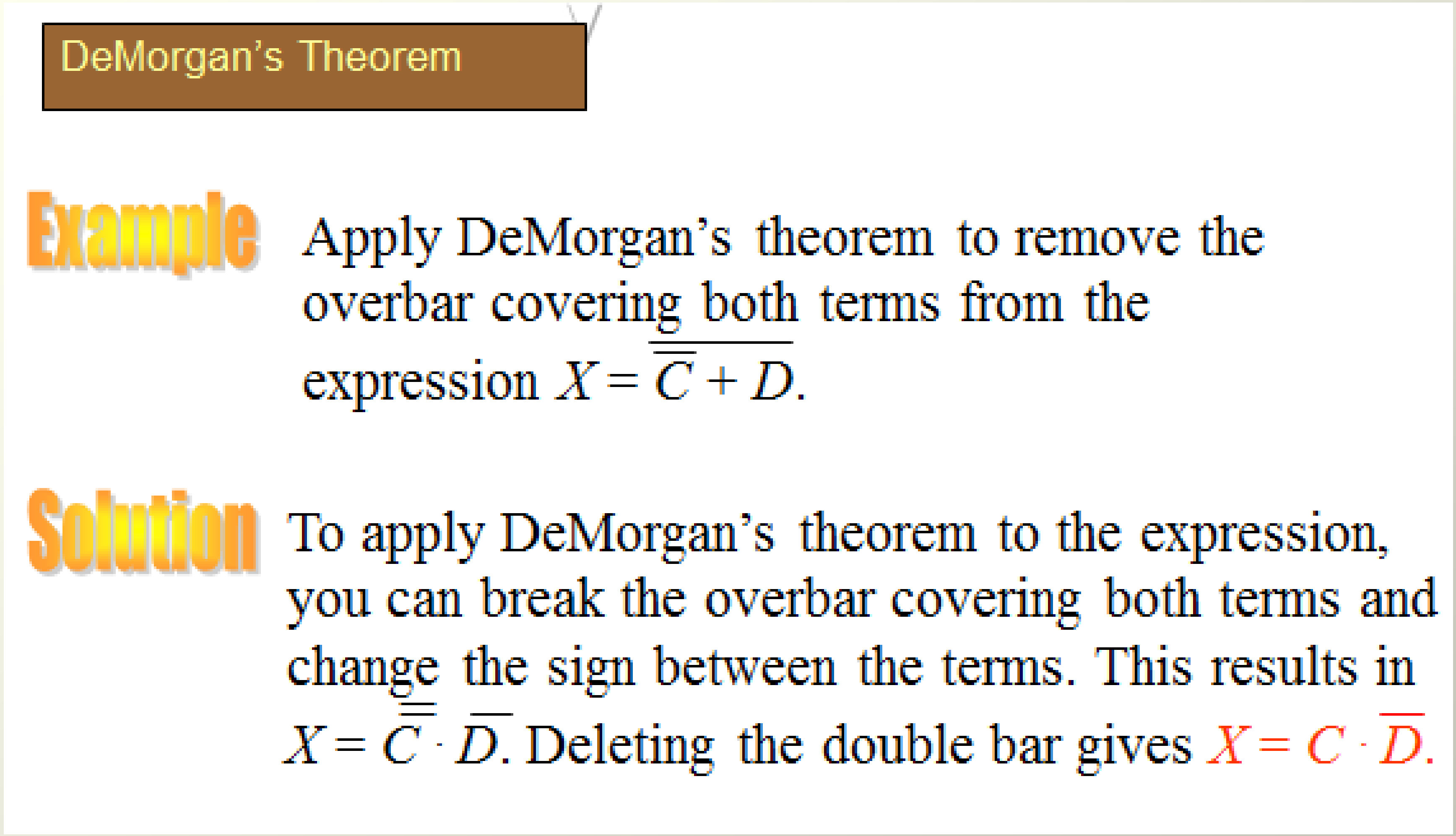
Applying DeMorgan's second theorem to gates:



NOR

Negative-AND

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

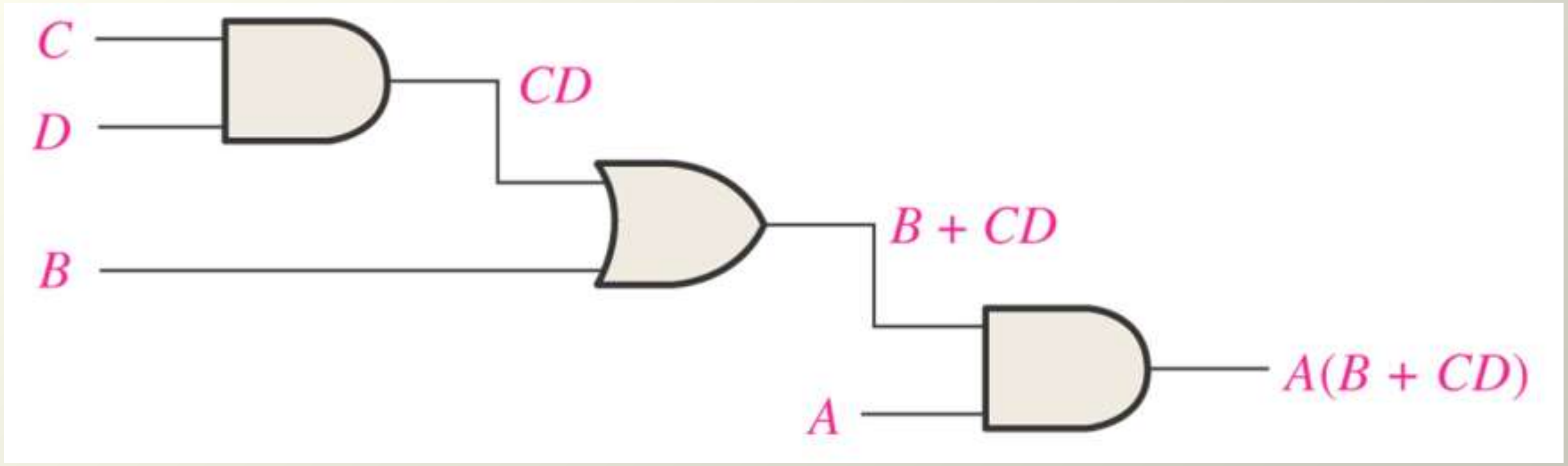


DeMorgan's Theorem

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

Solution 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 $X = \overline{\overline{C}} \cdot \overline{D}$. Deleting the double bar gives $X = C \cdot \overline{D}$.

A logic circuit showing the development of the Boolean expression for the output.



Example 1

Determine if the following equation is valid

$$\bar{x}_1\bar{x}_3 + x_2x_3 + x_1\bar{x}_2 = \bar{x}_1x_2 + x_1x_3 + \bar{x}_2\bar{x}_3$$

Left-Hand Side (LHS)

Row number	x_1	x_2	x_3	$\overline{x}_1 \overline{x}_3$	$x_2 x_3$	$x_1 \overline{x}_2$	f
0	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
2	0	1	0	1	0	0	1
3	0	1	1	0	1	0	1
4	1	0	0	0	0	1	1
5	1	0	1	0	0	1	1
6	1	1	0	0	0	0	0
7	1	1	1	0	1	0	1

Right-Hand Side (RHS)

