CHAPTER 3

Boolean Algebra

Contents

Boolean Algebra

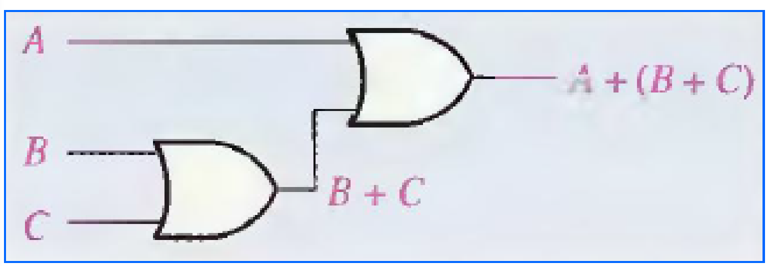
- Boolean algebra is the mathematics of digital systems.
- A basic knowledge of Boolean algebra is indispensable to the study and analysis of logic.
- In the last lecture, Boolean operations and expressions in terms of their relationship to NOT, AND, OR, NAND, and NOR gates were introduced.

Associative Law

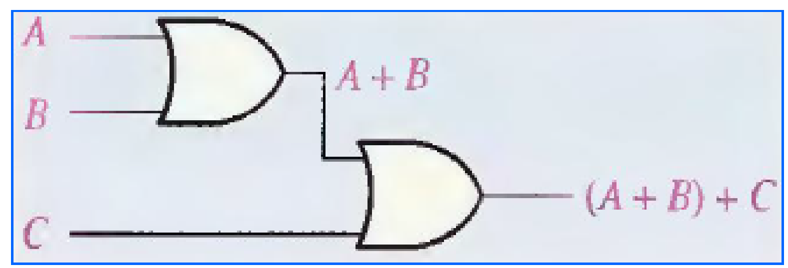
- The associative law
  - The associative law of addition is written as follows for three variables:

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

  - This law states that when ORing more than two variables, the result is the same regardless of the grouping of the variables.



=

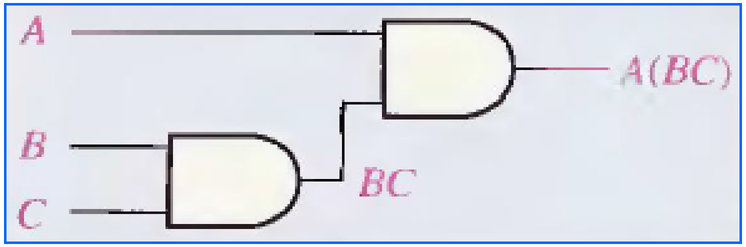


Associative Law

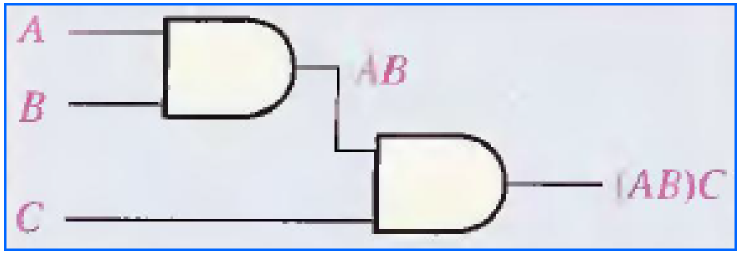
- The associative law of multiplication is written as follows for three variables:

  A(BC) = (AB)C

- This law states that it makes no difference in what order the variables are grouped when ANDing more than two variables.



