EEE130 Digital Electronics I
Lecture #4

- Boolean Algebra and Logic Simplification -

By Dr. Shahrel A. Suandi

Topics to be discussed

- 4-1 Boolean Operations and Expressions
- 4-2 Laws and Rules of Boolean Algebra
- 4-3 DeMorgan’s Theorems
- 4-4 Boolean Analysis of Logic Circuits
- 4-5 Simplification Using Boolean Algebra
- 4-6 Standard Forms of Boolean Expressions
- 4-7 Boolean Expression and Truth Tables
- 4-8 Karnaugh Map
- 4-9 Karnaugh Map SOP Minimization
- 4-10 Karnaugh Map POS Minimization
- 4-11 Five-Variable Karnaugh Maps

Introduction

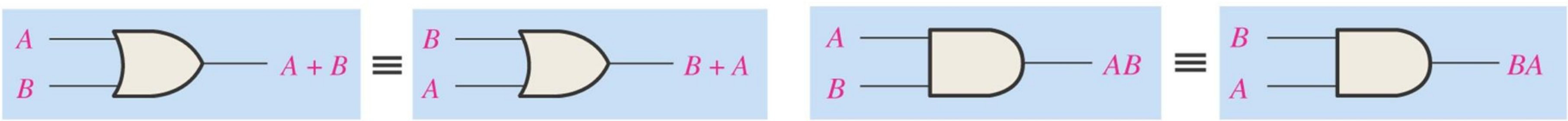
- 1854 – George Boole published a work titled “An Investigation of the Laws of Though, on Which Are Founded the Mathematical Theories of Logic and Probabilities”
  - Logical algebra = Boolean algebra
- 1938 – an MIT scholar, Claude Shannon applied Boole’s work to the analysis and design of logic circuits
  - Published a paper titled “A Symbolic Analysis of Relay and Switching Circuits”

4-1 Boolean Operations and Expressions

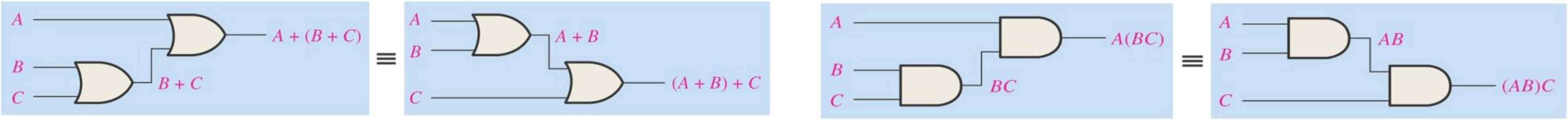
- Important terms:
  - *Variable*: a symbol (written in italic uppercase letter) used to represent a logical quantity
  - *Complement*: is the inverse of a variable and is indicated by a bar
  - *Literal*: is a variable or the complement of a variable
- Boolean Addition
  - In Boolean algebra, a **sum term** is a sum of literals
  - Equivalent to the **OR** operation
- Boolean Multiplication
  - In Boolean algebra, a **product term** is the product of literals
  - Equivalent to the **AND** operation

4-2 Laws and Rules of Boolean Algebra

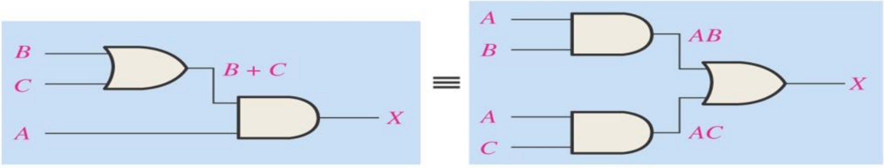
- Laws of Boolean Algebra:
  - Basic laws:
    - Commutative laws – this law states that the order in which the variables are ORed or ANDed makes no difference to the output



    - Associative laws – this law states that when ORing or ANDing more than two variables, the result is the same regardless of the grouping of the variables



    - Distributive law – 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 variables and then ORing the products. This law expresses the process of factoring in which the common variable *A* is factored out of the product terms



