KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING

COE 272
Digital Systems

Lecture 2: Boolean Algebra and Binary Logic (Part 1)

Boolean (Binary) Algebra

- Boolean Algebra is used to simplify and analyze digital (i.e. logic) circuits
- Because of the binary numbers, it is also called *binary or logical algebra*
- Boolean Algebra was invented by George Boole in 1854

Boolean Vs. Conventional Algebra

- Boolean Algebra is different in the following ways:
  - Symbols used in Boolean Algebra do not represent numerical values
  - Arithmetic operations (addition, subtraction, division, etc.) are not performed in Boolean algebra
  - There are no fractions, negative numbers, square, square roots, etc.
  - The most important point is that Boolean algebra allows only two values (0 and 1) for any variable

Logical Operations

- Various logical operations and their Boolean expressions are expressed in the table below:

#	Name of the gate	Logical operation	Boolean expression
1	NOT gate	Inversion	$Y = \overline{A}$
2	AND gate	Logical Multiplication	Y=AB
3	OR gate	Logical addition	Y = A + B
4	NAND gate	NOT AND	$Y = \overline{AB}$
5	NOR gate	NOT OR	$Y = \overline{A + B}$
6	Exclusive OR	Addition/Subtraction	$Y = A \oplus B$
7	Exclusive NOR	NOT XOR	$Y = \overline{A \oplus B}$

Boolean Postulates

1(a). Closure with respect to operator (+)

- When operator (+) is used over two binary elements, it results in a unique binary element.

1(b). Closure with respect to operator (.)

- When operator (.) is used over two binary elements, it results in a unique binary element.

2(a). The identity element with respect to (+) is designated by 0

$$A+0 = 0+A = A$$

2(b). The identity element with respect to (.) is designated by 1

$$A \cdot 1 = 1 \cdot A = A$$

Boolean Postulates

3(a). Commutative with respect to (+)

$$A+B = B+A$$

3(b). Commutative with respect to (.)

$$A \cdot B = B \cdot A$$

4(a). (.) is distributive over (+)

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

4(b). (+) is distributive over (.)

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

Boolean Postulates

5. For every binary element A, there exists an element called complement of A such that

$$A + \bar{A} = 1 \text{ and } A \cdot \bar{A} = 0$$

6. In a set of binary elements, there always exists at least two elements A and B, such that A ≠ B, A = 0 then B = 1 and vice versa