Row number	x_1	x_2	x_3	$\overline{x}_1 x_2$	$x_1 x_3$	$\overline{x}_2\ \overline{x}_3$	f
0	0	0	0	0	0	1	1
1	0	0	1	0	0	0	0
2	0	1	0	1	0	0	1
3	0	1	1	1	0	0	1
4	1	0	0	0	0	1	1
5	1	0	1	0	1	0	1
6	1	1	0	0	0	0	0
7	1	1	1	0	1	0	1

$$\underbrace{\bar{x}_1\bar{x}_3 + x_2x_3 + x_1\bar{x}_2}_{\text{LHS}} \overset{?}{=} \underbrace{\bar{x}_1x_2 + x_1x_3 + \bar{x}_2\bar{x}_3}_{\text{RHS}}$$

f
1
0
1
1
1
1
0
1

f
1
0
1
1
1
1
0
1

Canonical forms

- Canonical forms
  - Standard forms for Boolean expressions
  - Derived from truth table
  - Generally not the simplest forms (can be minimized)
- Two canonical forms
  - Sum-of-products (minterms)
  - Product-of-sums (maxterms)

Sum-of-products (SOP)

- Also called *disjunctive normal form* (DNF) or *minterm expansion*

minterm

A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

$$F = \overset{001}{A'B'C} + \overset{011}{A'BC} + \overset{101}{AB'C} + \overset{110}{ABC'} + \overset{111}{ABC}$$

$$F' = A'B'C' + A'BC' + AB'C'$$

0= '

NOTICE 1 TO SOLVE

Minterms

- Variables appear exactly once in each minterm in true or inverted form (but not both)

A	B	C	minterms	
0	0	0	A'B'C'	m0
0	0	1	A'B'C	m1
0	1	0	A'BC'	m2
0	1	1	A'BC	m3
1	0	0	AB'C'	m4
1	0	1	AB'C	m5
1	1	0	ABC'	m6
1	1	1	ABC	m7

short-hand notation

F in canonical form:

$$\begin{aligned} F(A,B,C) &= \Sigma m(1,3,5,6,7) \\ &= m1 + m3 + m5 + m6 + m7 \\ &= A'B'C+A'BC+AB'C+ABC'+ABC \end{aligned}$$

Product-of-sums (POS)

- Also called *conjunctive normal form* (CNF) or *maxterm expansion*

A	B	C	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

$$F = \overset{000}{(A + B + C)}\ \overset{010}{(A + B' + C)}\ \overset{100}{(A' + B + C)}$$

maxterm

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

1 = '

NOTICE 0 TO SOLVE

Maxterms

- Variables appear exactly once in each maxterm in true or inverted form (but not both)

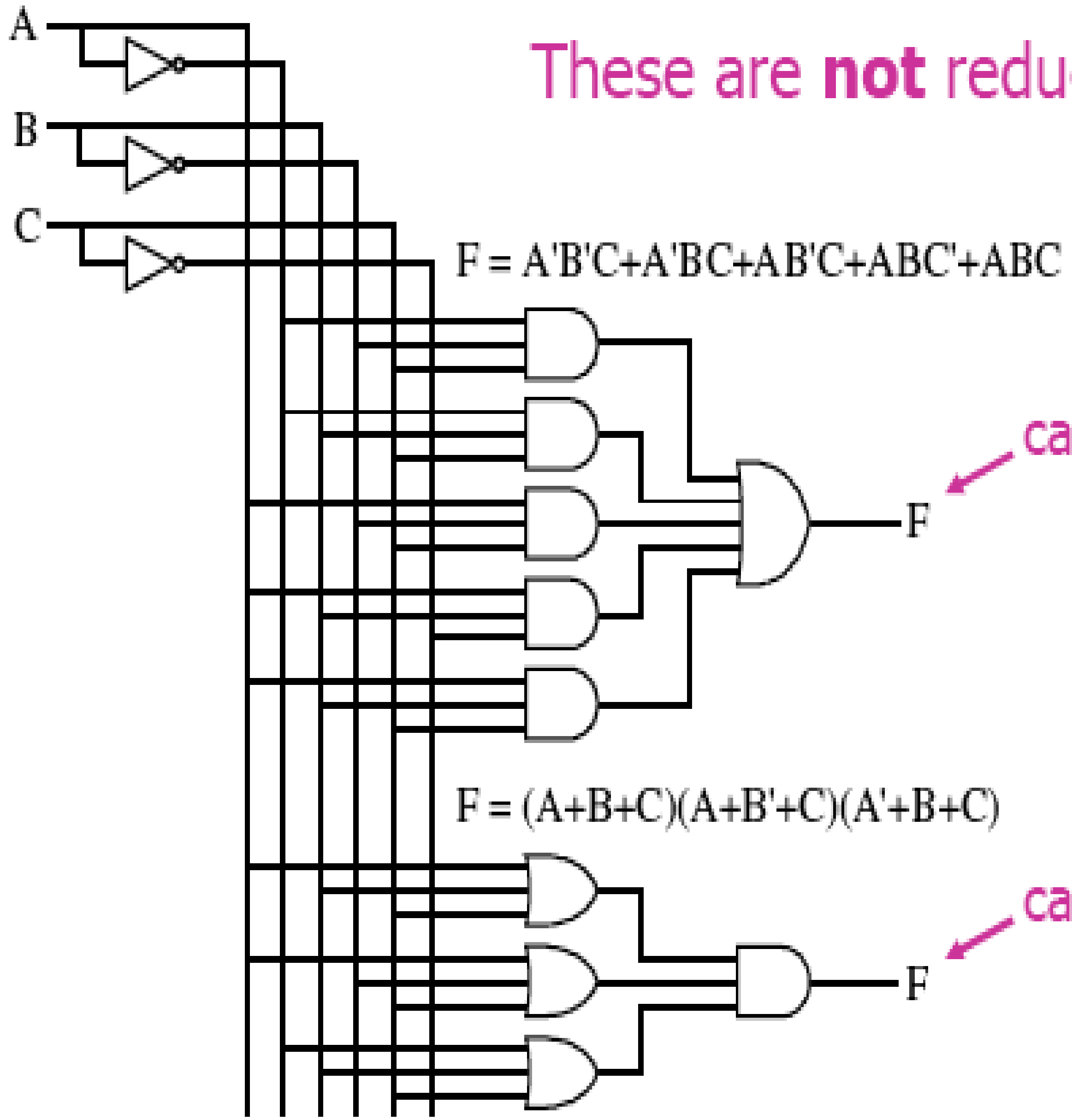
A	B	C	maxterms	
0	0	0	A+B+C	M0
0	0	1	A+B+C'	M1
0	1	0	A+B'+C	M2
0	1	1	A+B'+C'	M3
1	0	0	A'+B+C	M4
1	0	1	A'+B+C'	M5
1	1	0	A'+B'+C	M6
1	1	1	A'+B'+C'	M7

short-hand notation

F in canonical form:

$$\begin{aligned} F(A,B,C) &= \Pi M(0,2,4) \\ &= M0 \bullet M2 \bullet M4 \\ &= (A+B+C)(A+B'+C)(A'+B+C) \end{aligned}$$

These are **not** reduced forms for F



F = A'B'C+A'BC+AB'C+ABC'+ABC

canonical sum-of-products

F = (A+B+C)(A+B'+C)(A'+B+C)

canonical product-of-sums

Example

- Truth table for $f_1(a,b,c)$ at right

- The canonical sum-of-products form for f_1 is
  $f_1(a,b,c) = m_1 + m_2 + m_4 + m_6$
  $= a'b'c + a'bc' + ab'c' + abc'$

- The canonical product-of-sums form for f_1 is
  $f_1(a,b,c) = M_0 \bullet M_3 \bullet M_5 \bullet M_7$
  $= (a+b+c)\bullet(a+b'+c')\bullet$
  $(a'+b+c')\bullet(a'+b'+c')$.

a	b	c		f_1
0	0	0		0
0	0	1		1
0	1	0		1
0	1	1		0
1	0	0		1
1	0	1		0
1	1	0		1
1	1	1		0

Example 2

Design the minimum-cost **product-of-sums** and **sum-of-product** expression for the function

$f(x_1, x_2, x_3) = \Sigma\, m(0, 2, 4, 5, 6, 7)$

Minterms and Maxterms (with three variables)

