Overview

- Gate Circuits and Boolean Equations
  - Binary Logic and Gates
  - Boolean Algebra
  - Standard Forms
- Additional Gates and Circuits
  - Other Gate Types
  - Exclusive-OR Operator and Gates
  - High-Impedance Outputs

Binary Logic and Gates

- Digital circuits are hardware components (based on transistors) that manipulate binary information
- We model the transistor-based electronic circuits as logic gates.

Binary Logic

- Binary variables take on one of two values.
- Logical operators operate on binary values and binary variables.
- Basic logical operators are the logic functions AND, OR and NOT.
- Logic gates implement logic functions.
- Boolean Algebra: a useful mathematical system for specifying and transforming logic functions.

Binary Variables

- Recall that the two binary values have different names:
  - True/False
  - On/Off
  - Yes/No
  - 1/0
- We use 1 and 0 to denote the two values.

- Computers are implementations of Boolean logic.
- Boolean functions are completely described by truth tables.
- Logic gates are small circuits that implement Boolean operators.
- The basic gates are AND, OR, and NOT.
    - The XOR gate is very useful in parity checkers and adders.
- The “universal gates” are NOR, and NAND.

Logical Operations

- The three basic logical operations are:
  - AND
  - OR
  - NOT
- AND is denoted by a dot (·).
- OR is denoted by a plus (+).
- NOT is denoted by an overbar (¯), a single quote mark (') after, or (~) before the variable.

Notation Examples

- **Examples:**
  - $Y = A \times B$ is read "Y is equal to **A AND B.**"
  - $z = x + y$ is read "z is equal to **x OR y.**"
  - $X = \overline{A}$ is read "X is equal to **NOT A.**"

- **Note: The statement:**

  1 + 1 = 2 (read "one <u>plus</u> one equals two")

  is not the same as

  1 + 1 = 1 (read "1 <u>or</u> 1 equals 1").

Operator Definitions

- **Operations are defined on the values "0" and "1" for each operator:**

AND	OR	NOT
$0 \cdot 0 = 0$	$0 + 0 = 0$	$\overline{0} = 1$
$0 \cdot 1 = 0$	$0 + 1 = 1$	$\overline{1} = 0$
$1 \cdot 0 = 0$	$1 + 0 = 1$	
$1 \cdot 1 = 1$	$1 + 1 = 1$	

Truth Tables

- ***Truth table*** **-** a tabular listing of the values of a function **for all possible combinations** of values on its arguments
- Example: Truth tables for the basic logic operations:

AND		
X	**Y**	**$Z = X \cdot Y$**
0	0	0
0	1	0
1	0	0
1	1	1

OR		
X	**Y**	**$Z = X + Y$**
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
X	**$Z = \overline{X}$**
0	1
1	0

Basic Theorems

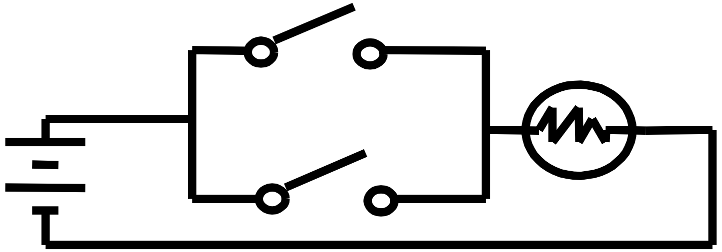
Table 2.1
Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

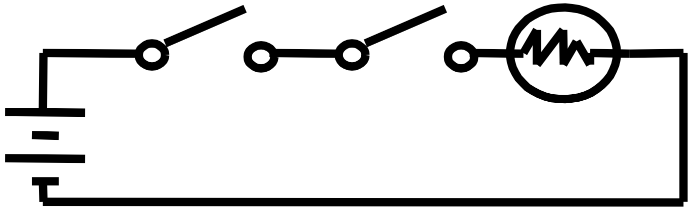
Logic Function Implementation

- **Using Switches**
  - **For inputs:**
    - **logic 1 is <u>switch closed</u>**
    - **logic 0 is <u>switch open</u>**
  - **For outputs:**
    - **logic 1 is <u>light on</u>**
    - **logic 0 is <u>light off</u>.**
  - **NOT uses a switch such that:**
    - **logic 1 is <u>switch open</u>**
    - **logic 0 is <u>switch closed</u>**

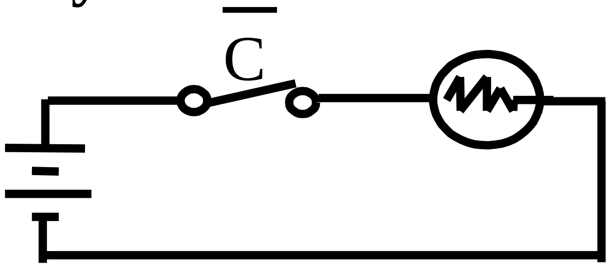
Switches in parallel => OR



Switches in series => AND



Normally-closed switch => NOT



Review of Boolean algebra

- **Not** is a horizontal bar above the number
  - $\overline{0} = 1$
  - $\overline{1} = 0$
- **Or** is a plus
  - $0+0 = 0$
  - $0+1 = 1$
  - $1+0 = 1$
  - $1+1 = 1$
- **And** is multiplication
  - $0*0 = 0$
  - $0*1 = 0$
  - $1*0 = 0$
  - $1*1 = 1$

Logic Gate

Logic Gate

A gate is an digital circuit which operates on one or more signals and produce single output.

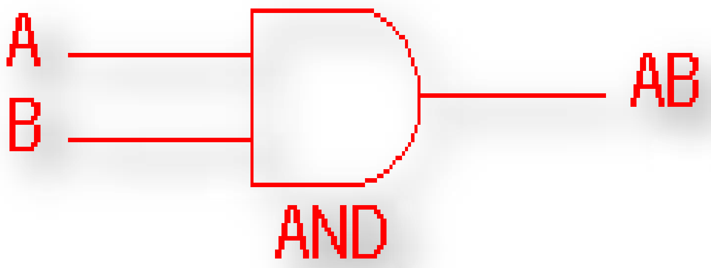
Gates are digital circuits because the input and output signals are denoted by either 1(high voltage) or 0(low voltage).