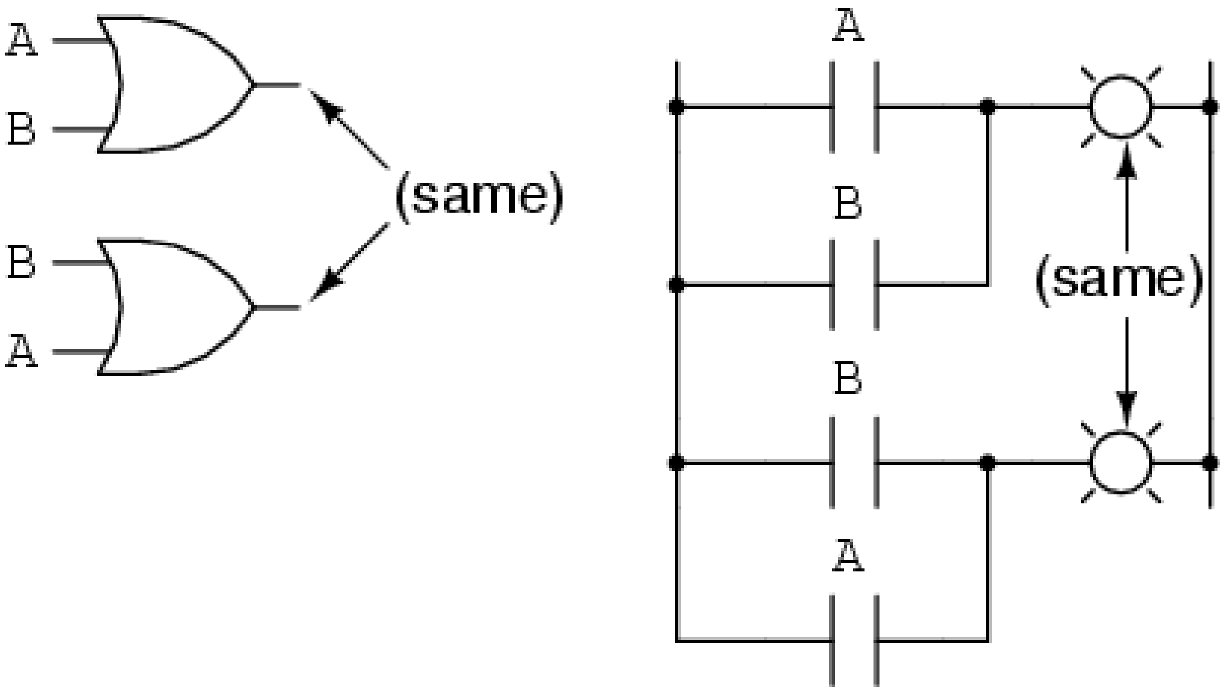
Boolean Law

- Commutative Law
- AND Law
- Associative Law
- OR Law
- Distributive Law
- Inversion Law