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \overline{x}_1\overline{x}_2 x_3$	$M_1 = x_1 + x_2 + \overline{x}_3$
2	0	1	0	$m_2 = \overline{x}_1 x_2\overline{x}_3$	$M_2 = x_1 + \overline{x}_2 + x_3$
3	0	1	1	$m_3 = \overline{x}_1 x_2 x_3$	$M_3 = x_1 + \overline{x}_2 + \overline{x}_3$
4	1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$	$M_4 = \overline{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\overline{x}_2 x_3$	$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$
6	1	1	0	$m_6 = x_1 x_2\overline{x}_3$	$M_6 = \overline{x}_1 + \overline{x}_2 + x_3$
7	1	1	1	$m_7 = x_1 x_2 x_3$	$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$

Minterms and Maxterms (with three variables)

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \overline{x}_1\overline{x}_2x_3$	$M_1 = x_1 + x_2 + \overline{x}_3$
2	0	1	0	$m_2 = \overline{x}_1x_2\overline{x}_3$	$M_2 = x_1 + \overline{x}_2 + x_3$
3	0	1	1	$m_3 = \overline{x}_1x_2x_3$	$M_3 = x_1 + \overline{x}_2 + \overline{x}_3$
4	1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$	$M_4 = \overline{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\overline{x}_2x_3$	$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$
6	1	1	0	$m_6 = x_1x_2\overline{x}_3$	$M_6 = \overline{x}_1 + \overline{x}_2 + x_3$
7	1	1	1	$m_7 = x_1x_2x_3$	$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$

The function is 1 for these rows

Minterms and Maxterms (with three variables)

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \overline{x}_1\overline{x}_2 x_3$	$M_1 = x_1 + x_2 + \overline{x}_3$
2	0	1	0	$m_2 = \overline{x}_1 x_2\overline{x}_3$	$M_2 = x_1 + \overline{x}_2 + x_3$
3	0	1	1	$m_3 = \overline{x}_1 x_2 x_3$	$M_3 = x_1 + \overline{x}_2 + \overline{x}_3$
4	1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$	$M_4 = \overline{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\overline{x}_2 x_3$	$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$
6	1	1	0	$m_6 = x_1 x_2\overline{x}_3$	$M_6 = \overline{x}_1 + \overline{x}_2 + x_3$
7	1	1	1	$m_7 = x_1 x_2 x_3$	$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$

The function is 1 for these rows

The function is 0 for these rows

From SOP to POS and back

- Minterm to maxterm
  - Use maxterms that aren't in minterm expansion
  - $F(A,B,C) = \sum m(1,3,5,6,7) = \prod M(0,2,4)$
- Maxterm to minterm
  - Use minterms that aren't in maxterm expansion
  - $F(A,B,C) = \prod M(0,2,4) = \sum m(1,3,5,6,7)$

From SOP to POS and back

- Minterm of F to minterm of F'
  - Use minterms that don’t appear
  - $F(A,B,C) = \sum m(1,3,5,6,7)$ $F' = \sum m(0,2,4)$

- Maxterm of F to maxterm of F'
  - Use maxterms that don’t appear
  - $F(A,B,C) = \prod M(0,2,4) F' = \prod M(1,3,5,6,7)$

SOP, POS, and DeMorgan's

- Product-of-sums
  - F' = (A+B+C')(A+B'+C')(A'+B+C')(A'+B'+C')

- Apply DeMorgan's to get SOP

$$\overline{A + B} = \overline{A}.\overline{B} \text{ ....} (1)$$

$$\overline{A.B} = \overline{A} + \overline{B} \text{ ... ...} (2)$$

  - (F')' = ((A+B+C')(A+B'+C')(A'+B+C')(A'+B'+C'))'
  - F = A'B'C + A'BC + AB'C + ABC

SOP, POS, and DeMorgan's

- Sum-of-products
  - F' = A'B'C' + A'BC' + AB'C'

- Apply DeMorgan's to get POS

$$\overline{A + B} = \overline{A}\,.\,\overline{B} \ldots\ldots\ldots\ldots(1)$$

$$\overline{A\,.\,B} = \overline{A} + \overline{B} \ldots\ldots\ldots(2)$$

  - (F')' = (A'B'C' + A'BC' + AB'C')'
  - F = (A+B+C)(A+B'+C)(A'+B+C)

Conversion of SOP from standard to canonical form

- Expand *non-canonical* terms by inserting equivalent of 1 in each missing variable x: $(x + x') = 1$
- Remove duplicate minterms
- $f_1(a,b,c) = a'b'c + bc' + ac'$

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

Conversion of POS from standard to canonical form

- Expand noncanonical terms by adding 0 in terms of missing variables (*e.g.*, xx' = 0) and using the distributive law (e.g., A + (BC) = (A + B).(A + C))
- Remove duplicate maxterms
- $f_1(a,b,c) = (a+b+c)\bullet(b'+c')\bullet(a'+c')$