There are three basic gates and are:

1. AND gate
2. OR gate
3. NOT gate

AND gate

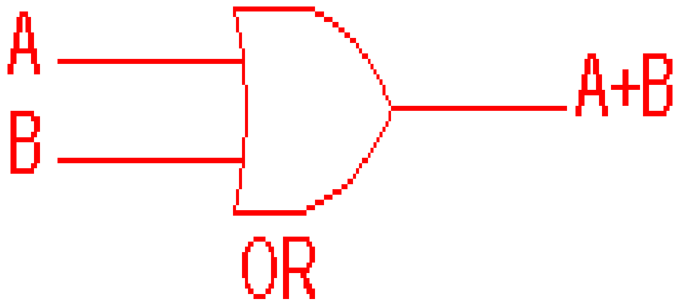
- The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high.
- AND gate takes two or more input signals and produce only one output signal.



Input A	Input B	Output AB
0	0	0
0	1	0
1	0	0
1	1	1

OR gate

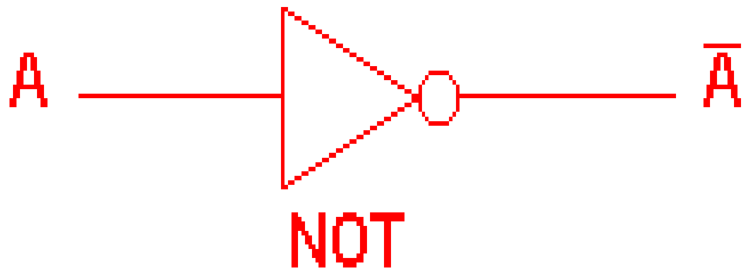
- The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high.
- OR gate also takes two or more input signals and produce only one output signal.



Input A	Input B	Output A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT gate

- The NOT gate is an electronic circuit that gives a high output (1) if its input is low .
- NOT gate takes only one input signal and produce only one output signal.
- The output of NOT gate is complement of its input.
- It is also called inverter.



Input A	Output A
0	1
1	0

DeMorgan's Theorem

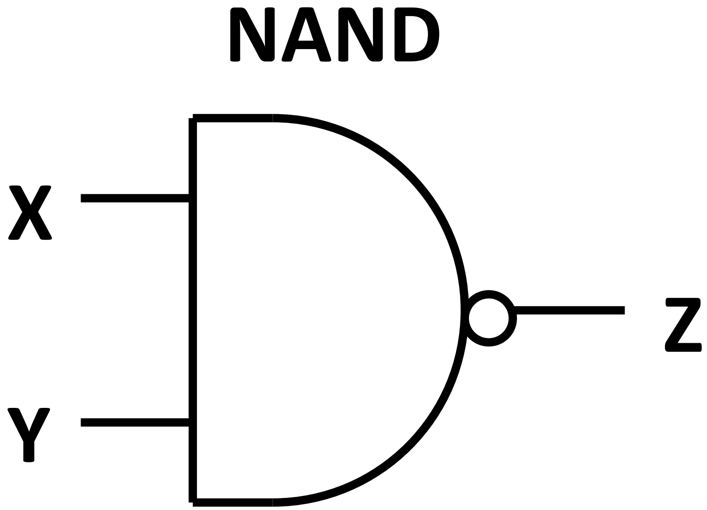
- $(x + y)' = x'y'$,$(xy)' = x' + y'$
- By means of truth table

x	y	x'	y'	$x+y$	$(x+y)'$	$x'y'$	xy	$x'+y'$	$(xy)'$
0	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	0	0	1	1
1	0	0	1	1	0	0	0	1	1
1	1	0	0	1	0	0	1	0	0

NAND Gate

Known as a “universal” gate because ANY digital circuit can be implemented with NAND gates alone.

NAND Gate

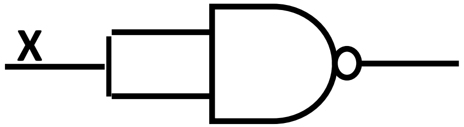


X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

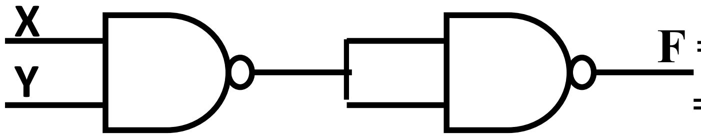
Z = ~(X & Y)

Nand(Z,X,Y)

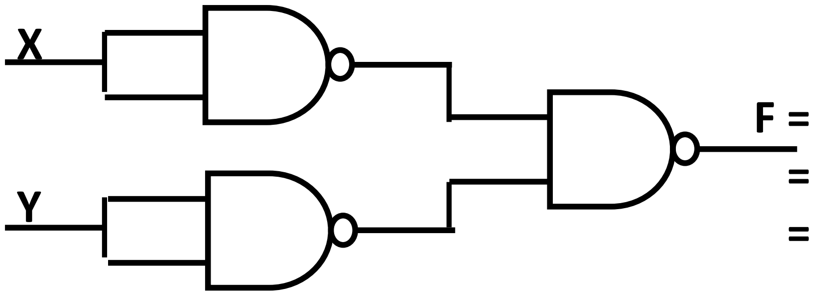
NAND Gate



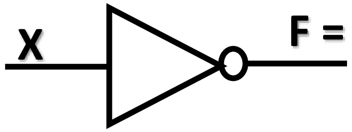
F = (X•X)’
= X’+X’
= X’