=

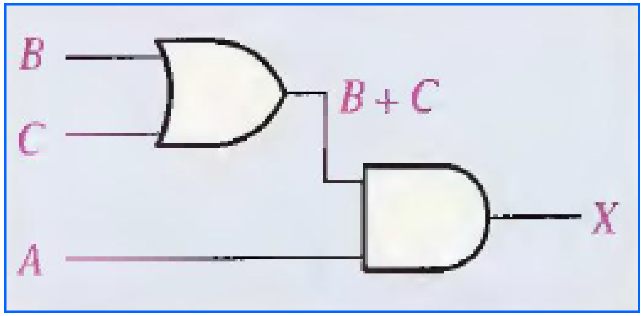


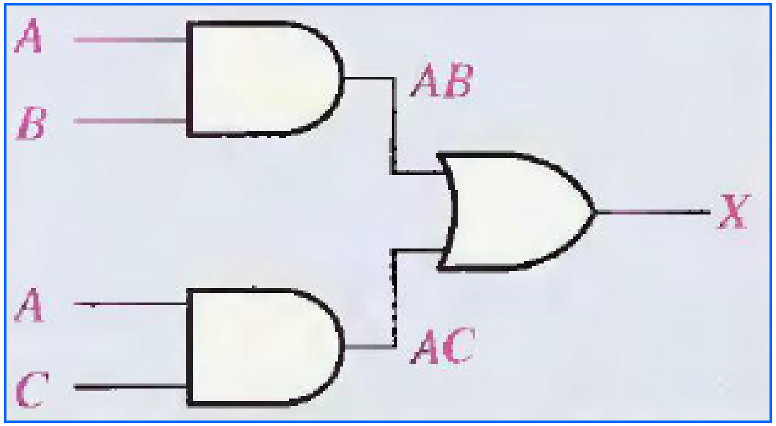
Distributive Law

The distributive law is written for three variables as follows:

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

This law states that ORing two or more variables and then ANDing the result with a single variable is equivalent to ANDing the single variable with each of the two or more variabIes and then ORing the products.





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. $A + \overline{A} = 1$
7. $A \cdot A = A$
8. $A \cdot \overline{A} = 0$
9. $\overline{\overline{A}} = A$
10. $A + AB = A$
11. $A + \overline{A}B = A + B$
12. $(A + B)(A + C) = A + BC$

A, *B*, or *C* can represent a single variable or a combination of variables.

Rules for Boolean Algebra

$A + AB = A$

- This rule can be proved by applying the distributive law, rule 2, and rule 4 as follows:

Proof

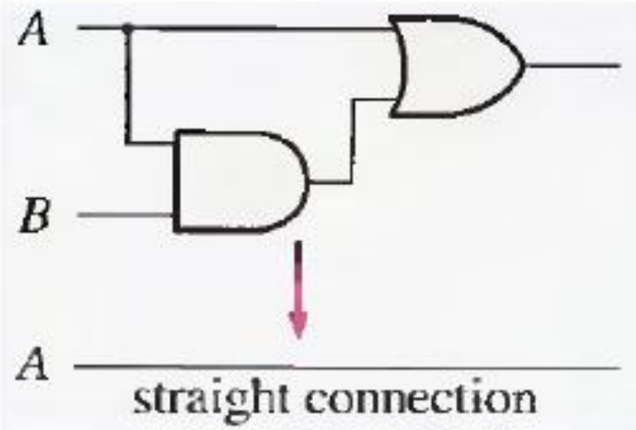
$A + AB$	$= A(1 + B)$	Factoring (distributive law)
	$= A \cdot 1$	Rule 2: $(1 + B) = 1$
	$= A$	Rule 4: $A \cdot 1 = A$

Rules of Boolean Algebra

The proof is shown in Table below, which shows the truth table and the resulting logic circuit simplification.

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

↑ A and A + AB: equal ↑



Rules for Boolean Algebra

Rule 11.

A + A'B = A + B

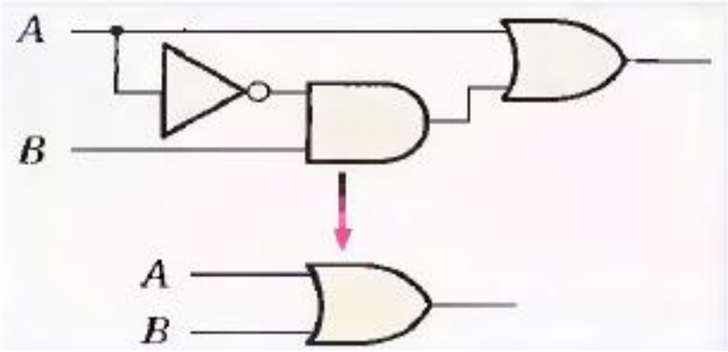
This rule can be proved as follows:

$A + \overline{A}B$	$= (A + AB) + \overline{A}B$	Rule 10: $A = A + AB$
	$= (AA + AB) + \overline{A}B$	Rule 7: $A = AA$
	$= AA + AB + A\overline{A} + \overline{A}B$	Rule 8: adding $A\overline{A} = 0$
	$= (A + \overline{A})(A + B)$	Factoring
	$= 1 \cdot (A + B)$	Rule 6: $A + \overline{A} = 1$
	$= A + B$	Rule 4: drop the 1

- **The proof is shown in Table 4--3, which shows the truth table and the resulting logic circuit simplification.**

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

$A + \bar{A}B$ and $A + B$: equal



Rules for Boolean Algebra

Rule 12.