  $= (a+b+c)\bullet(aa'+b'+c')\bullet(a'+bb'+c')$

  $= (a+b+c)\bullet(a+b'+c')\bullet(a'+b'+c')\bullet(a'+b+c')\bullet(a'+b'+c')$

  $= (a+b+c)\bullet(a+b'+c')\bullet(a'+b'+c')\bullet(a'+b+c')$

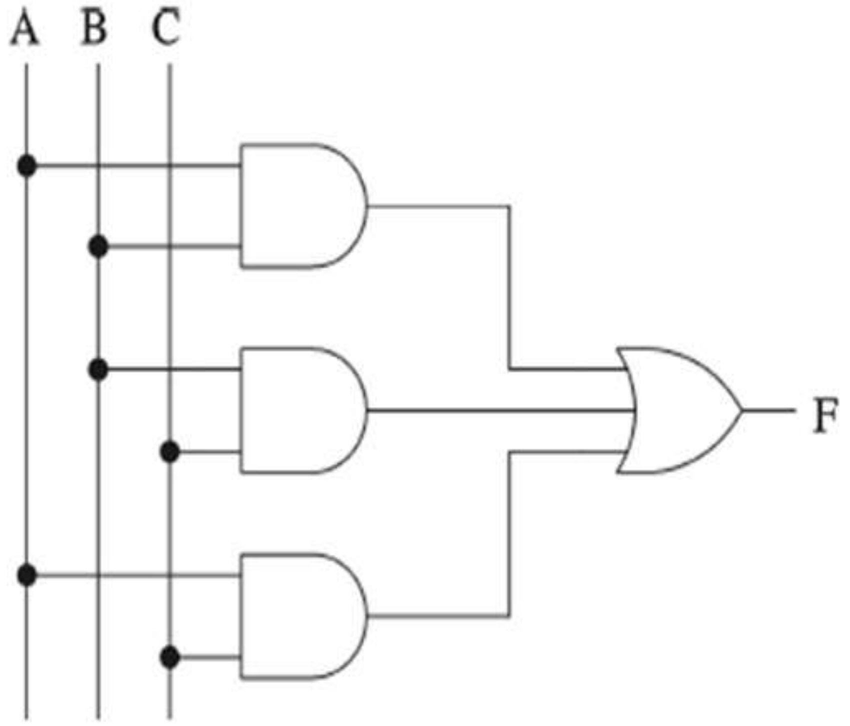
Exercise

3-input majority function

A	B	C	F
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

- Logical expression form

$$F = A\,B + B\,C + A\,C$$



- Majority function example

Added extra

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

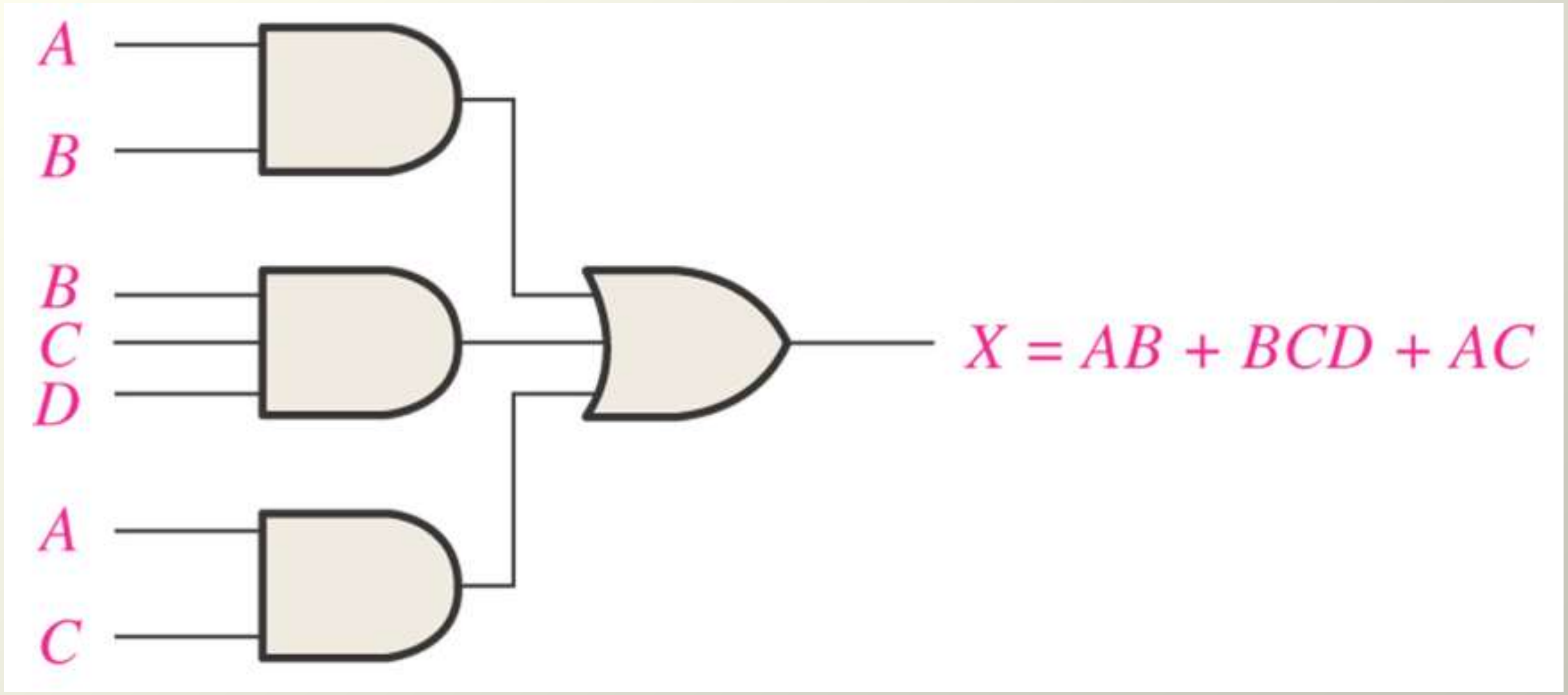
$$\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC + ABC + ABC$$