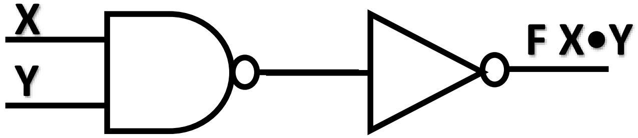


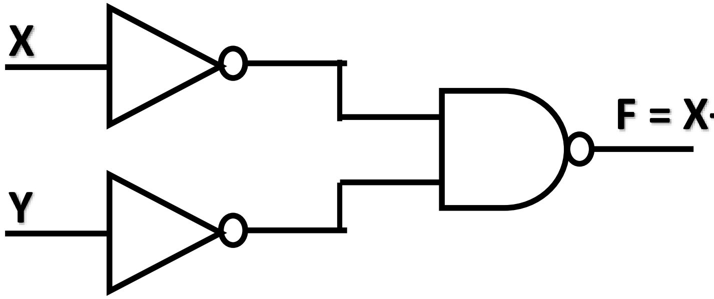
F = ((X•Y)’)’
= (X’+Y’)’
= X”•Y”
= X•Y



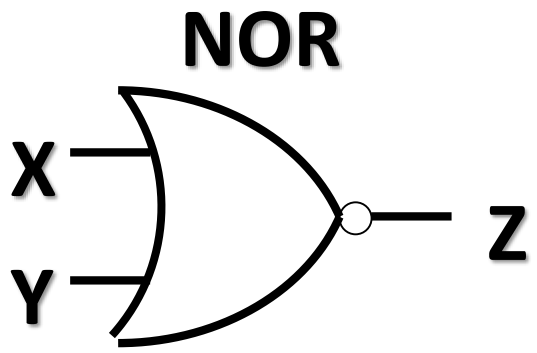
F = (X’•Y’)’
= X’’+Y’’
= X+Y







NOR Gate

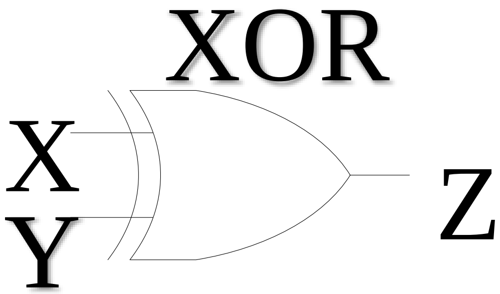


Z = ~(X | Y)

nor(Z,X,Y)

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

Exclusive-OR Gate



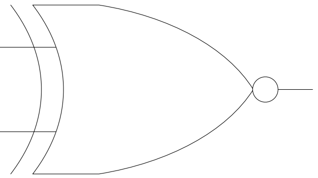
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

$Z = X \wedge Y$

xor(Z,X,Y)

Exclusive-NOR Gate

XNOR

X
Y

Z

$Z = \sim (X \wedge Y)$

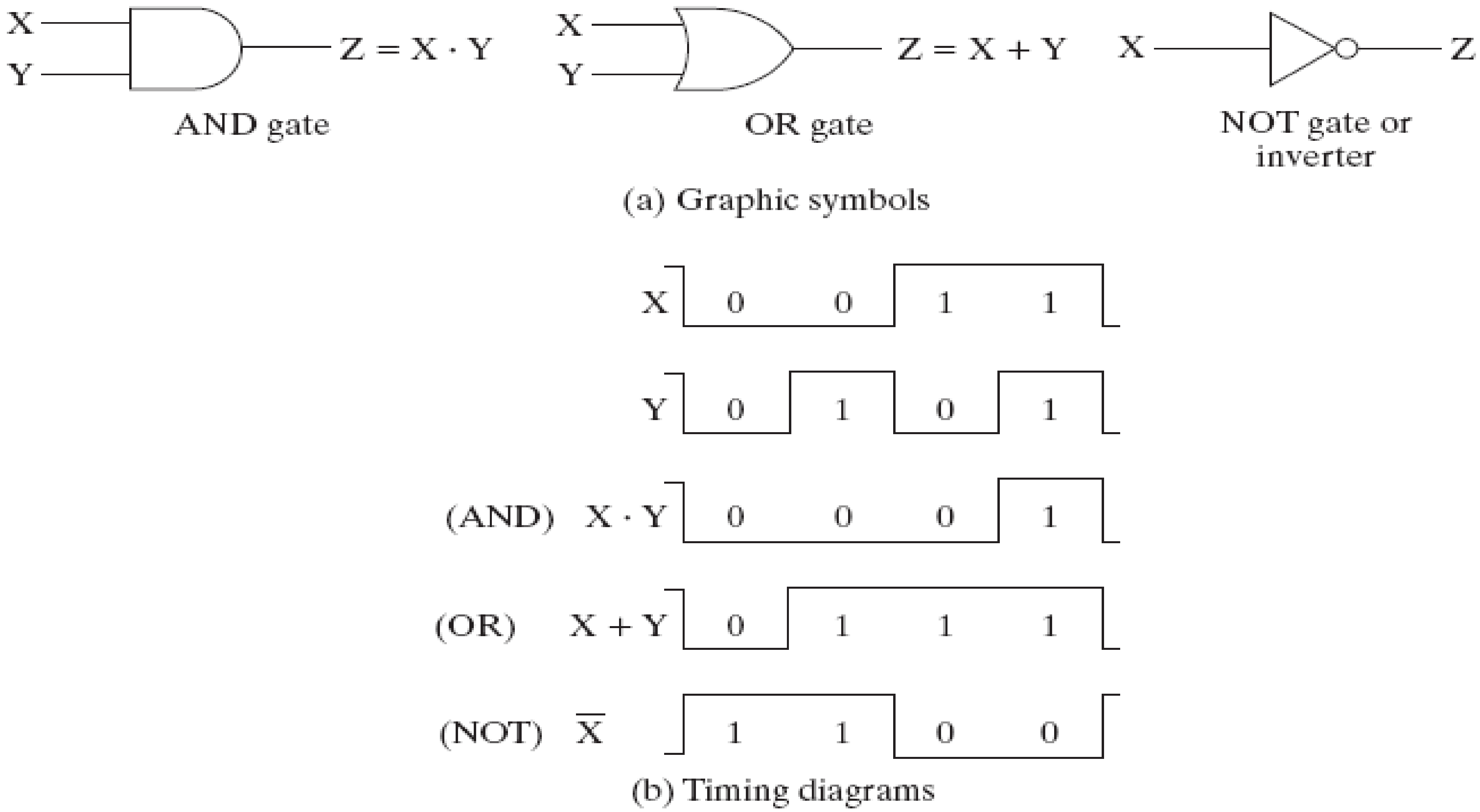
$Z = X \sim\wedge Y$

Xnor (Z,X,Y)

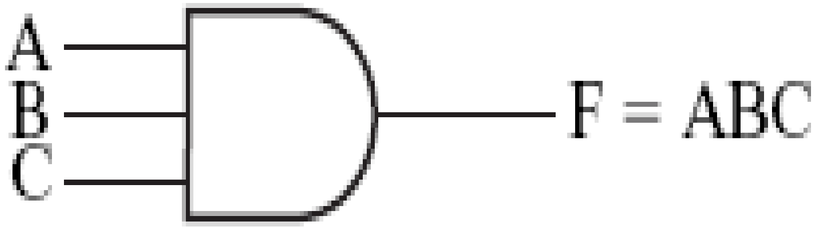
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