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

Proof

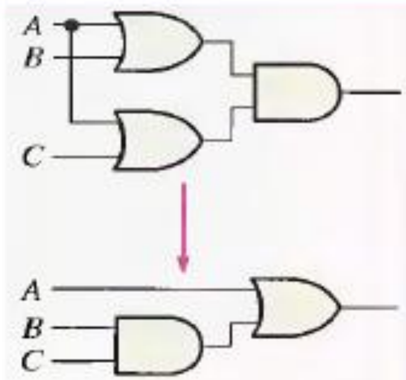
$(A + B)(A + C)$	$= AA + AC + AB + BC$	Distributive law
	$= A + AC + AB + BC$	Rule 7: $AA = A$
	$= A(1 + C) + AB + BC$	Factoring (distributive law)
	$= A \cdot 1 + AB + BC$	Rule 2: $1 + C = 1$
	$= A(1 + B) + BC$	Factoring (distributive law)
	$= A \cdot 1 + BC$	Rule 2: $1 + B = 1$
	$= A + BC$	Rule 4: $A \cdot 1 = A$

Truth Table Proof of Rule 12

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

Circuit Diagram



Examples

Simplify the boolean functions to minimum number of literals,

- $X+X'Y = (X+X')(X+Y) = X+Y$
- $X(X'+Y) = XX' + XY = 0 + XY = XY$
- $X'Y'Z + X'YZ + XY'$

$=X'Z(Y' + Y) + XY'$

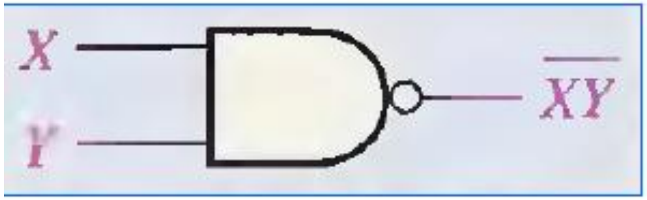
$=X'Z + XY'$

DeMorgan's Theorem

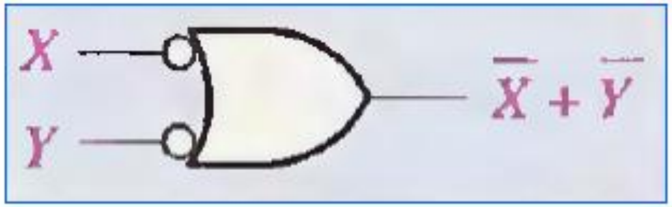
- $(x+y)' = x' \bullet y'$
- $(x \bullet y)' = x' + y'$
- Example

$$
\begin{aligned}
X &= [(A'+C) \bullet (B+D')]' \\
&= (A'+C)' + (B+D')' \\
&= (AC') + (B'D) \\
&= AC' + B'D
\end{aligned}
$$

Gate equivalencies and truth table of both the demorgan's Theorem is as follow,



NAND



Negative-OR

Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X} + \overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Truth Table Proof