- We can now simplify this expression as

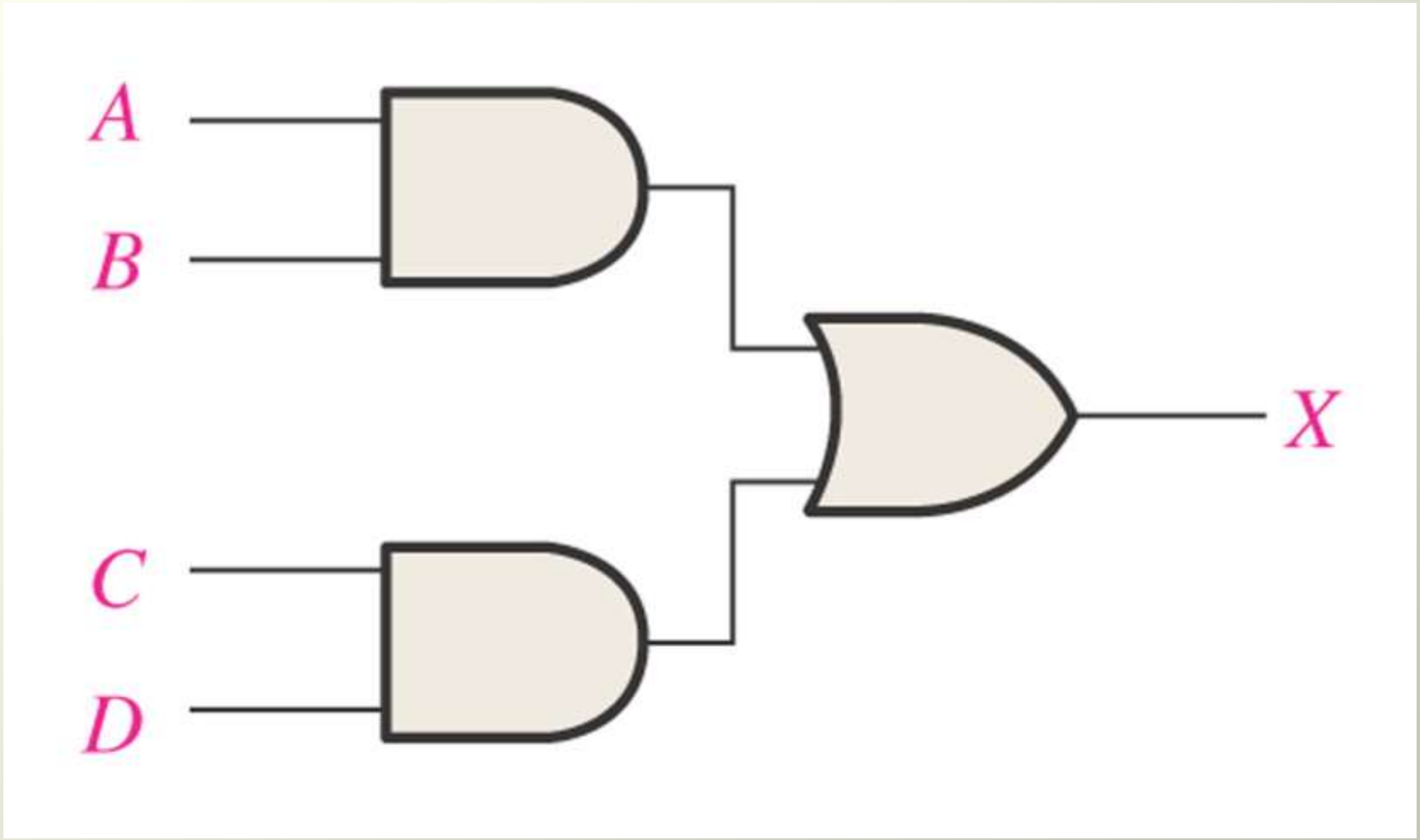
$$BC + AC + AB$$

- A difficult method to use for complex expressions

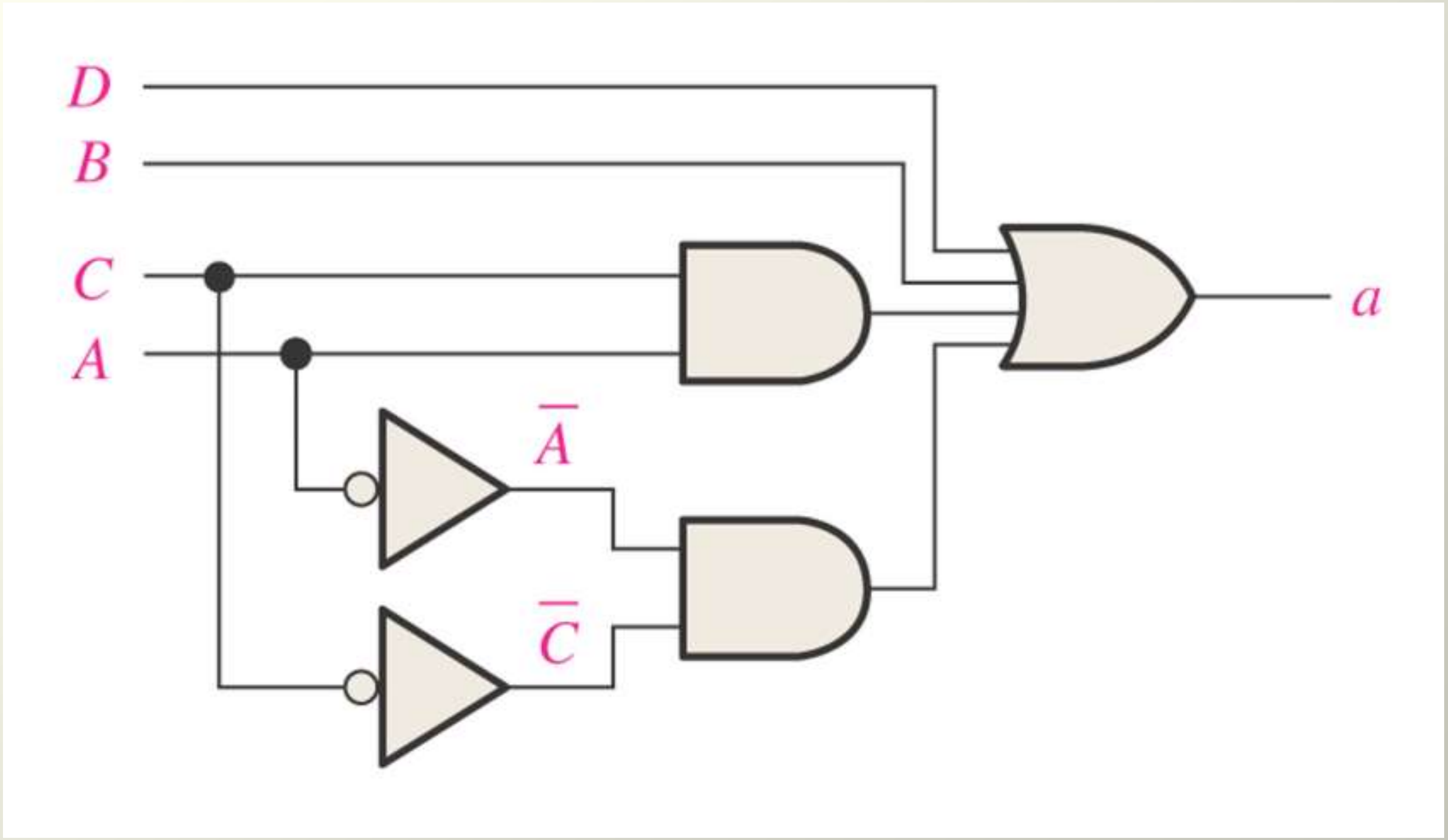
Implementation of the SOP expression *AB* + *BCD* + *AC*.



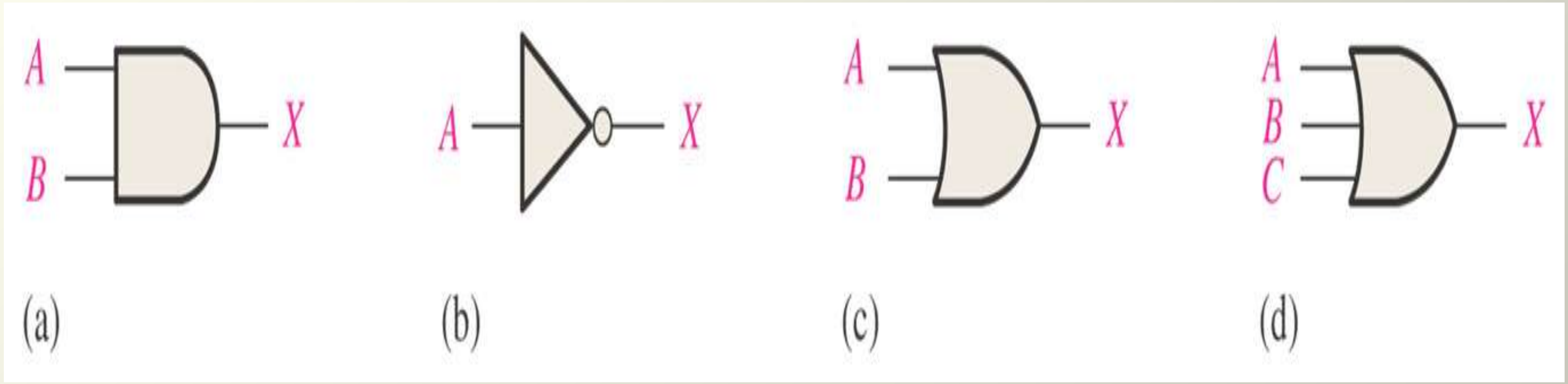
Exercise: The logic implementation for segment X.



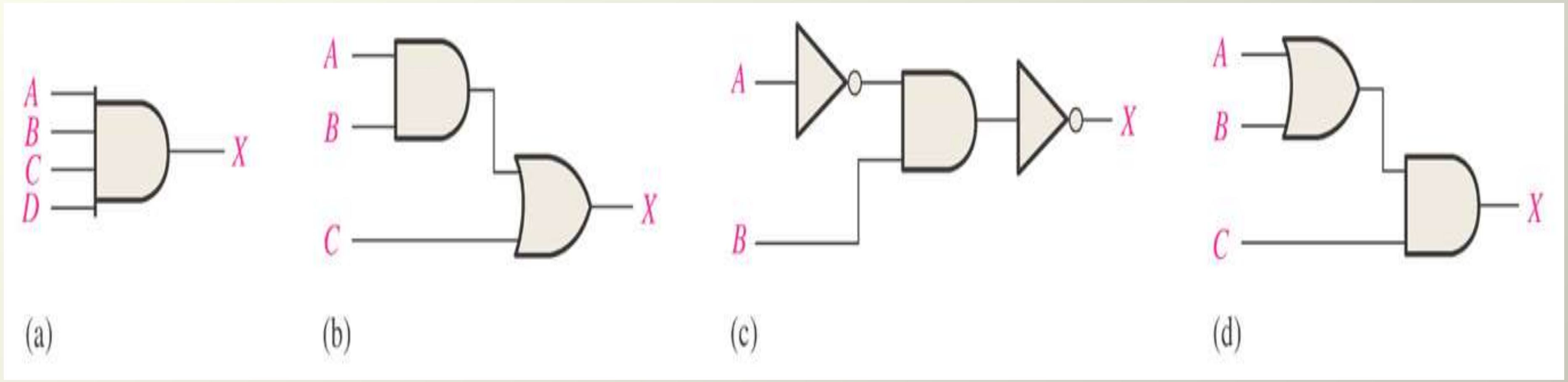
Exercise: The logic implementation for segment a.



Exercise: What is the Boolean expression for each of the logic gates?