Logic Gate Symbols and Behavior

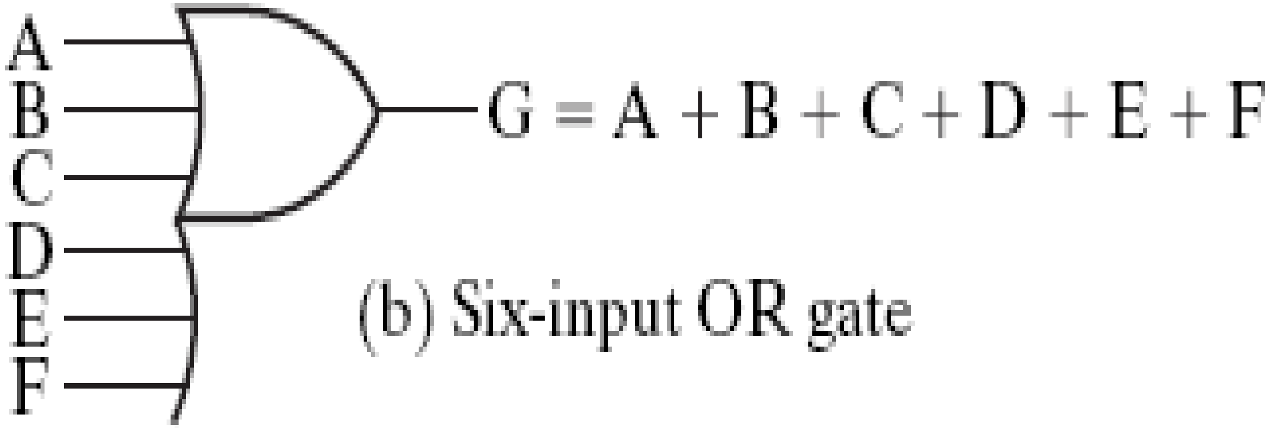
- **Logic gates have special symbols:**



AND and OR gates with more than two inputs



(a) Three-input AND gate



(b) Six-input OR gate

Boolean Algebra

- Boolean expression: a expression formed by binary variables, for example,

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

- Boolean function: a binary variable identifying the function followed by an equal sign and a Boolean expression for example

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

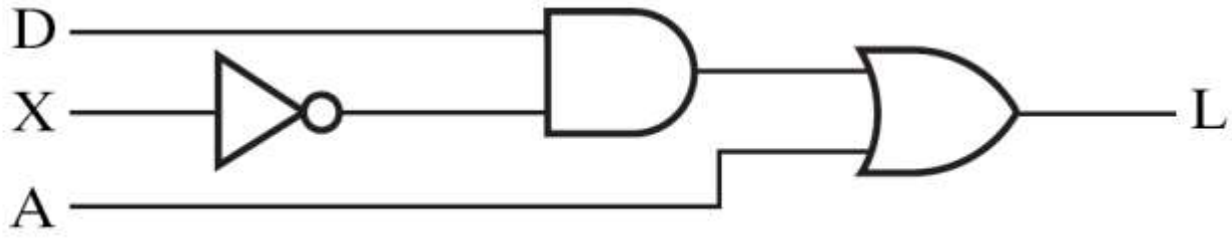
Truth table and Logic circuit

For the Boolean function

$$L(D,\ X,\ A) = D\bar{X} + A$$

Truth Table
for the Function $L = D\overline{X} + A$

D	X	A	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



Logic Circuit Diagram

Basic identities of Boolean Algebra

- An algebraic structure defined on a set of at least two elements together with three binary operators (denoted +, · and -) that satisfies the following basic identities:

1.	$X + 0 = X$	2.	$X \cdot 1 = X$	
3.	$X + 1 = 1$	4.	$X \cdot 0 = 0$	
5.	$X + X = X$	6.	$X \cdot X = X$	
7.	$X + \overline{X} = 1$	8.	$X \cdot \overline{X} = 0$	
9.	$\overline{\overline{X}} = X$			
10.	$X + Y = Y + X$	11.	$XY = YX$	Commutative
12.	$(X + Y) + Z = X + (Y + Z)$	13.	$(XY)Z = X(YZ)$	Associative
14.	$X(Y + Z) = XY + XZ$	15.	$X + YZ = (X + Y)(X + Z)$	Distributive
16.	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	17.	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$	DeMorgan's

Truth Table to Verify DeMorgan's Theorem

Truth Tables to Verify DeMorgan's Theorem

A)	X	Y	$X+Y$	$\overline{X+Y}$	B)	X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X}\cdot\overline{Y}$
	0	0	0	1		0	0	1	1	1
	0	1	1	0		0	1	1	0	0
	1	0	1	0		1	0	0	1	0
	1	1	1	0		1	1	0	0	0

Extension of DeMorgan's Theorem:

$$\overline{X_1 + X_2 + \cdots + X_n} = \overline{X}_1 \overline{X}_2 \cdots \overline{X}_n$$

Boolean Operator Precedence

- The order of evaluation in a Boolean expression is:

1. Parentheses
2. NOT
3. AND
4. OR

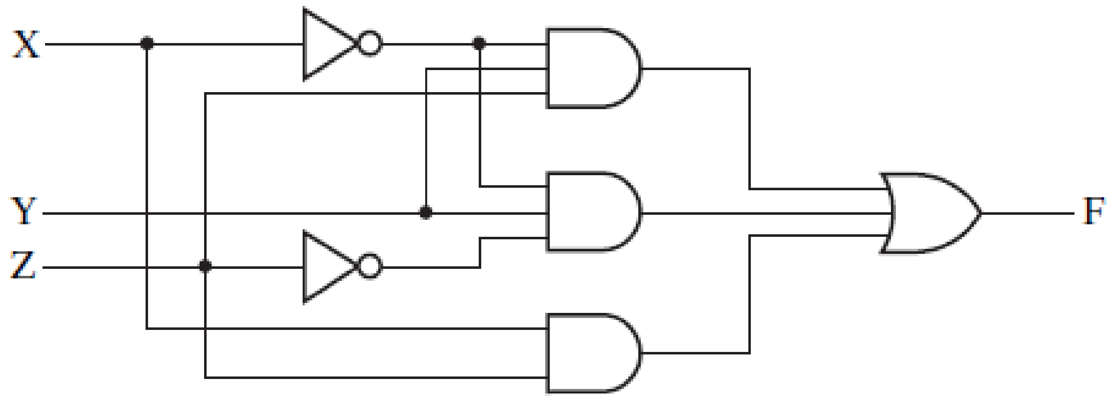
- **Example:** $F = A\,(B + C)\,(C + D)$