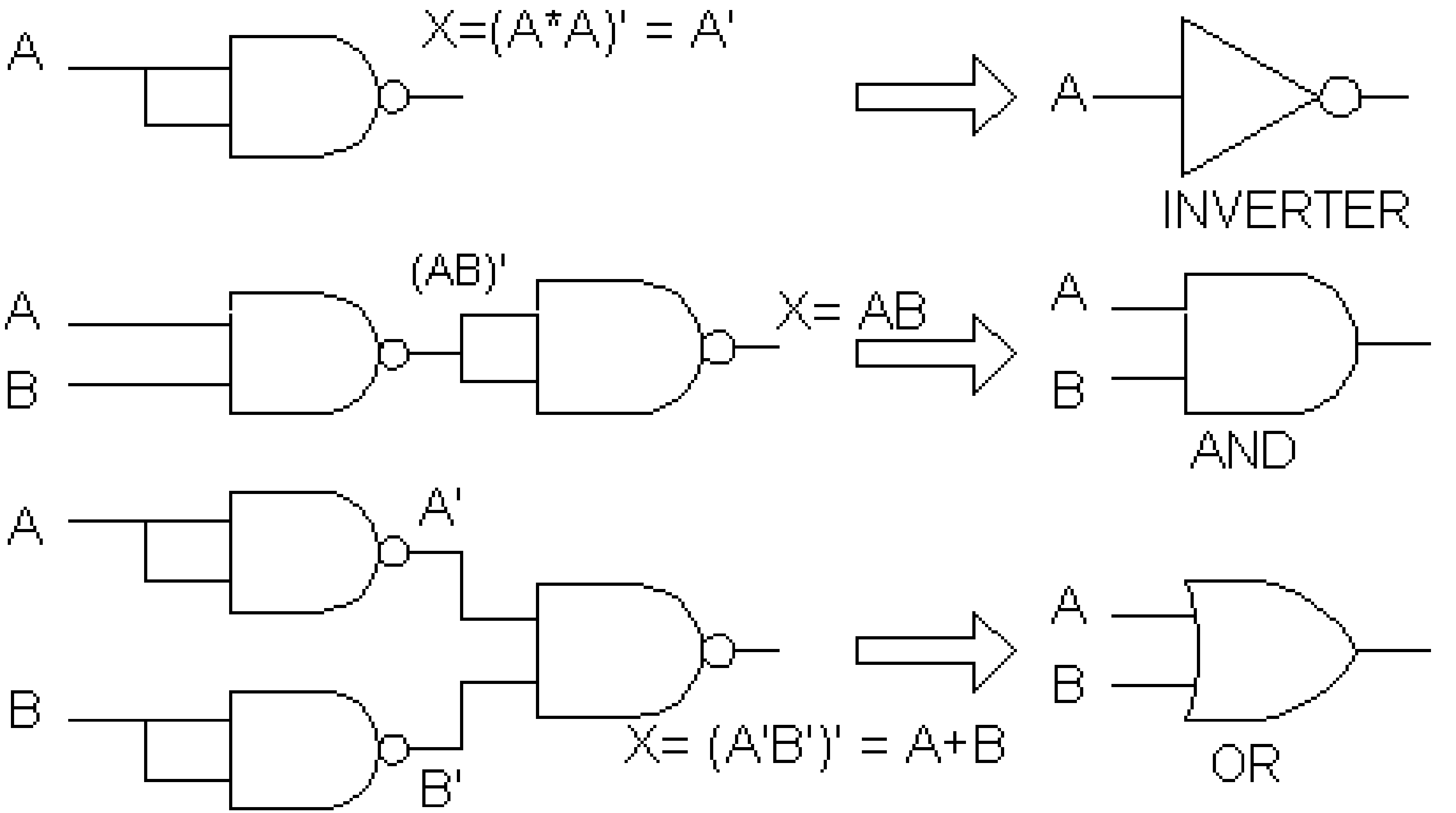
Three Variables DeMorgan's Theorem

- (x+y+z)' = x' • y' • z'
- (xyz)' = x' + y' + z`
-

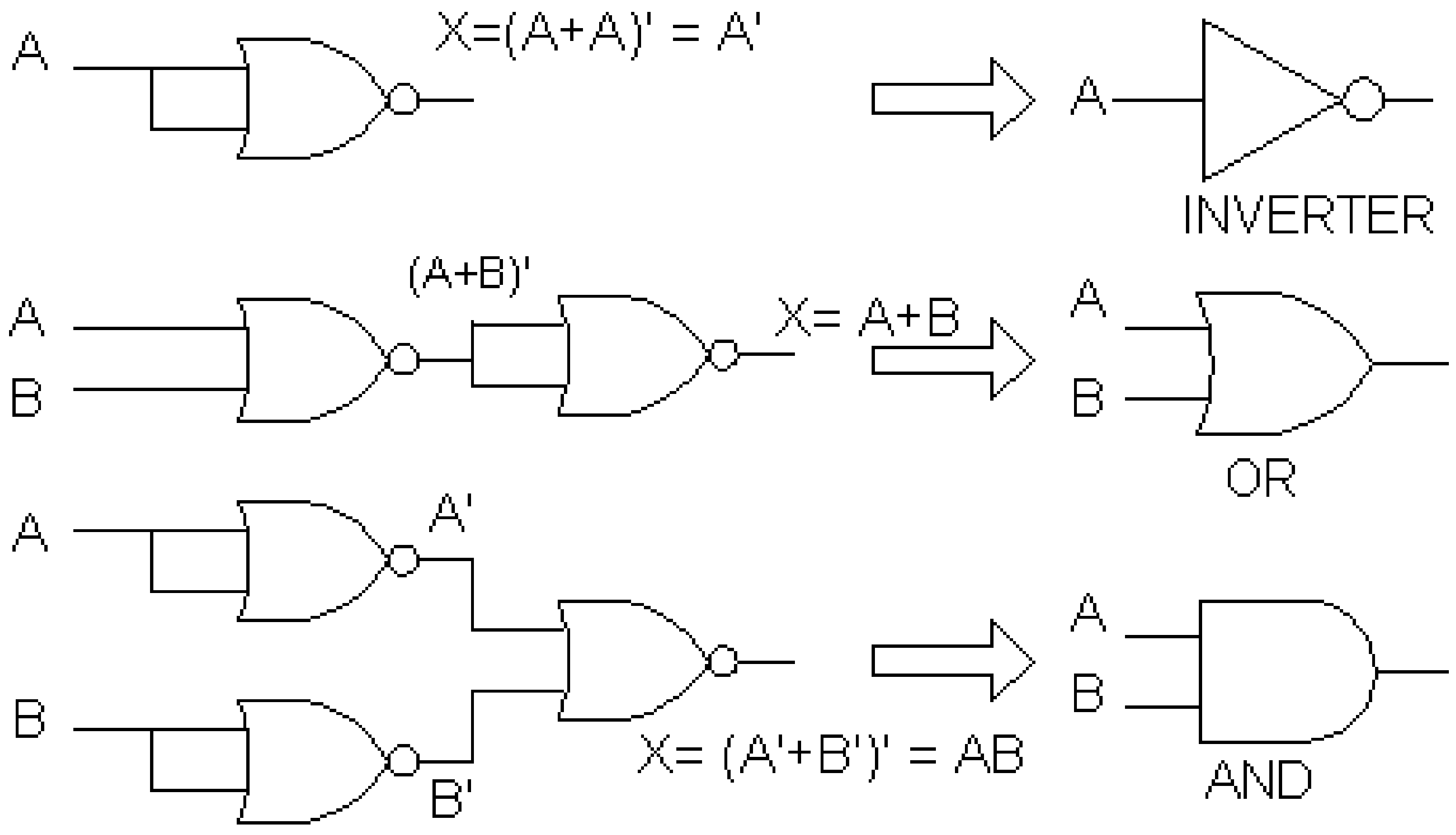
EXAMPLE:

- Apply DeMorgan's theorems to each of the following expressions:
- (a) $\overline{(A+B+C)D}$
- (b) $\overline{ABC+DEF}$
- (c) $\overline{A\overline{B}+\overline{C}D+EF}$