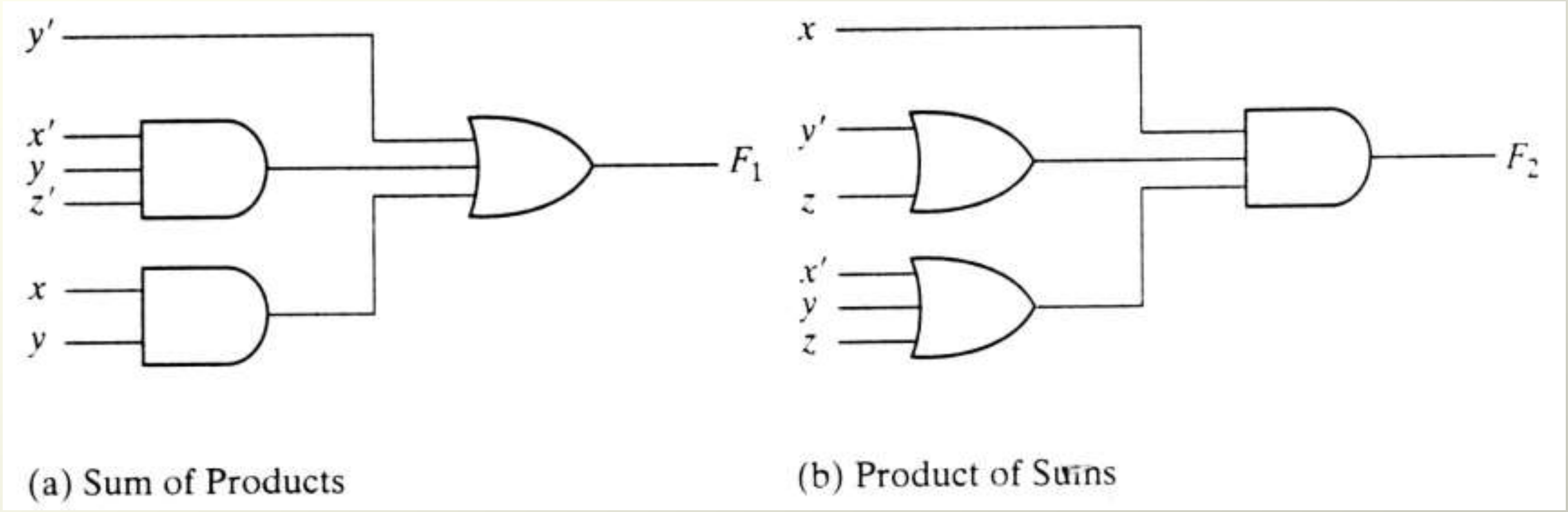
Exercise: What is the Boolean expression for each of the logic gates?



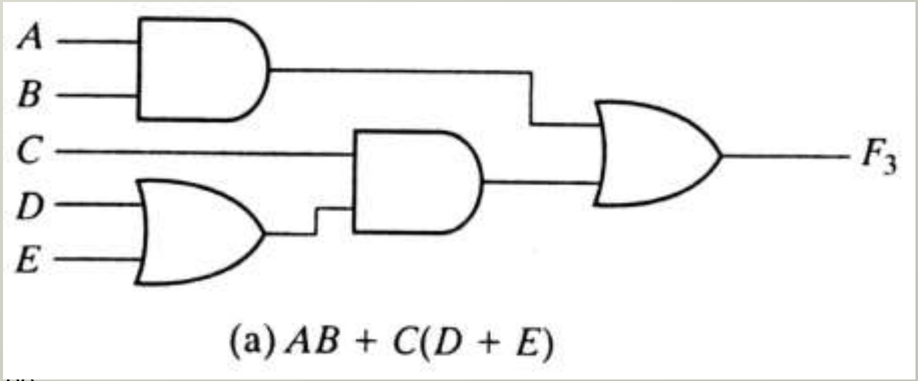
- Example: Prove

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

- Two-level implementation



Multi-level implementation



- Multiple NOR = a complement of OR gate
- Multiple NAND = a complement of AND
- The cascaded NAND operations = sum of products
- The cascaded NOR operations = product of sums

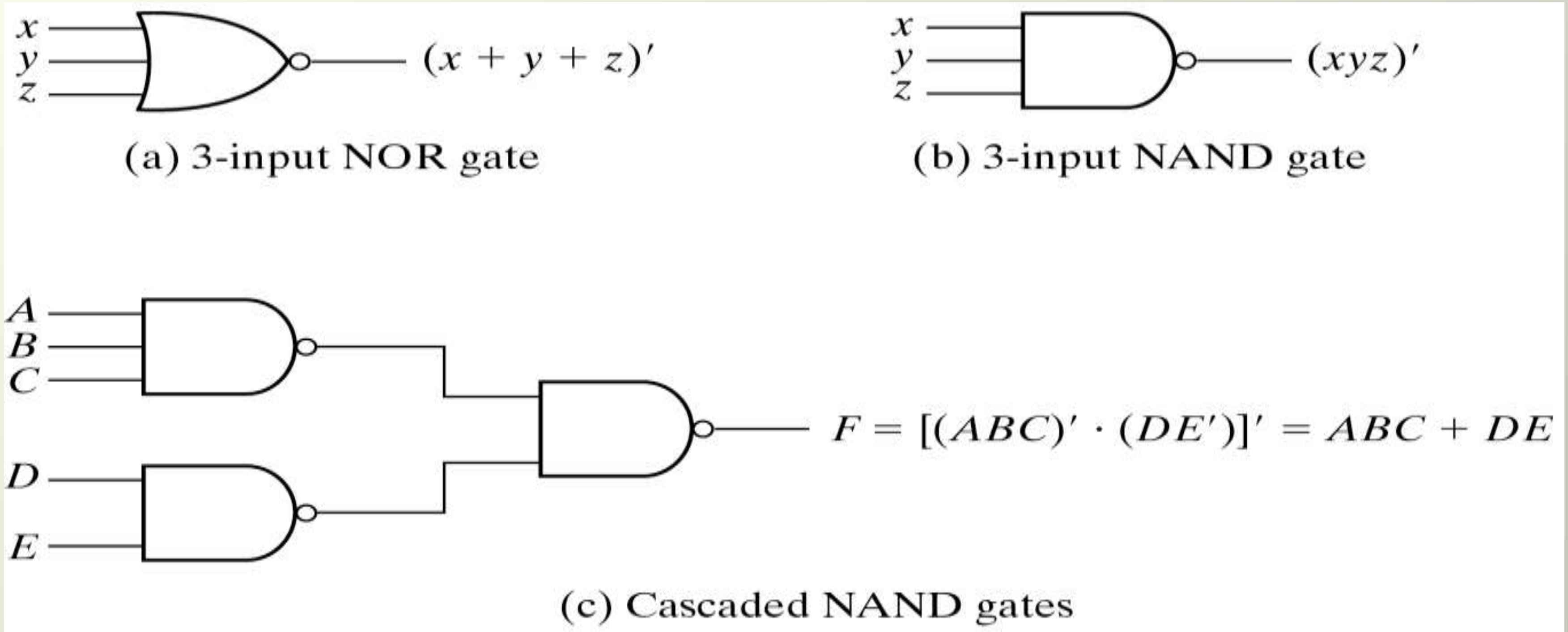
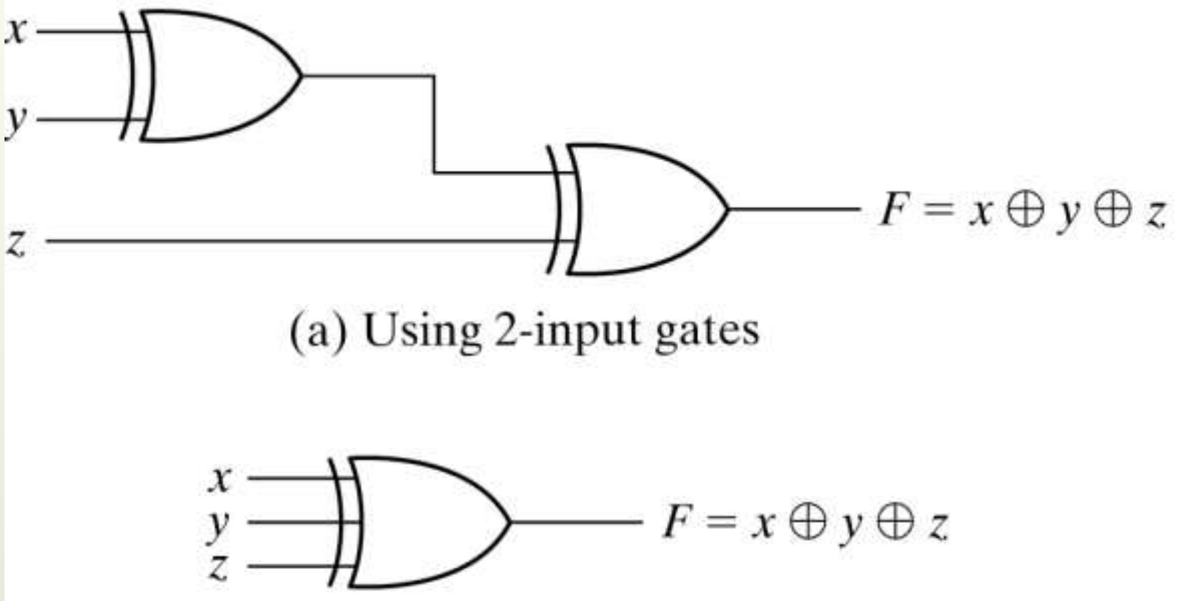