Boolean Algebraic Manipulation

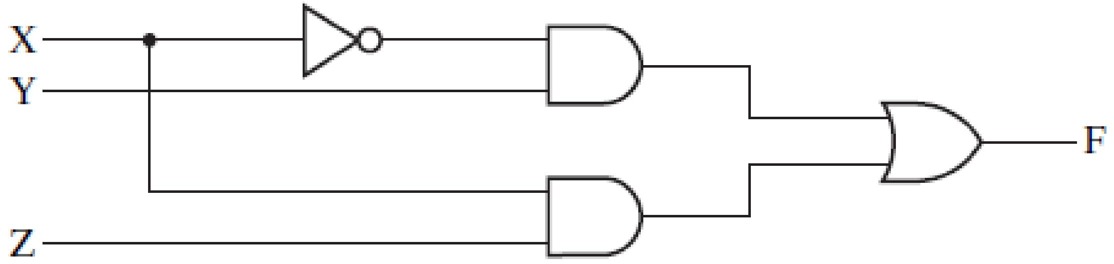
$$F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$$

$$= \overline{X}Y(Z + \overline{Z}) + XZ$$

$$= \overline{X}Y + XZ$$



(a) $F = \overline{X}YZ + \overline{X}Y\overline{Z} + XZ$



(b) $F = \overline{X}Y + XZ$

Truth Table for Boolean Function

X	Y	Z	(a) F	(b) F
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

Boolean Algebraic Manipulation

- $AB + \overline{A}C + BC = AB + \overline{A}C$ (Consensus Theorem)

Proof Steps Justification (identity or theorem)

$AB + \overline{A}C + BC$

$= AB + \overline{A}C + 1 \cdot BC$

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

$=$

Example: Complementing Function

$$F_1 = \overline{X}Y\overline{Z} + \overline{X}\,\overline{Y}Z$$

$$F_2 = X(\overline{Y}\,\overline{Z} + YZ)$$

$$\overline{F_1} = ?$$

$$\overline{F_2} = ?$$

- By DeMorgan's Theorem (Example 2-2)
- By duality (Example 2-3)

Canonical Forms

- It is useful to specify Boolean functions in a form that:
  - Allows comparison for equality.
  - Has a correspondence to the truth tables
- Canonical Forms in common usage:
  - Sum of Minterms (SOM)
  - Product of Maxterms (POM)

Maxterms and Minterms

- **Examples: Two variable minterms and maxterms.**

Index	Minterm	Maxterm
0	$\bar{x}\,\bar{y}$	$x + y$
1	$\bar{x}\,y$	$x + \bar{y}$
2	$x\,\bar{y}$	$\bar{x} + y$
3	$x\,y$	$\bar{x} + \bar{y}$

- **The index above is important for describing which variables in the terms are true and which are complemented.**

Minterms

- Minterms are AND terms with every variable present in either true or complemented form.
- Given that each binary variable may appear normal (e.g., x) or complemented (e.g., $\overline{\mathbf{x}}$), there are 2^n minterms for n variables.
- Example: Two variables (X and Y)produce
  2 x 2 = 4 combinations:

XY (both normal)

$X\overline{Y}$ (X normal, Y complemented)

$\overline{X}Y$ (X complemented, Y normal)

$\overline{X}\,\overline{Y}$ (both complemented)

Thus there are <u>four minterms</u> of two variables.

Maxterms

- Maxterms are OR terms with every variable in true or complemented form.
- Given that each binary variable may appear normal (e.g., x) or complemented (e.g., $\bar{x}$), there are 2^n maxterms for n variables.
- Example: Two variables (X and Y) produce 2 x 2 = 4 combinations:

(both normal) $\mathbf{X + Y}$

(x normal, y complemented) $\mathbf{X + \overline{Y}}$

(x complemented, y normal) $\mathbf{\overline{X} + Y}$

(both complemented) $\mathbf{\overline{X} + \overline{Y}}$

Minterms for three variables

Minterms for Three Variables

X	Y	Z	Product Term	Symbol	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$	m_0	1	0	0	0	0	0	0	0
0	0	1	$\overline{X}\,\overline{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\overline{X}Y\overline{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\overline{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\overline{Y}\,\overline{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\overline{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\overline{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

Maxterms for three variables

Maxterms for Three Variables

X	Y	Z	Sum Term	Symbol	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	$X+Y+Z$	M_0	0	1	1	1	1	1	1	1
0	0	1	$X+Y+\overline{Z}$	M_1	1	0	1	1	1	1	1	1
0	1	0	$X+\overline{Y}+Z$	M_2	1	1	0	1	1	1	1	1
0	1	1	$X+\overline{Y}+\overline{Z}$	M_3	1	1	1	0	1	1	1	1
1	0	0	$\overline{X}+Y+Z$	M_4	1	1	1	1	0	1	1	1
1	0	1	$\overline{X}+Y+\overline{Z}$	M_5	1	1	1	1	1	0	1	1
1	1	0	$\overline{X}+\overline{Y}+Z$	M_6	1	1	1	1	1	1	0	1
1	1	1	$\overline{X}+\overline{Y}+\overline{Z}$	M_7	1	1	1	1	1	1	1	0

Minterm and Maxterm Relationship

- Review: DeMorgan's Theorem

$$\overline{x \cdot y} = \bar{x} + \bar{y} \text{ and } \overline{x + y} = \bar{x} \times \bar{y}$$

- Two-variable example:

$$M_2 = \bar{x} + y \text{ and } m_2 = x \cdot \bar{y}$$

Thus M_2 is the complement of m_2 and vice-versa.

- Since DeMorgan's Theorem holds for *n* variables, the above holds for terms of *n* variables
- giving:

$$M_i = \overline{m_i} \text{ and } m_i = \overline{M_i}$$

Thus M_i is the complement of m_i.