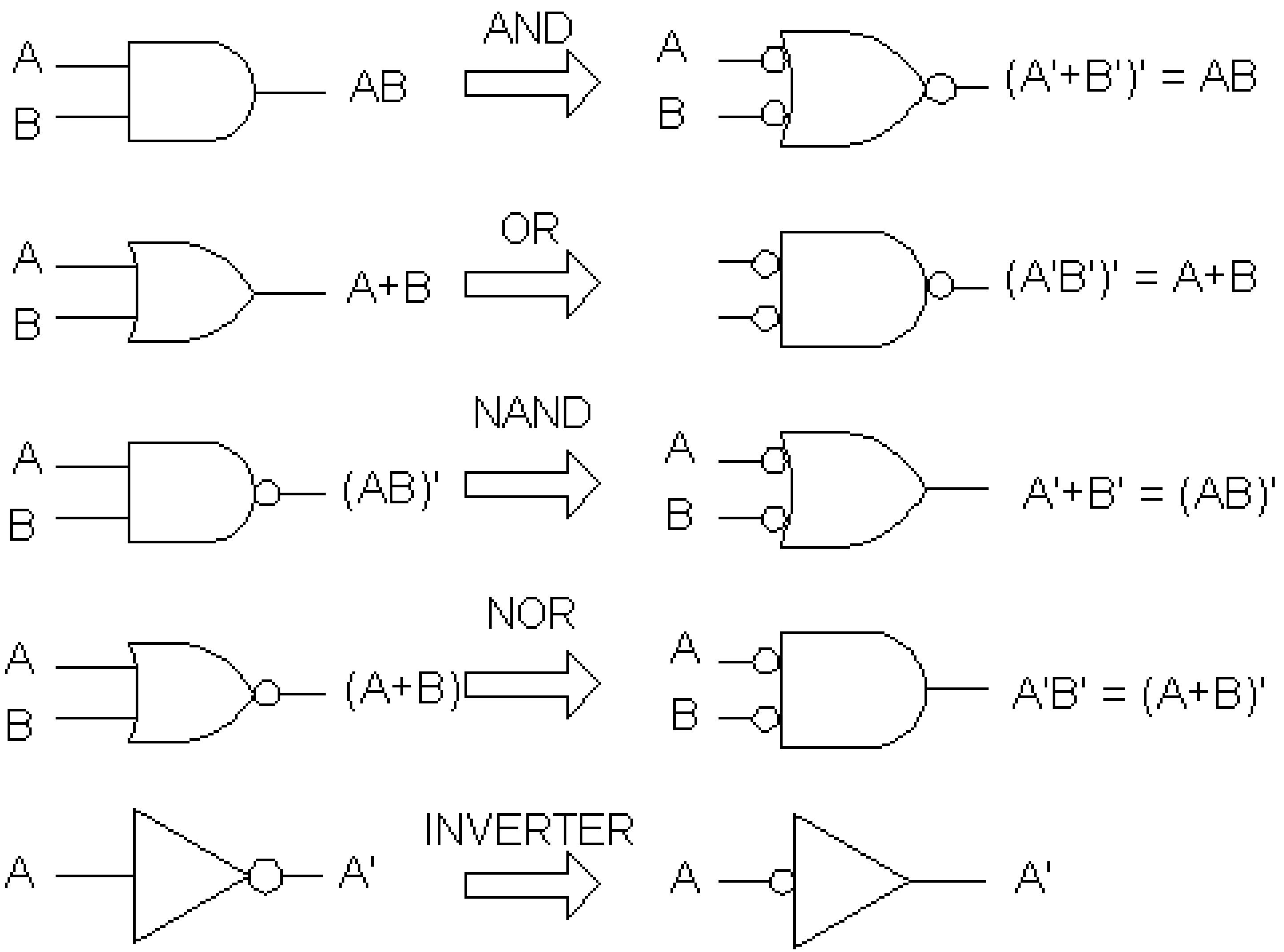
Universality of NAND Gates



Universality of NOR Gates



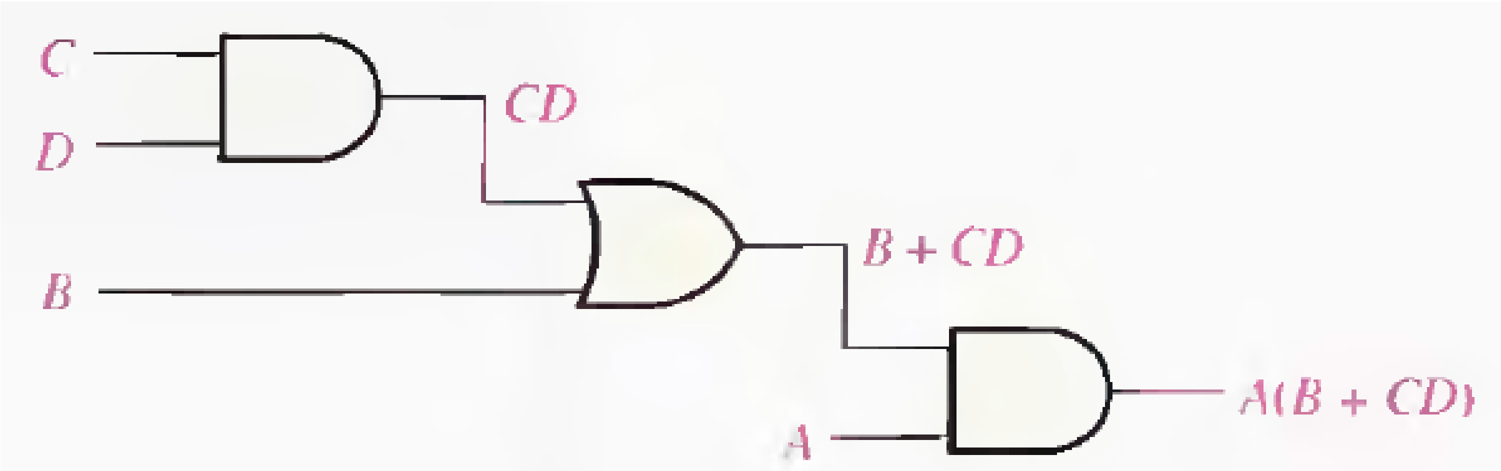
Alternate Logic Gate Representations



Boolean Expression for logic circuits

To derive the Boolean expression for a given logic circuit, begin at the left-most inputs and work toward the final output, writing the expression for each gate.