Fig. 2-7 Multiple-input and cascated NOR and NAND gates

- The XOR and XNOR gates are commutative and associative
- Multiple-input XOR gates are uncommon.
- XOR is an odd function: it is equal to 1 if the inputs variables have an odd number of 1's



(a) Using 2-input gates

(b) 3-input gate

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(c) Truth table

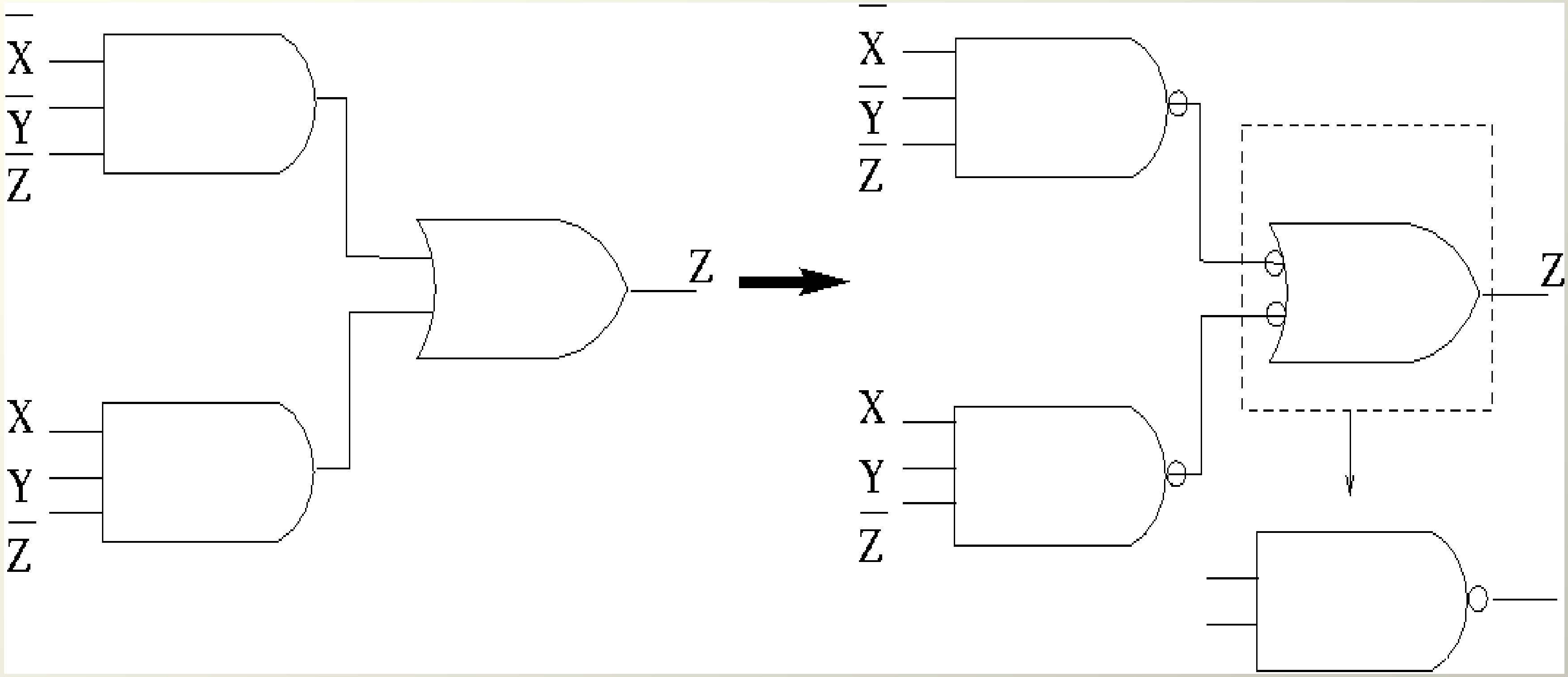
Fig. 2-8 3-input exclusive-OR gate

Two-Level NAND Gate Implementation - Example

$F(X,Y,Z) = \Sigma m(0,6)$

1. Express F in SOP form:
   $F = X'Y'Z' + XYZ'$
2. Obtain the AND-OR implementation for F.
3. Add bubbles and inverters to transform AND-OR to NAND-NAND gates.

Example (cont.)



Two-level implementation with NANDs

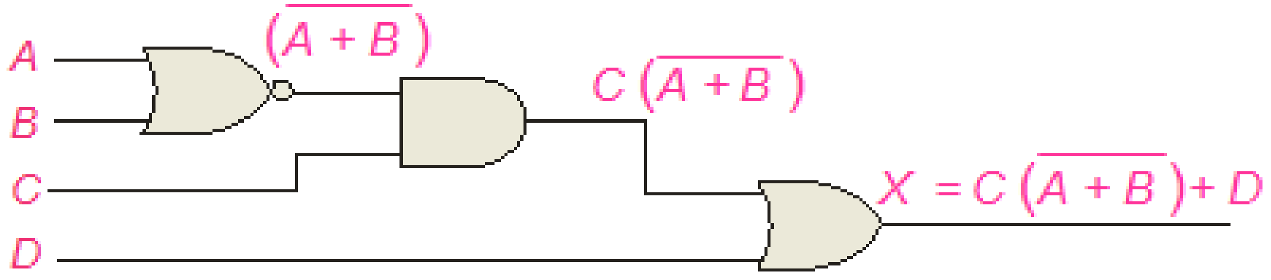
F = X’Y’Z’ + XYZ’

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.