Function Tables for Both

- **Minterms of 2 variables** **Maxterms of 2 variables**

x y	m_0	m_1	m_2	m_3
0 0	1	0	0	0
0 1	0	1	0	0
1 0	0	0	1	0
1 1	0	0	0	1

x y	M_0	M_1	M_2	M_3
0 0	0	1	1	1
0 1	1	0	1	1
1 0	1	1	0	1
1 1	1	1	1	0

- **Each column in the maxterm function table is the complement of the column in the minterm function table since M_i is the complement of m_i.**

Observations

- **In the function tables:**
  - **Each minterm has one and only one 1 present in the 2^n terms (a minimum of 1s). All other entries are 0.**
  - **Each maxterm has one and only one 0 present in the 2^n terms All other entries are 1 (a maximum of 1s).**
- **We can implement any function by "ORing" the minterms corresponding to "1" entries in the function table. These are called the minterms of the function.**

We can implement any function by "ANDing" the maxterms corresponding to "0" entries in the function table. These are called the maxterms of the function.

This gives us two <u>canonical forms</u>:

<u>Sum of Minterms (SOM)</u>

<u>Product of Maxterms (POM)</u>

for stating any Boolean function.

Conversion of Minterm and Maxterm

$$F = \overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}Y\overline{Z} + X\overline{Y}Z + XYZ = m_0 + m_2 + m_5 + m_7 = \sum m(0, 2, 5, 7)$$

$$\overline{F} = \overline{X}\,\overline{Y}Z + \overline{X}YZ + X\overline{Y}\,\overline{Z} + XY\overline{Z} = m_1 + m_3 + m_4 + m_6 = \sum m(1, 3, 4, 6)$$

Conversion of Minterm and Maxterm

$$\overline{F} = m_1 + m_3 + m_4 + m_6$$

$$\Rightarrow F = \overline{m_1 + m_3 + m_4 + m_6} = \overline{m_1} \cdot \overline{m_3} \cdot \overline{m_4} \cdot \overline{m_6}$$

$$\Rightarrow F = M_1 \cdot M_3 \cdot M_4 \cdot M_6 = (X + Y + \overline{Z})(X + \overline{Y} + Z)(\overline{X} + Y + Z)(\overline{X} + \vec{Y} + Z)$$

$$= \prod M(1, 3, 4, 6)$$

Canonical Sum of Minterms

- **Any Boolean function can be expressed as a Sum of Minterms.**
  - **For the function table, the minterms used are the terms corresponding to the 1's**
  - **For expressions, expand all terms first to explicitly list all minterms. Do this by "ANDing" any term missing a variable v with a term ($\mathbf{v} + \overline{\mathbf{v}}$).**
- **Example: Implement $\mathbf{f} = \mathbf{x} + \overline{\mathbf{x}}\ \overline{\mathbf{y}}$ as a sum of minterms.**

  First expand terms: $\mathbf{f} = \mathbf{x}(\mathbf{y} + \overline{\mathbf{y}}) + \overline{\mathbf{x}}\ \overline{\mathbf{y}}$

  Then distribute terms: $\mathbf{f} = \mathbf{x}\mathbf{y} + \mathbf{x}\overline{\mathbf{y}} + \overline{\mathbf{x}}\ \overline{\mathbf{y}}$

  Express as sum of minterms: f = $m_3 + m_2 + m_0$

Another SOM Example

Expand by using truth table

$$E = \overline{Y} + \overline{XZ}$$

According to truth table Table 2-8,

$$E = \sum m(0,1,2,4,5) = \prod M(???)$$

TABLE 2-8
Boolean Functions of Three Variables

(a)	X	Y	Z	F	$\overline{F}$	(b)	X	Y	Z	E
	0	0	0	1	0		0	0	0	1
	0	0	1	0	1		0	0	1	1
	0	1	0	1	0		0	1	0	1
	0	1	1	0	1		0	1	1	0
	1	0	0	0	1		1	0	0	1
	1	0	1	1	0		1	0	1	1
	1	1	0	0	1		1	1	0	0
	1	1	1	1	0		1	1	1	0

Standard Sum-of-Products (SOP)

- **A sum of minterms form for *n* variables can be written down directly from a truth table.**
  - **Implementation of this form is a two-level network of gates such that:**
  - **The first level consists of *n*-input AND gates, and**
  - **The second level is a single OR gate (with fewer than 2^n inputs).**
- **This form often can be simplified so that the corresponding circuit is simpler.**

Standard Sum-of-Products (SOP)

Example: $F = \overline{Y} + \overline{X}Y\overline{Z} + XY$

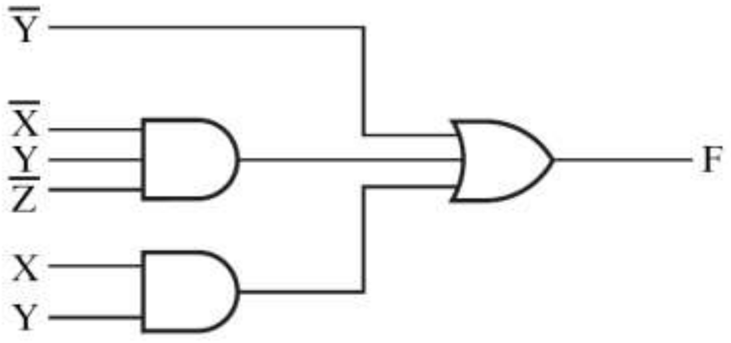


Fig. 2-5

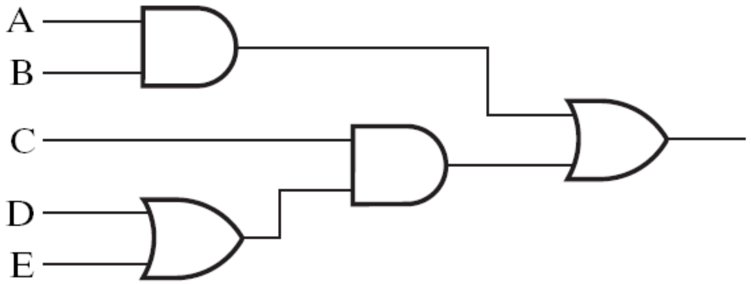
- **a two-level implementation/two-level circuit**

Product-of-Sums (POS): $F = X(\overline{Y} + Z)(X + Y + \overline{Z})$

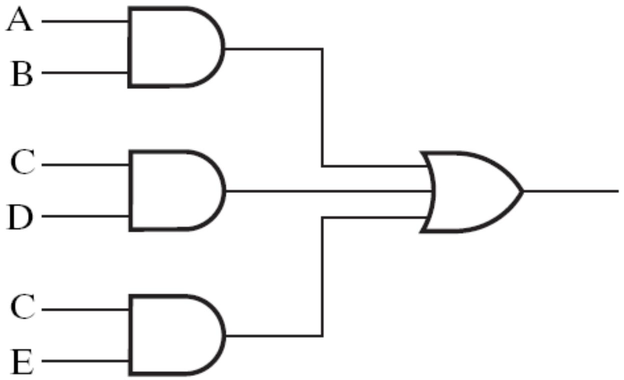
What's the implementation?