For the example circuit in Figure below, the Boolean expression is determined as follows:



Simplification using Boolean Algera

- Many times in the application of Boolean algebra, you have to reduce a particular expression to its simplest form or change its form to a more convenient one to implement the expression most efficiently.
- The approach taken in this section is to use the basic laws, rules, and theorems of Boolean algebra to manipulate and simplify an expression.
- This method depends on a thorough knowledge of Boolean algebra and considerable practice in its application,

Example

Using Boolean algebra techniques, simplify this expression:

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

Solution: The following is not necessarily the only approach.

Step 1: Apply the distributive law to the second and third terms in the expression, as follows:

AB + AB + AC + BB + BC

Step 2: Apply rule 7 (BB = B) to the fourth term.

AB + AB + AC + B + BC

Step 3: Apply rule 5 (AB + AB = AB) to the first two terms.

AB + AC + B + BC

Example cont.

Step 4: Apply rule 10 (B + BC = B) to the last two terms.

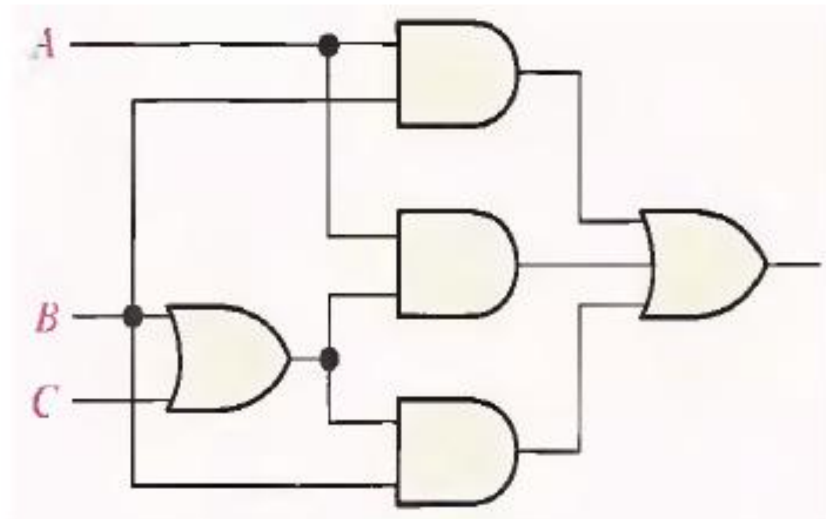
AB + AC + B

Step 5: Apply rule 10 (AB + B = B) to the first and third terms.

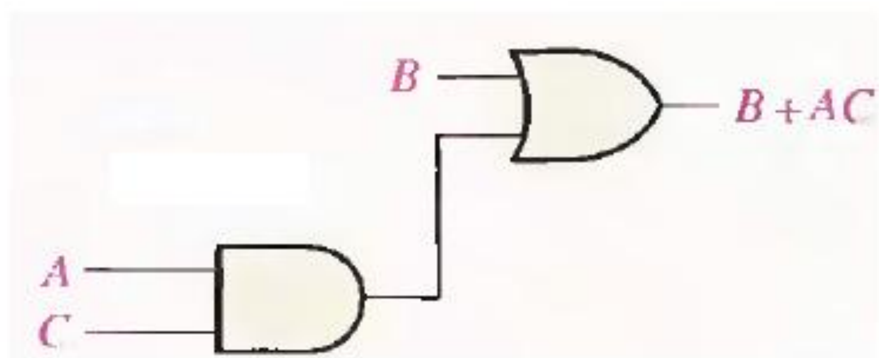
B+AC

- At this point the expression is simplified as much as possible.

Example cont.



=



Example

- Simplify the following Boolean expression to minimum number of literals,

$$[AB'(C + BD) + A'B']C$$

Solution

Step 1: Apply the distributive law to the terms within the brackets.

$$(AB'C + AB'BD + A'B')C$$

Step 2: Apply rule 8 (BB = 0) to the second term within the parentheses.

$$(AB'C + A.O.D + A'B')C$$

Step 3: Apply rule 3 (A.0.D = 0) to the second term within the parentheses.

$$(AB'C + O + A'B')C$$

Example cont

Step 4: Apply rule 1 (drop the 0) within the parentheses.

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

Step 5: Apply the distributive law.

$$AB'CC + A'B'C$$

Step 6: Apply rule 7 (CC = C) to the first term.

$$AB'C + A'B'C$$

Step 7: Factor out B′C.