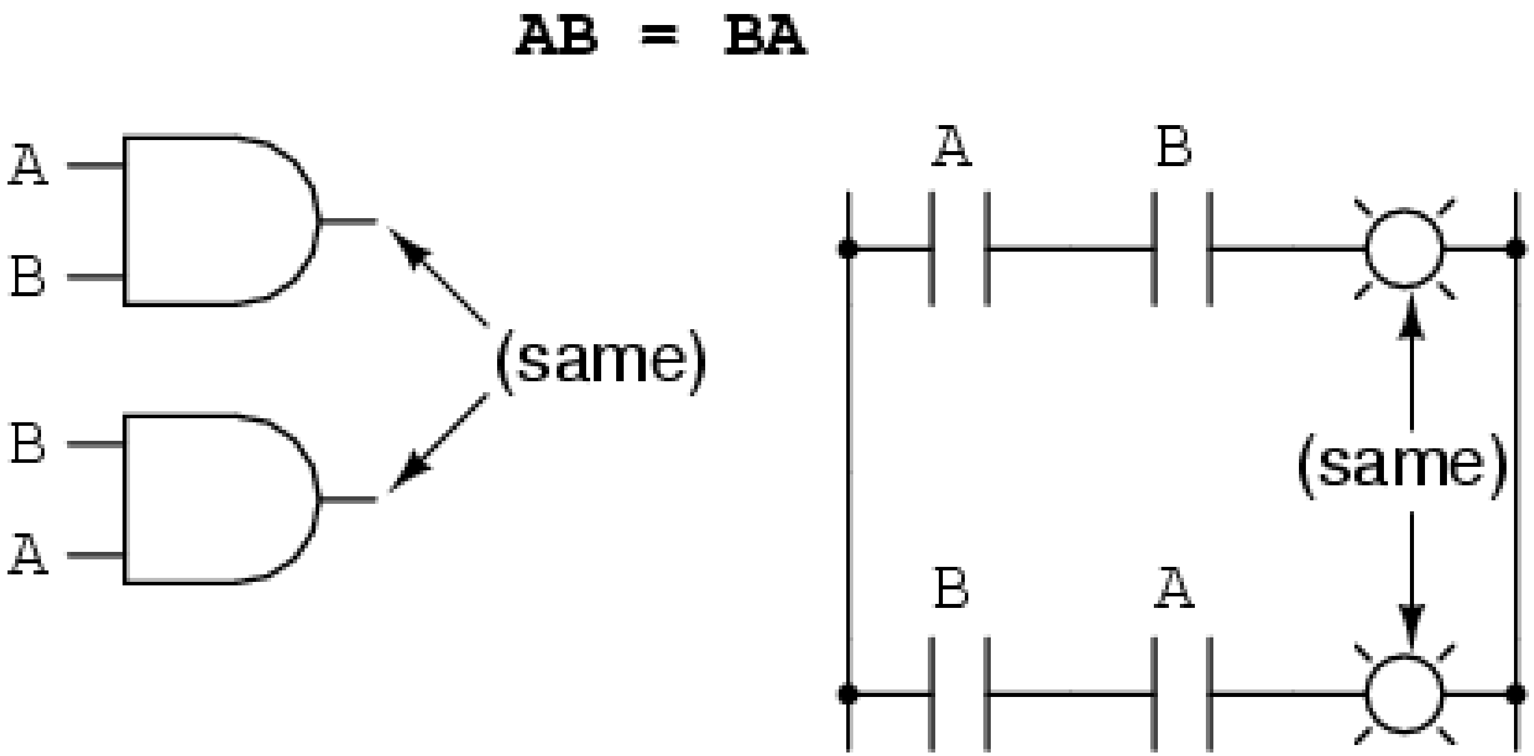
Commutative Law

Commutative property of addition

$$A + B = B + A$$

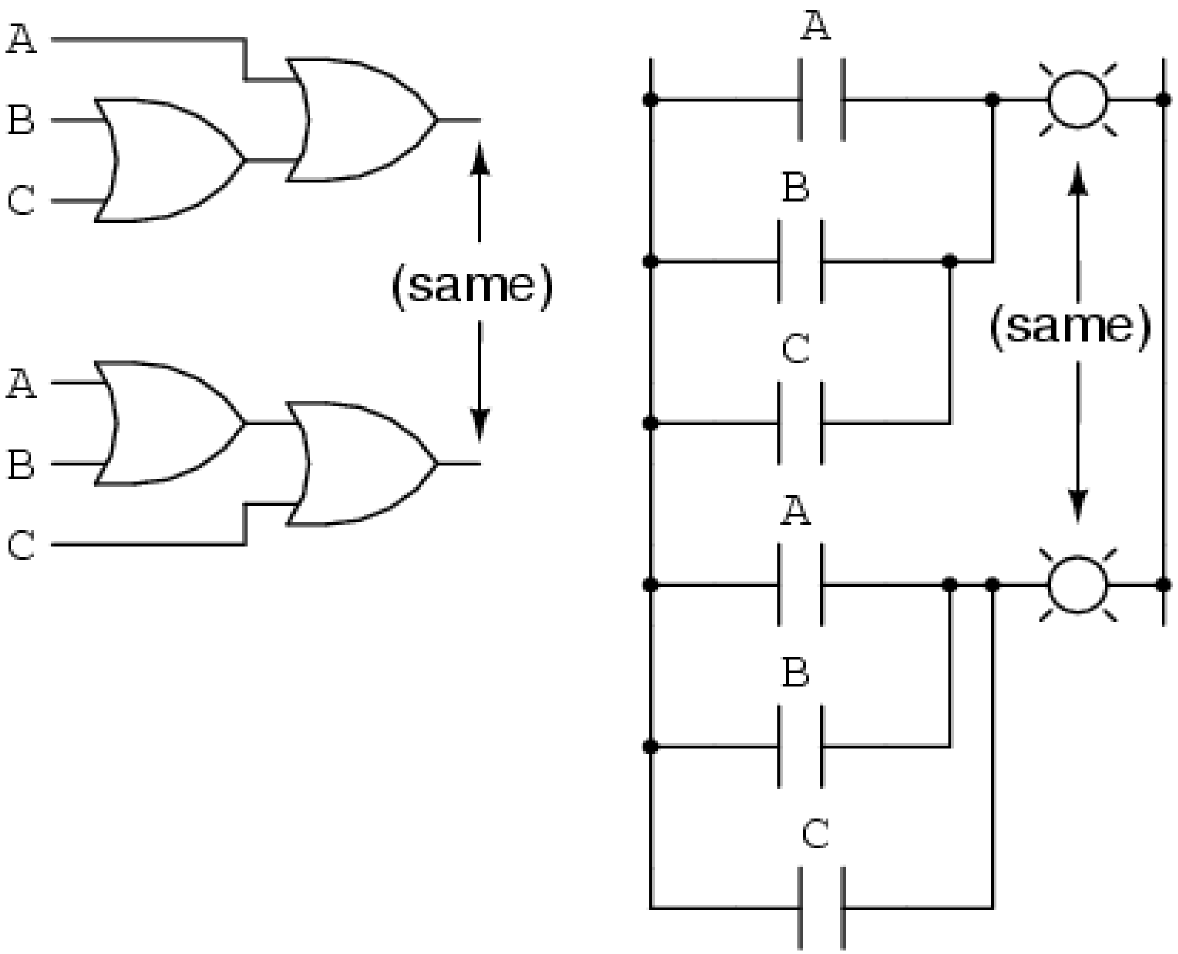


Commutative Law



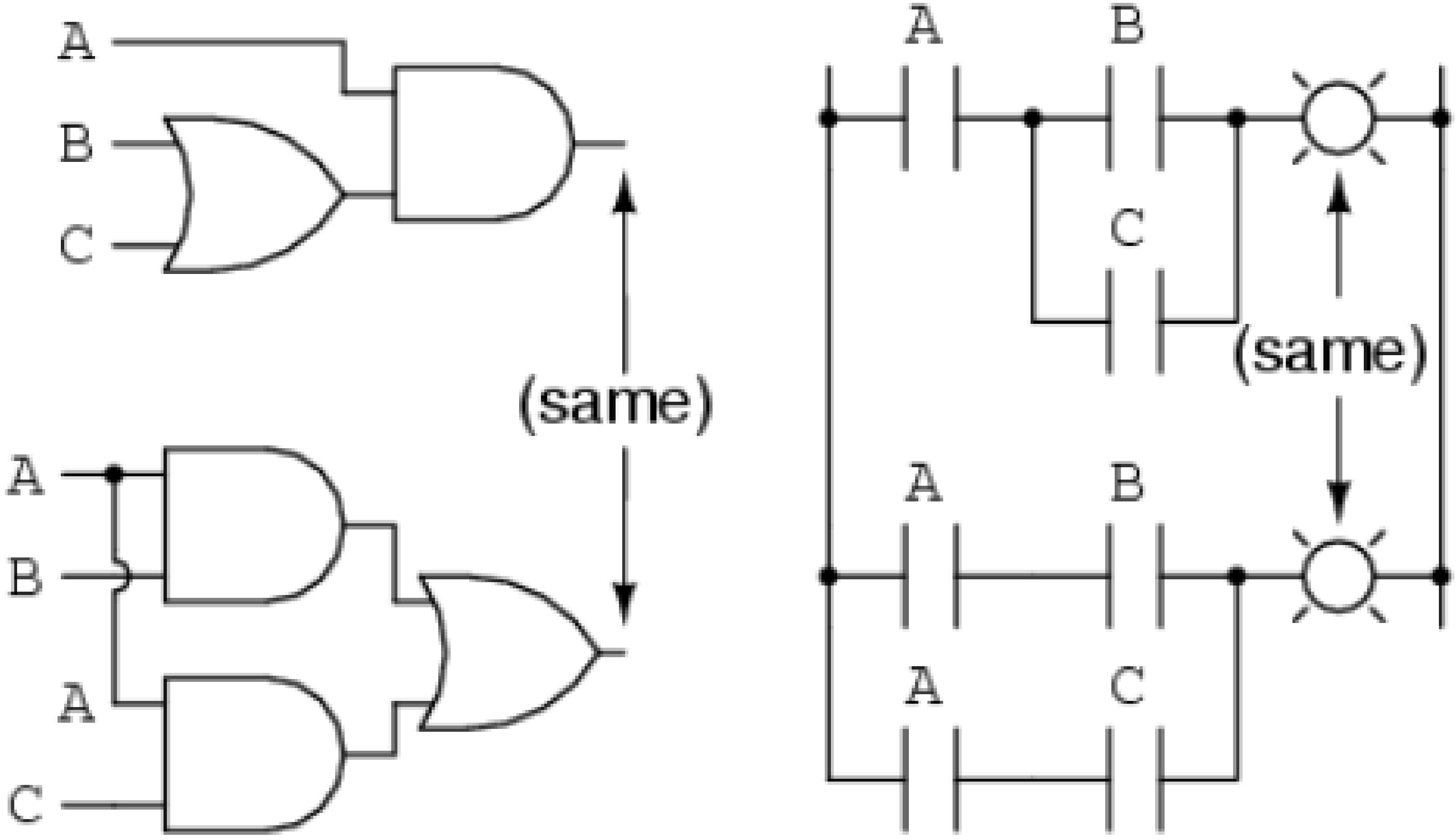
Associative Law