Convert non-SOP expression to SOP expression

$$F = AB + C(D+E) = AB + CD + CE$$



(a) AB + C(D + E)

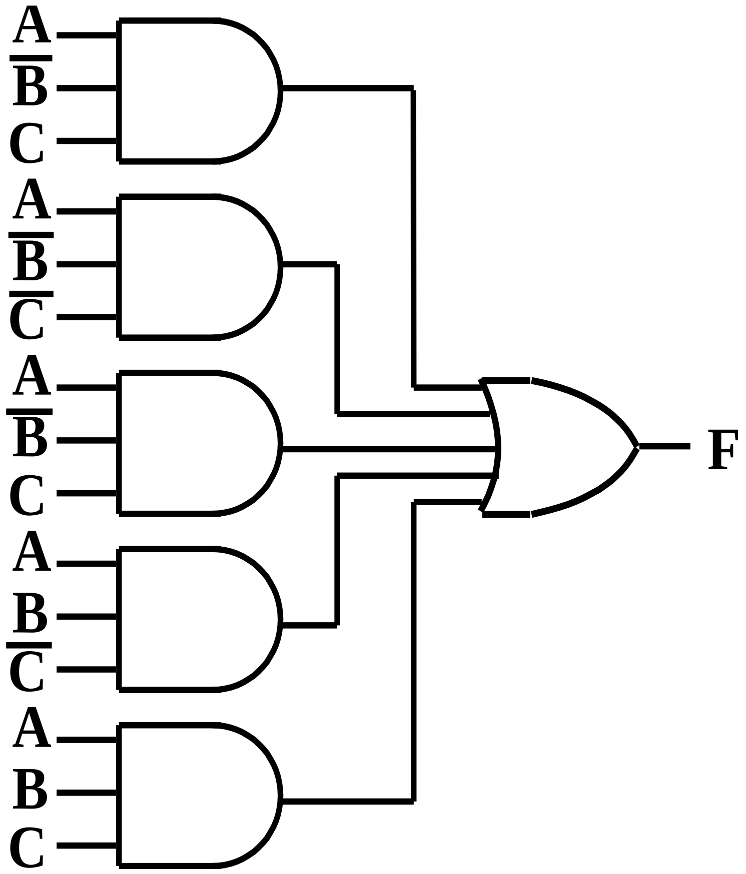


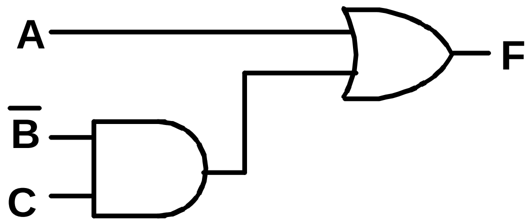
(b) AB + CD + CE

- The decision whether to use a two-level or multiple-level implementation is complex.
  - no. of gates
  - No. of gate inputs
  - amount of time delay

Simplification of two-level implementation of SOP expression

- **The two implementations for F are shown below – it is quite apparen**





SOP and POS Observations

- **The previous examples show that:**
  - **Canonical Forms (Sum-of-minterms, Product-of-Maxterms), or other standard forms (SOP, POS) differ in complexity**
  - **Boolean algebra can be used to manipulate equations into simpler forms.**
  - **Simpler equations lead to simpler two-level implementations**
- **Questions:**
  - **How can we attain a “simplest” expression?**
  - **Is there only one minimum cost circuit?**
  - **The next part will deal with these issues.**