$X = A(B + C)$ $\qquad$ $X = AB + AC$

4-2 (2) Cont'd

- Basic Laws and Boolean expressions
  - Commutative law

$$A + B = B + A \qquad AB = BA$$

  - Associative law

$$A + (B + C) = (A + B) + C \qquad A(BC) = (AB)C$$

  - Distributive law

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

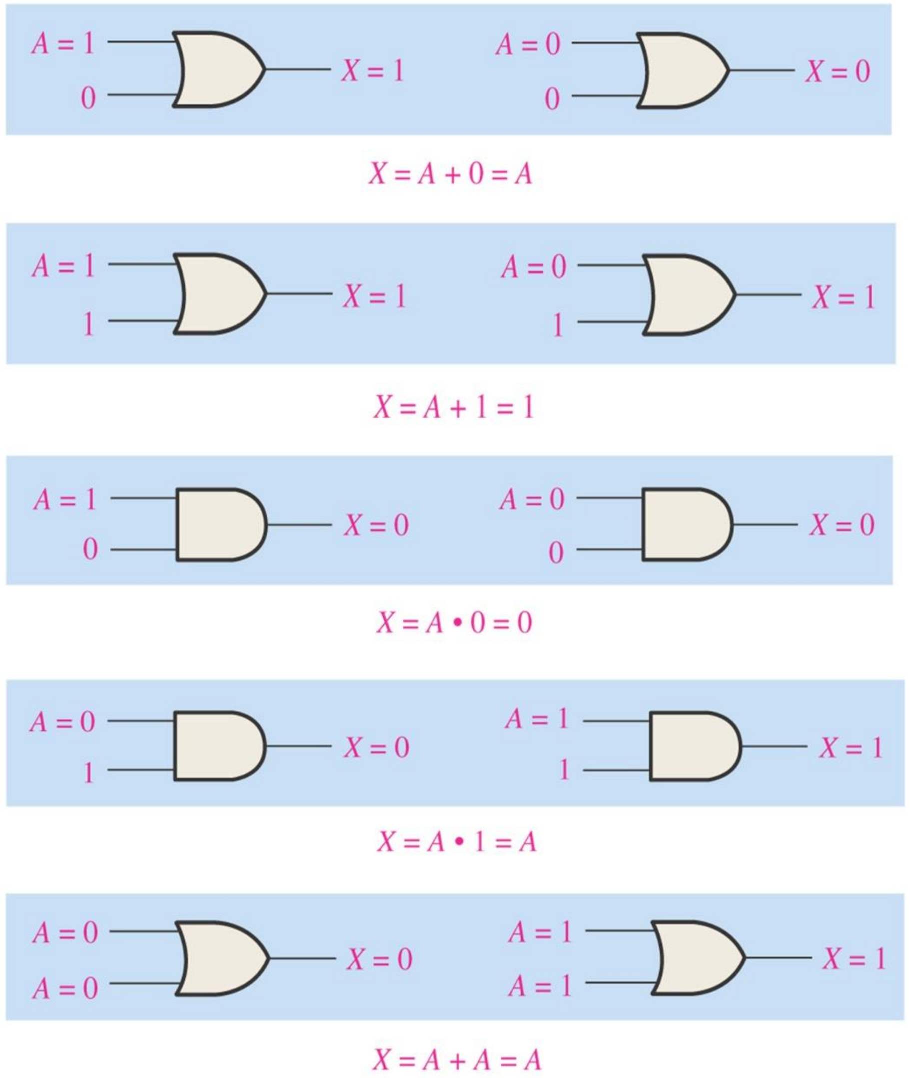
Rules of Boolean Algebra

- There are 12 rules for Boolean expressions. These rules are used to *manipulate and simplify* Boolean expressions
- What are the rules?? Refer to the table on the right for list below for details (or Table 4-1 in your text book)

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

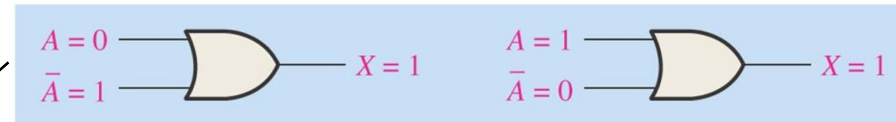
Rules Explanation(1)

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

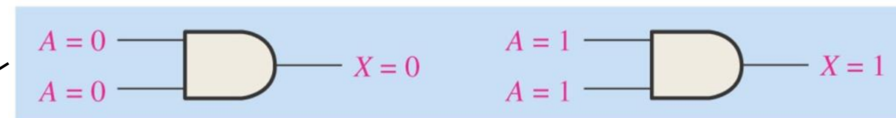


Rules Explanation(2)