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



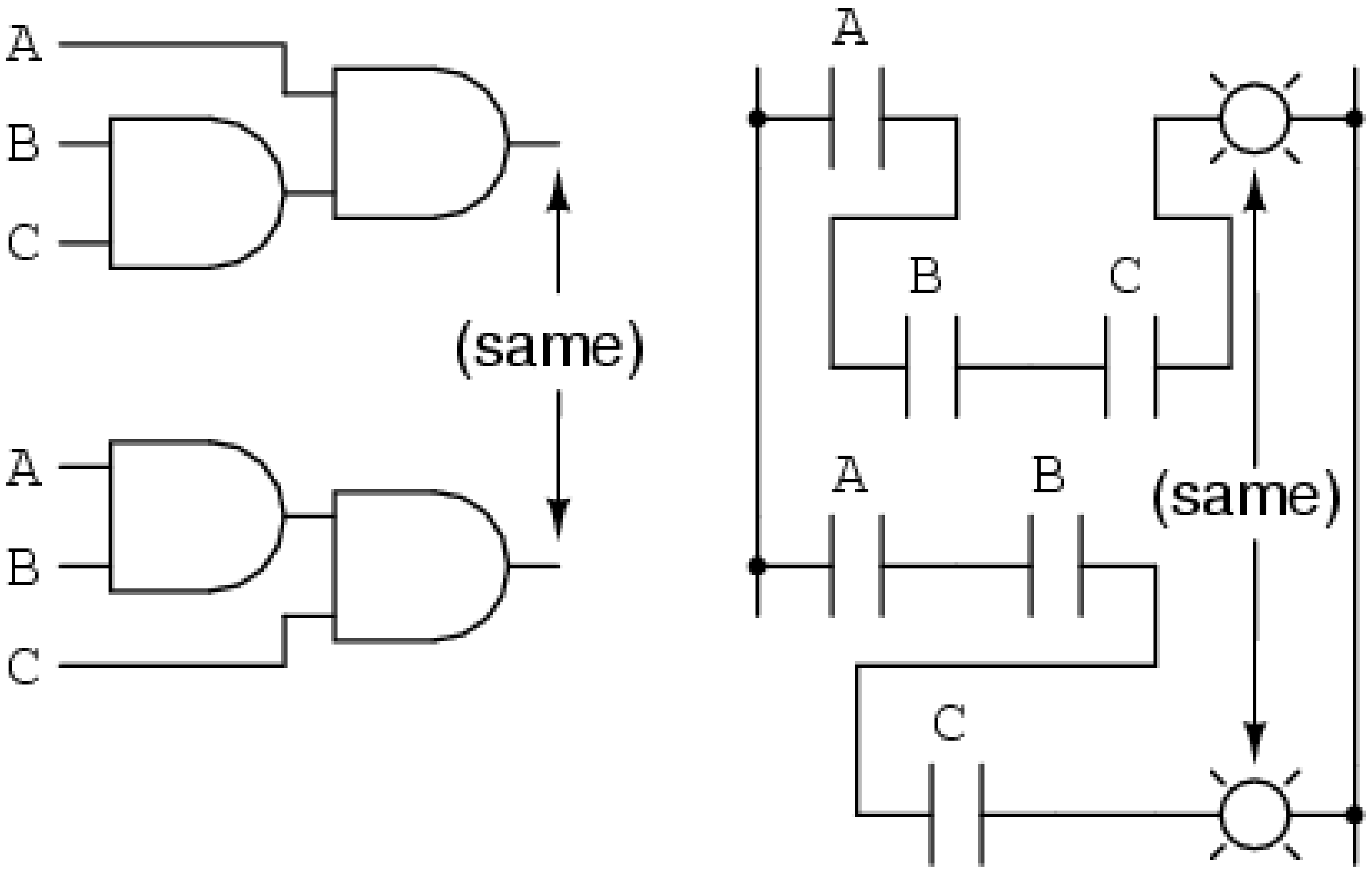
Distributive Law

A(B + C) = AB + AC



Associative Law

A(BC) = (AB)C



AND Law

- $A \cdot 0 = 0$
- $A \cdot 1 = A$
- $A \cdot A = A$
- $A \cdot \overline{A} = 0$