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

Step 8: Apply rule 6 (A + A′ = 1).

$$B'C \cdot 1 = B'C$$

Example

Solve the expression to minimum number of literals

$$A'BC + AB'C' + A'B'C' + AB'C + ABC$$

Solution:

$A'BC + AB'C' + A'B'C' + AB'C + ABC$

$BC(A' + A) + AB'C' + A'B'C' + AB'C$

$BC.\ 1 + AB'(C' + C) + A'B'C'$

$BC + AB'.\ 1 + A'B'C'$

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

$BC + B'(A + A'C')$ **($A+A'C'=A+C'$ Rule11)**

$BC + B'(A + C')$

$BC + AB' + B'C'$ (Ans)

Self Questions

Simplify the Followings if possible,

- $(AB + AC)' + A'B'C$
- $(AB)' + (AC)' + (ABC)'$
- $A + AB + AB'C$
- $(A' + B)C + ABC$
- $AB'C(BD + CDE) + AC'$

Standard forms of Boolean expressions

- All Boolean expressions, regardless of their form, can be converted into either of two standard forms
  - **Sum-of-products (SOP) form**
    - **Also called Minterms**
  - **Product of Sum (POS) form**
    - **Also called Maxterms**
- Standardization makes the evaluation, simplification, and implementation of Boolean expressions much more systematic and easier.

Sum of Product (SOP) Form

- A product term was defined as a term consisting of the product (Boolean multiplication) of literals (variables or their complements).
- When two or more product terms are summed by Boolean addition. the resulting expression is a sum-of-products (SOP).
  - Some examples are
    - AB + ABC
    - ABC + CDE + B`CD`
    - A`B + A`BC` + AC

SOP (Minterms)

- Also, an SOP expression can contain a single-variable like,

$$A + A'BC' + BC'D'$$

- **Domain of a Boolean Expression**
  - The domain of a general Boolean expression is the set of variables contained in the expression in either complemented or uncomplemented form.
  - For example, the domain of the expression $A'B + AB'C$ is the set of variables A, B, C.

SOP Form

- Truth Table for SOP form

x	y	z	Terms	
0	0	0	$x'y'z'$	m_0
0	0	1	$x'y'z$	m_1
0	1	0	$x'y\,z'$	m_2
0	1	1	$x'y\,z$	m_3
1	0	0	$x\,y'z'$	m_4
1	0	1	$x\,y'z$	m_5
1	1	0	$x\,y\,z'$	m_6
1	1	1	$x\,y\,z$	m_7

Examples

- Convert each of the following Boolean expressions to SOP form,
  - $AB + B(CD + EF)$
    $= AB + BCD + BEF$
  - $(A + B)(B + C + D)$
    $= AB + AC + AD + BB + BC + BD$
  - $[(A + B)' + C]'$
    $= ((A + B)')'C' = (A + B)C'$
    $= AC' + BC'$

Problem: Convert $A'BC' + (A + B')(B + C' + AB')$ to SOP form.

Minterms and Maxterms

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

Canonical FORMS

- There are two types of canonical forms:
  - the sum of minterms
  - The product of maxterms

Sum of minterms

- $f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$
- $f_2 = x'yz + xy'z + xyz^1 + xyz = m_3 + m_5 + m_6 + m_7$

x	y	Z	f_1	f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Product of maxterms

- The complement of f_1 is read by forming a minterm for each combination that produces a 0 as:
- $f_1' = x'y'z' + x'yz' + x'yz + xy'z + xyz'$
- $f_1 = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z)(x' + y' + z)$

  $= Mo\ M_2\ M3\ M5\ M6$
- Similarly
- f2 = ?

Example: Sum of Minterms

- Express the Boolean function $F = A + B'C$ in a sum of minterms.
- F=A+B'C

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

Example: Product of Maxterms

- Express the Boolean function $F = xy' + yz$ in a product of maxterm form.
- $F = xy' + yz = (xy' + y)(xy' + z) = (x + y)(y' + y)(x + z)(y' + z)$

  $= (x + y)(x + z)(y' + z) = (x + y + zz')(x + yy' + z)(xx' + y' + z)$

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

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

  $= M_0\ M_1\ M_2\ M_6$

  $= \Pi\ (0,1,2,6)$

STANDARD FORMS

- There are two types of standard forms:
  - the sum of products (SOP)
  - The product of sums (POS).

Sum of Products

- The *sum of products* is a Boolean expression containing AND terms, called *product terms,* of one or more literals each. The *sum* denotes the ORing of these terms.
- $F = xy + z + xy'z'$. (SOP)

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.
- $F = z(x+y)(x+y+z)$ (POS)
- F = x (xy' + zy) (nonstandard form)

QUESTIONS?