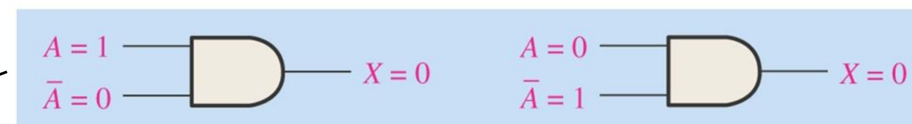
1. $A + 0 = A$
2. $A + 1 = 1$
3. $A \cdot 0 = 0$
4. $A \cdot 1 = A$
5. $A + A = A$
6. $A + \bar{A} = 1$
7. $A \cdot A = A$
8. $A \cdot \bar{A} = 0$
9. $\bar{\bar{A}} = A$
10. $A + AB = A$
11. $A + \bar{A}B = A + B$
12. $(A + B)(A + C) = A + BC$



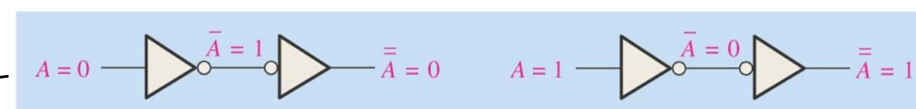
$X = A + \bar{A} = 1$



$X = A \bullet A = A$



$X = A \bullet \bar{A} = 0$



$\bar{\bar{A}} = A$

$$
\begin{aligned}
A + AB &= A(1 + B) \\
&= A \cdot 1 \\
&= A
\end{aligned}
$$

Rules Explanation(3)

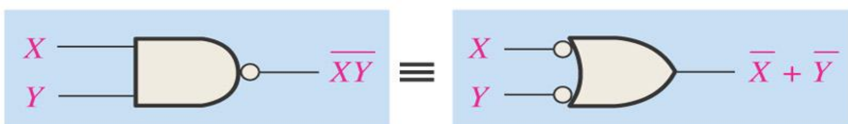
1. $A + 0 = A$
2. $A + 1 = 1$
3. $A \cdot 0 = 0$
4. $A \cdot 1 = A$
5. $A + A = A$
6. $A + \bar{A} = 1$
7. $A \cdot A = A$
8. $A \cdot \bar{A} = 0$
9. $\bar{\bar{A}} = A$
10. $A + AB = A$
11. $A + \bar{A}B = A + B$
12. $(A + B)(A + C) = A + BC$

$$\begin{aligned} A + \bar{A}B &= (A + AB) + \bar{A}B \\ &= (AA + AB) + \bar{A}B \\ &= AA + AB + A\bar{A} + \bar{A}B \\ &= (A + \bar{A})(A + B) \\ &= 1 \cdot (A + B) \\ &= A + B \end{aligned}$$

$$\begin{aligned} (A + B)(A + C) &= AA + AC + AB + BC \\ &= A + AC + AB + BC \\ &= A(1 + C) + AB + BC \\ &= A \cdot 1 + AB + BC \\ &= A(1 + B) + BC \\ &= A \cdot 1 + BC \\ &= A + BC \end{aligned}$$

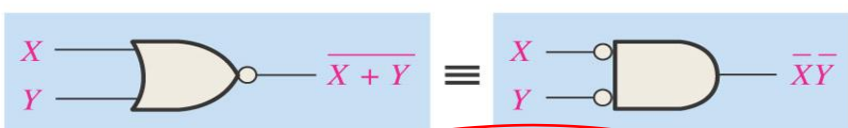
4-3 DeMorgan’s Theorems

- Why DeMorgan’s Theorems are important for Boolean algebra?
  - It provides mathematical verification of the equivalency of the NAND and negative-OR gates and NOR and negative-AND gates
  - In other words, it helps us to simplify Boolean expressions by utilizing:
    - NAND or negative-OR
    - NOR or negative-AND
- So please remember these two diagrams (and the truth table, of course)