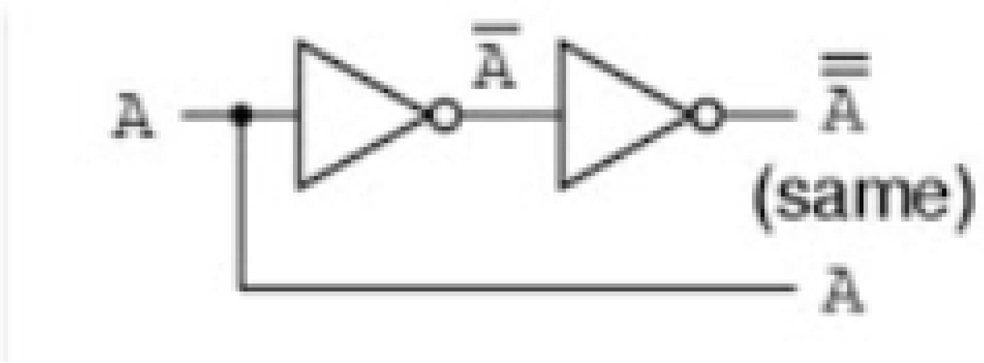
OR Law

- $A + 0 = A$
- $A + 1 = 1$
- $A + A = A$
- $A + \overline{A} = 1$

Inversion Law

- This uses the NOT operation
- It states that the double inversion of a variable results in the original variable itself.