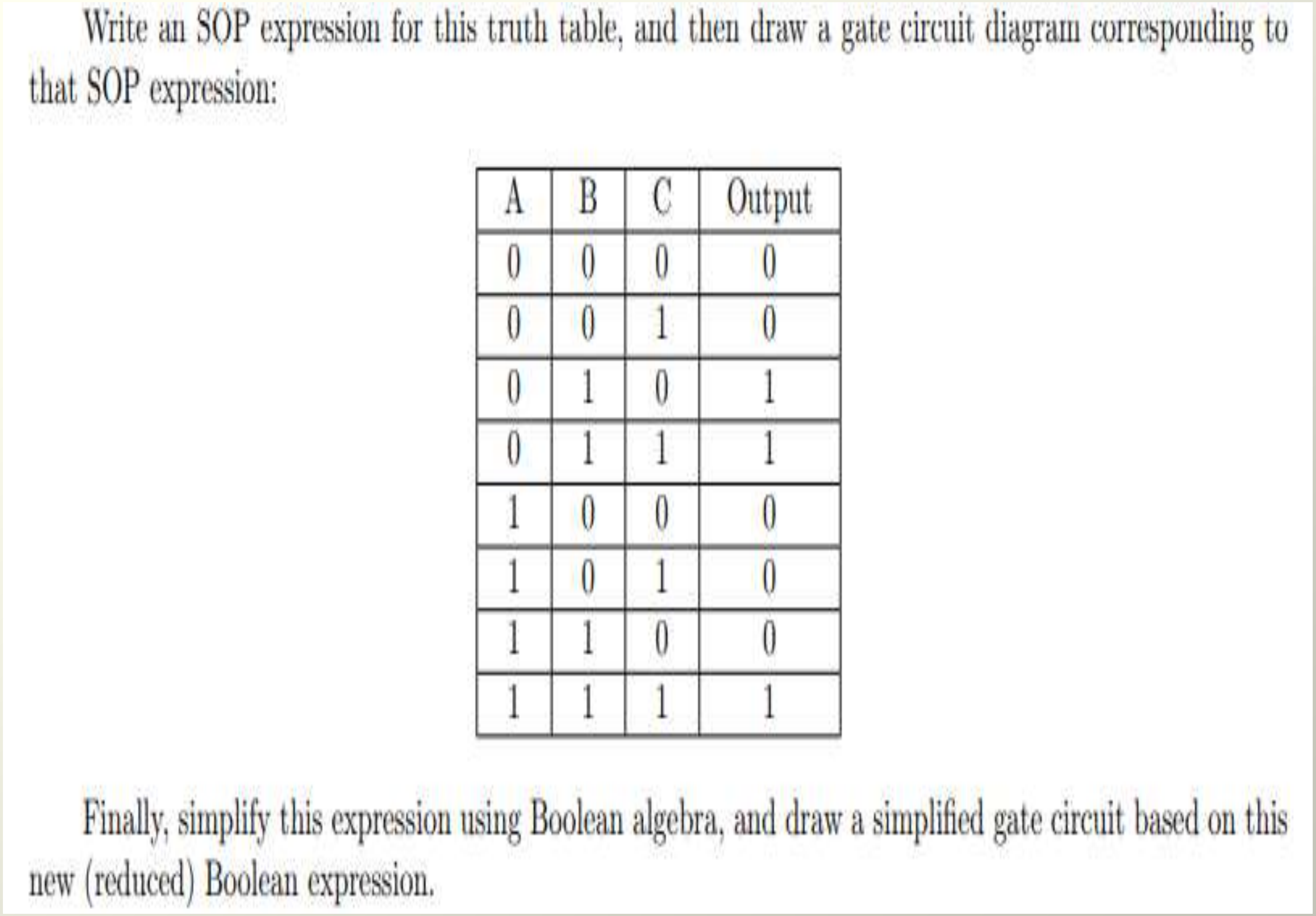
Example Apply Boolean algebra to derive the expression for *X*.

Solution Write the expression for each gate:



Applying DeMorgan's theorem and the distribution law:

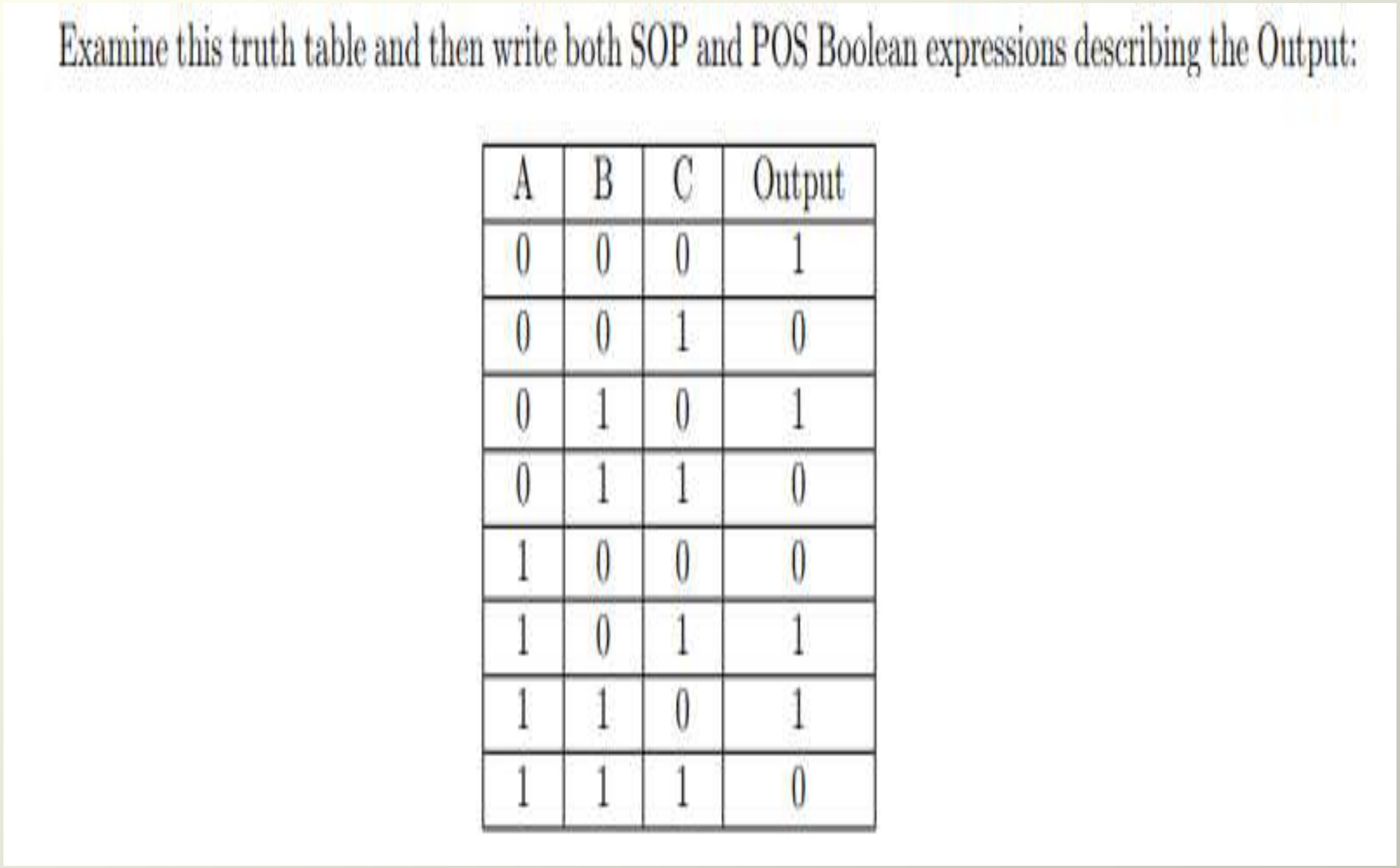
$$X = C(\overline{A}\ \overline{B}) + D = \overline{A}\,\overline{B}\,C + D$$



Write an SOP expression for this truth table, and then draw a gate circuit diagram corresponding to that SOP expression:

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Finally, simplify this expression using Boolean algebra, and draw a simplified gate circuit based on this new (reduced) Boolean expression.


Examine this truth table and then write both SOP and POS Boolean expressions describing the Output:

A	B	C	Output
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Quiz

1. 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. $A + AB = A$

2. 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

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

   a. A

   b. B

   c. 0

   d. 1

4. The Boolean expression $A + 1$ is equal to

   a. A

   b. B

   c. 0

   d. 1

Quiz

5. 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

6. 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

Answers:

1. b	6. c
2. c	
3. a	
4. d	
5. a	

The end