POWER CONSUMPTION OF SYSTEM

Basic Theorem of Boolean Algebra

T1 : Properties of 0

(a) $0 + A = A$

(b) $0\,A = 0$

T2 : Properties of 1

(a) $1 + A = 1$

(b) $1\,A = A$

Basic Theorem of Boolean Algebra

T3 : Commutative Law

(a) $A + B = B + A$

(b) $A\ B = B\ A$

T4 : Associate Law

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

(b) $(A\ B)\ C = A\ (B\ C)$

T5 : Distributive Law

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

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

(c) A+A’B = A+B

Basic Theorem of Boolean Algebra

T6 : Indempotence (Identity) Law

(a) $A + A = A$

(b) $A A = A$

T7 : Absorption (Redundance) Law

(a) $A + A B = A$

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

Basic Theorem of Boolean Algebra

T8 : Complementary Law

(a) $X+X'=1$

(b) $X.X'=0$

T9 : Involution

(a) $x'' = x$

T10 : De Morgan's Theorem

(a) $(X+Y)'=X'.Y'$

(b) $(X.Y)'=X'+Y'$

De Morgan's Theorem 1

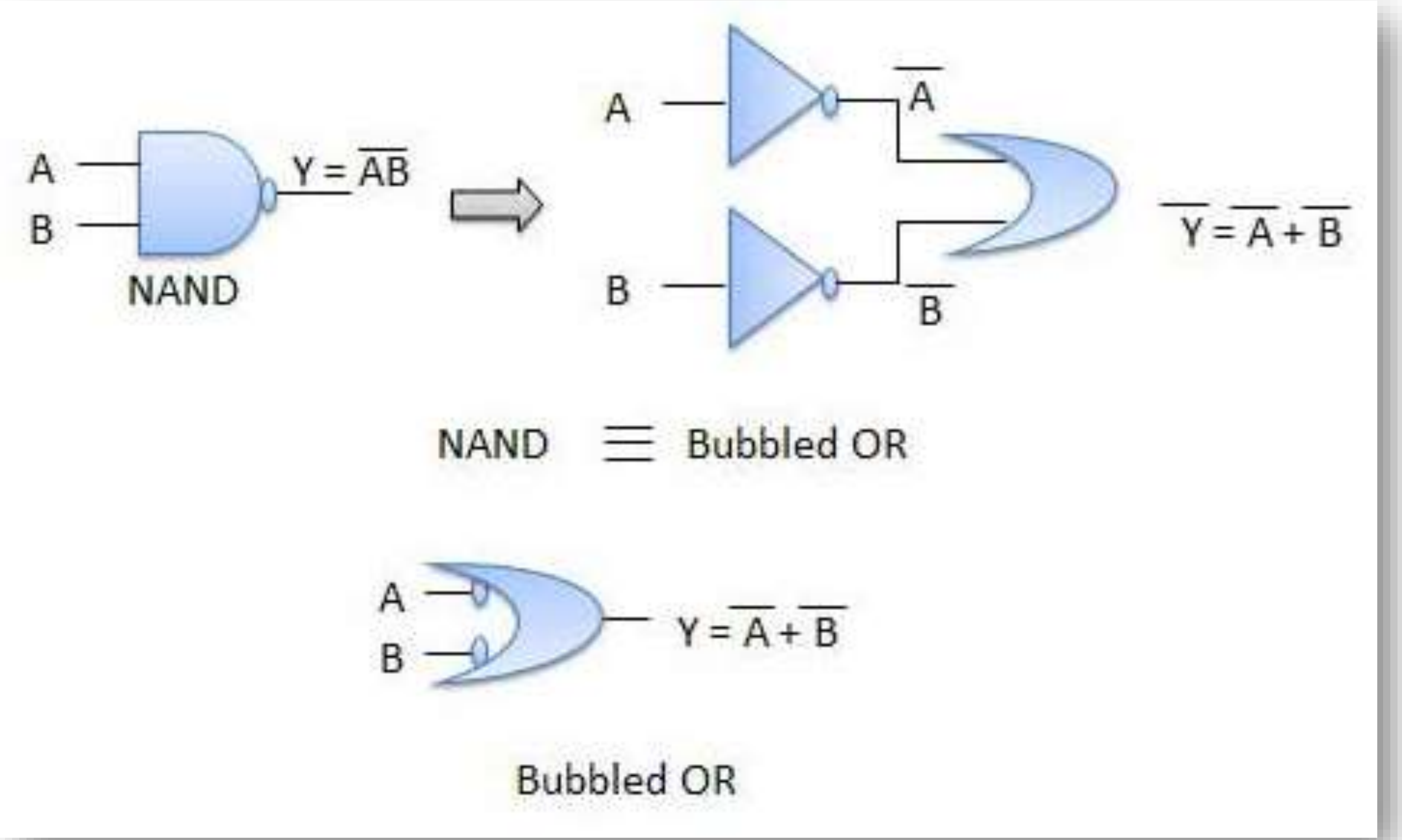
Theorem 1 $\overline{A \cdot B} = \overline{A} + \overline{B}$

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

NAND = Bubbled OR

De Morgan's Theorem 1

Theorem 1 $\overline{A \cdot B} = \overline{A} + \overline{B}$



De Morgan's Theorem 1

Theorem 1 $\overline{A \cdot B} = \overline{A} + \overline{B}$

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

De Morgan's Theorem 2

Theorem 1 $\overline{A + B} = \overline{A} \cdot \overline{B}$

De Morgan's Theorem 2

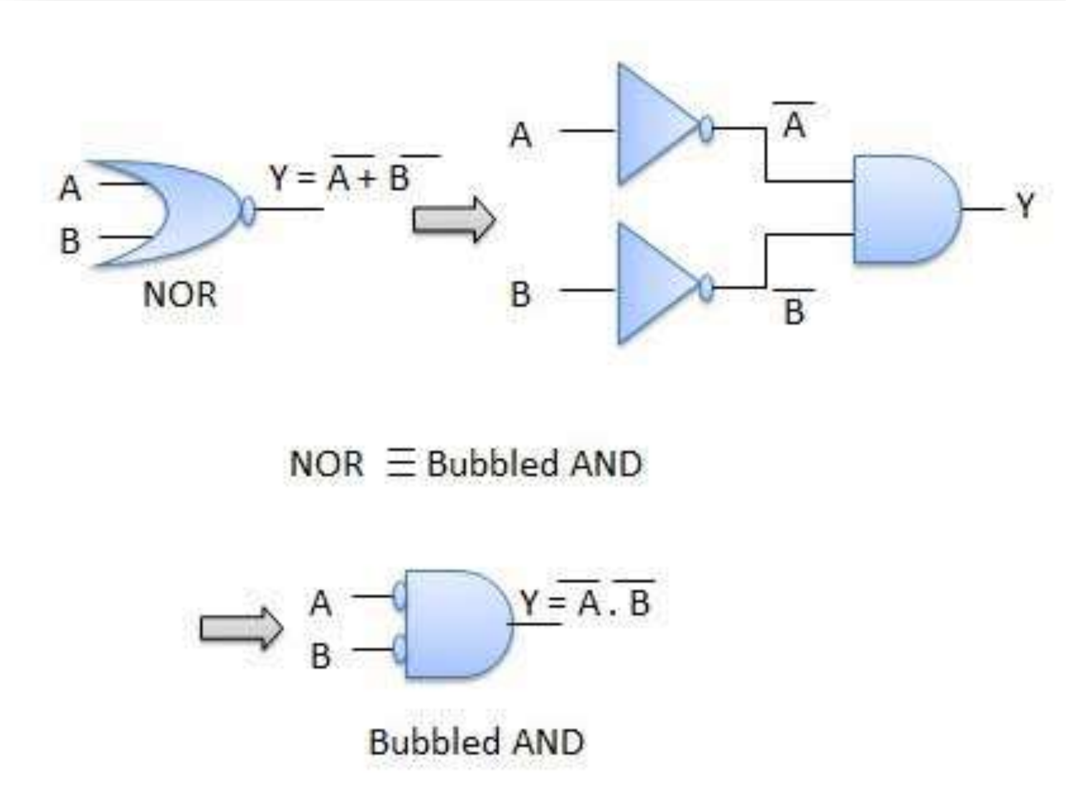
Theorem 2 $\overline{A + B} = \overline{A} \, . \, \overline{B}$

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

NOR = Bubbled AND

De Morgan's Theorem 2

Theorem 2 $\overline{A + B} = \overline{A} \cdot \overline{B}$



De Morgan's Theorem 2

Theorem 2 $\overline{A + B} = \overline{A} \cdot \overline{B}$

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