$$\overline{\overline{A}} = A$$



Other Important Rules

- For the simplification of the Boolean expressions, the following additional rules
  - $A + AB = A$
  - $A + \overline{A}B = A + B$
  - $(A + B)(A + C) = A + BC$

Proof of important rules

- $A + A \cdot B = A$

$$A + A \cdot B = A \cdot (1 + B)$$

But $1 + B = 1$

$$= A \cdot 1$$

$$= A$$

Proof of important rules

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

Substituting $A = A + AB$, we have

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

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

But we know that $A + \bar{A} = 1$

$= A + B . 1$

$= A + B$

Proof of important rules

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

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

But $A . A = A$ and $B . A = A . B$

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

But $A + AB = A$

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

$(A + B) . (A + C) = A . (1 + C) + BC$

But $1 + C = 1$

$= A . 1 + B . C$

$= A + B . C$

Duality Principle

- According to the duality principle, the following conversions are possible in a given Boolean expression
  - *We can change each AND operation to an OR operation*
  - *We can each OR operation to an AND operation*
  - *We can compliment any 1 or 0 appearing in the expression*

Duality Principle

- The duality principle is sometimes useful in creating new expressions from the given Boolean expression

- Any theorem or identity in switching algebra remains true if 0 and 1 are swapped and (.) and (+) are also swapped.

- For example,
  - A + 1 = 1; A . 0 = 0
  - A . (B+ C) = A . B + A . C; A+(B . C) = A + B . A + C

Duality Practice Questions

- $A + A \cdot B = A$

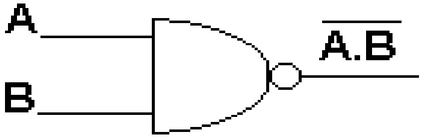
- $A + \bar{A} \cdot B = A + B$

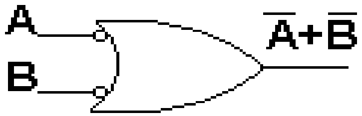
- $A + \bar{A} = 1$

- $A \cdot \bar{B} + \bar{A} \cdot B + A \cdot B + \bar{A} \cdot \bar{B} = 1$

DeMorgan's Theorem

- Theorem 1:
  - $\overline{AB} = \overline{A} + \overline{B}$; NAND = Bubbled OR





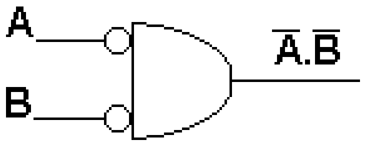
A	B	Z
$\overline{A}$	$\overline{B}$	Z
$\overline{A}$	B	Z
A	$\overline{B}$	Z
A	B	$\overline{Z}$

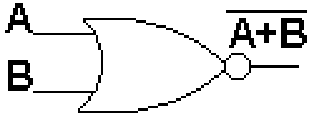
$$\overline{A.B} = \overline{A}+\overline{B}$$

A	B	Z
$\overline{A}$	$\overline{B}$	Z
$\overline{A}$	B	Z
A	$\overline{B}$	Z
A	B	$\overline{Z}$

DeMorgan's Theorem

- Theorem 2:
  - $\overline{A + B} = \overline{A} \,.\, \overline{B}$; NOR = Bubbled AND





A	B	Z
$\overline{A}$	$\overline{B}$	Z
$\overline{A}$	B	$\overline{Z}$
A	$\overline{B}$	$\overline{Z}$
A	B	$\overline{Z}$

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

A	B	Z
$\overline{A}$	$\overline{B}$	Z
$\overline{A}$	B	$\overline{Z}$
A	$\overline{B}$	$\overline{Z}$
A	B	$\overline{Z}$

Boolean Functions and Expression

- A Boolean Function is expressed by a Boolean Expression.
- Boolean Expressions consist of binary variables, the constant 0 and 1 and the logical operation symbols.

$$F(A,B,C,D) = A + B\bar{C} + ADC$$

Boolean Function

Boolean Expression

- NB: There is an order of precedence when evaluating expressions. Below is the order ;
  - Parentheses (innermost are done first)
  - Complements (NOT)
  - AND
  - OR

Practice Questions

- Simplify the following expressions
  - $Y = \left(\overline{\overline{AB} + \overline{A} + AB}\right)$
  - $A\overline{B} + \overline{A}B + \overline{A}\,\overline{B} + AB$
  - $A\overline{B}C + \overline{A}BC + ABC$
  - $Y = (AB + C)(AB + D)$

Ways of Simplifying Boolean Function

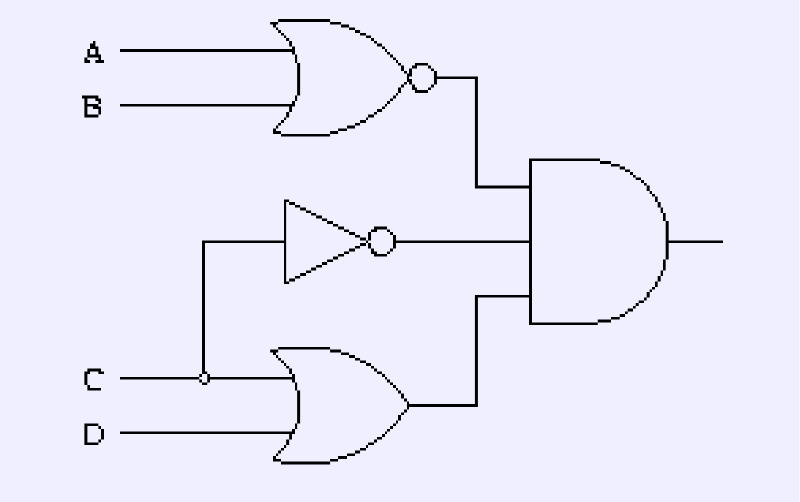
- Algebraic Method

- Karnaugh-map (K-Map)

- Quine-McClusky Method

SOP and POS Representations

- Assuming that a logic expression is:
  - $Y = (\overline{A + B})(C + D)\overline{C}$
  - A,B,C,D are called literals and expressions is realized as shown below.



SOP and POS Representations

- Any logic expression is expressed in the following two standard forms:
  - Sum-of-Products form (SOP)
  - Product-of-Sums form (POS)

- These two forms are helpful for reducing given logic expressions to its simplest form

Sum-of-Products (SOP) Form

- An expression is of the sum-of-products form if it is formed by the sum of products, and all products are formed by single variables only

- Examples:
  - $A\overline{B} + C\overline{D}E + A\overline{C}\overline{E}$
  - ABC + DEFG + H
  - $A + \overline{B} + C + \overline{D}E$

- The following however, are not sum-of-products
  - (A + B)CD + EF
  - (X + Y)(X + Z)

Product-of-Sums (POS) Form

- A POS is formed by product of sums in which all the sums are single variables only

- Examples:
  - $(A + \overline{B})\ (C + \overline{D} + E)\ (A + \overline{C} + \overline{E})$
  - $(A + B)\ (C + D + E)\ F$

- The following however, are not sum-of-products
  - $(A + B)CD + EF$
  - $A + \overline{B} + C + \overline{D}E$

Standard or Canonical SOP and POS (Minterm and Maxterms)

- Standard SOP
  - *Each product term consists of all the literals in the complemented or uncomplemented form*
  - *Each of the products in the SSOP form is called a minterm. Thus the form is also called sum-of-minterms*

$$f(A,B,C,D) = AB\overline{C}D + \overline{A}BCD + ABCD$$

- Standard POS
  - *Each sum term consists of all the literals in the complemented or uncomplemented form*
  - *Each of the products in the SPOS form is called a maxterm. Thus the form is also called product-of-maxterms*

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

Conversion procedure from SOP to Standard SOP

1. For each term, find the missing literal

2. Then, we AND term with the term found by ORing the missing literal and its complement

Example of Conversion (SOP to SSOP)

- Convert the expression $Y(A,B,C) = AB + A\bar{C} + BC$ into the standard form
  - Find the missing literals
    - AB – missing literal is C
    - AC – missing literal is B
    - BC – missing literal is A
  - We AND each term with (missing literal + Its compliment)
    - $Y = AB(C+ \bar{C}) + A\bar{C}(B + \bar{B}) + BC(A + \bar{A})$
    - $Y = ABC + AB\bar{C} + AB\bar{C} + A\bar{B}\bar{C} + ABC + \bar{A}BC$
    - $Y = (ABC + ABC) + (AB\bar{C} + AB\bar{C}) + A\bar{B}\bar{C} + \bar{A}BC$

      But A+ A = A, therefore $ABC + ABC = ABC$ and $AB\bar{C} + AB\bar{C} = AB\bar{C}$
    - $Y = ABC+ AB\bar{C} + A\bar{B}\bar{C} + \bar{A}BC$

Conversion procedure from POS to standard POS

1. For each term, find the missing literal

2. Then, OR each term with (the ANDing of the missing term and its complement)

3. Simplify expression to get the standard POS

Example of Conversion (POS to SPOS)

- Convert the expression $Y(A,B,C) = (A+B)(\bar{B} + C)$ into the standard form
  - Find the missing literals
    - AB – missing literal is C
    - BC – missing literal is A
  - OR each term with the (missing literal and its complement)
    - $Y = (A + B + C.\bar{C})(\bar{B} + C + A.\bar{A})$
      But $A + BC = (A + B)(A + C)$
    - $Y = (A + B + C)(A + B + \bar{C})(\bar{B} + C + A)(\bar{B} + C + \bar{A})$

Practice Examples

- Convert the following expressions into their Standard SOP or POS forms:
  - Y = AB + AC + BC
  - Y = A + BC + ABC
  - Y = (A+ B) ($\bar{B}$ + C)

Minterm Designation

- Writing minterm for a particular combination of ABC
  - *Let ABC = 011, assuming ABC are inputs to an AND gate, we want the output of the AND gate to be 1*
  - *As a result all the inputs must be 1s. hence we complement the input which is 0.*
  - *Therefore minterm =* $\bar{A}BC$ *corresponding to ABC = 011*

Maxterm Designation

- Writing maxterm for a particular combination of ABC
  - *Let ABC = 011, assuming ABC are inputs to an OR gate, we want the output of the OR gate to be 0*
  - *As a result all the inputs must be 0s. hence we complement the input which is 0.*
  - *Therefore maxterm =* $(A + \bar{B} + \bar{C})$ *corresponding to ABC = 011*

Minterm and Maxterm Designation

Variables			Minterm	Maxterm
A	B	C	m	M
0	0	0	$\overline{ABC}$ = m0	A + B + C = M0
0	0	1	$\bar{A}\ \bar{B}$ C = m1	A+ B+ $\bar{C}$ = M1
0	1	0	$\bar{A}\mathbf{B}\bar{C}$ = m2	A + $\bar{B}$ + C = M2
0	1	1	$\bar{A}$BC = m3	A + $\bar{B}$ + $\bar{C}$ = M3
1	0	0	A$\bar{B}\bar{C}$ = m4	$\bar{A}$ + B + C = M4
1	0	1	A$\bar{B}$C = m5	$\bar{A}$ + B + $\bar{C}$ = M5
1	1	0	AB$\bar{C}$ = m6	$\bar{A}$ + $\bar{B}$ + C = M6
1	1	1	ABC = m7	$\bar{A}$ + $\bar{B}$ + $\bar{C}$ = M7

Representing logical expressions from min/maxterms

- We can represent the logical expressions using minterms, such as,

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

  - $Y = m_7 + m_3 + m_4$
  - $Y = \sum(3, 4, 7)$

- We can represent the logical expressions using maxterms, such as,

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

  - $Y = M_2 \,.\, M_0 \,.\, M_6$
  - $Y = \prod(0, 2, 6)$

Writing SSOP from truth tables

- The follow shows the procedure on how to write SSOP from truth table:
  1. First, consider only the combinations that have their output as a 1.
  2. Then, write a product term of each such combination
  3. Lastly, OR all these product terms to get the SSOP form

Example

- From the truth table obtain the logical expression in the SSOP form

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Answer: $Y = \bar{A}B + A\bar{B}$

$Y = \sum(1, 2)$

Practice Question

- For the truth table given, write the logic expression of output Y in the SOP form

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Writing SPOS from truth tables

- The follow shows the procedure on how to write SPOS from truth table:
  1. First, consider only the combinations that have their output as a 0.
  2. Then, write the maxterms for such combinations
  3. Lastly, AND all the maxterms to obtain the SPOS form

Example

- From the truth table obtain the logical expression in the SSOP form

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Answer: $Y = (A + B)(\bar{A} + \bar{B})$

$Y = \prod(0, 3)$