NAND Negative-OR

$$XY = \overline{XY} \qquad X + Y = \overline{X} + \overline{Y}$$



NOR Negative-AND

$$X + Y = \overline{X + Y} \qquad XY = \overline{X}\,\overline{Y}$$

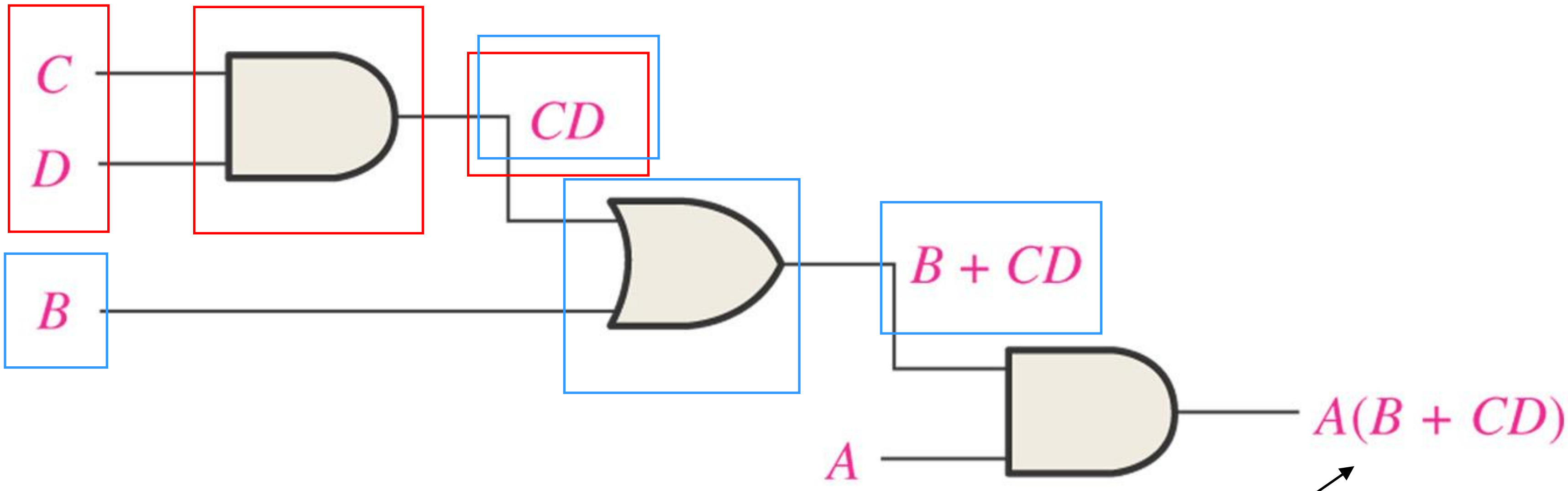
Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X} + \overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

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

4-4 Boolean Analysis of Logic Circuits

- Boolean algebra provides a concise way to express the operation of a logic circuit formed by a combination of logic gates so that the output can be determined for various combinations of input values
- Boolean expression for a logic circuit
  - Method of implementation
    - Begin at the left-most inputs and work toward the final output, writing the expression for each gate
    - Let's try to do example in Figure 4-16 (in the text book)
- Constructing a truth table for a logic circuit
  - Subsequent to the Boolean expression determination, the truth table is developed
  - We need to *evaluate the Boolean expression for all possible combinations of values for the input variables*
  - So, if there are *n* inputs then we need to evaluate 2^n combinations…

Figure 4-16



Begin from the left-most

The answer is

$$A(B + CD)$$

Evaluating the Expression

- We need to do these following items when evaluating the expression
  - Find the values of variables that make the expression equal to 1 (use Boolean addition and multiplication)
    - Evaluate the first term, *A*

$$A(B + CD) = 1 \cdot 1 = 1 \quad \text{if} \quad A = 1,\ B + CD = 1$$

    - Evaluate the second term, *(B + CD)*

$$B + CD = 1 + 0 = 1 \quad \text{if} \quad B = 1$$
$$B + CD = 0 + 1 = 1 \quad \text{if} \quad CD = 1$$
$$B + CD = 1 + 1 = 1 \quad \text{if} \quad B = 1, CD = 1$$

  - As we may see,
    - A(B+CD)=1
      - when A=1 and B=1 regardless of the values of C and D, or
      - when A=1 and C=1 and D=1 regardless of the value of B
    - A(B+CD) = 0 for other conditions

Putting the results in truth table format

- Remember these following items:

1. *List out* the 16 (or the *n*) input variable combinations of 1s and 0s in a binary sequence
2. *Place a 1 in the output column* for each combination of input variables that was determined in the evaluation
    - What are the inputs involved here??
        - Remember the observations that we made in the previous slide
3. Finally, *place a 0 in the output column* for all other combinations of input variables

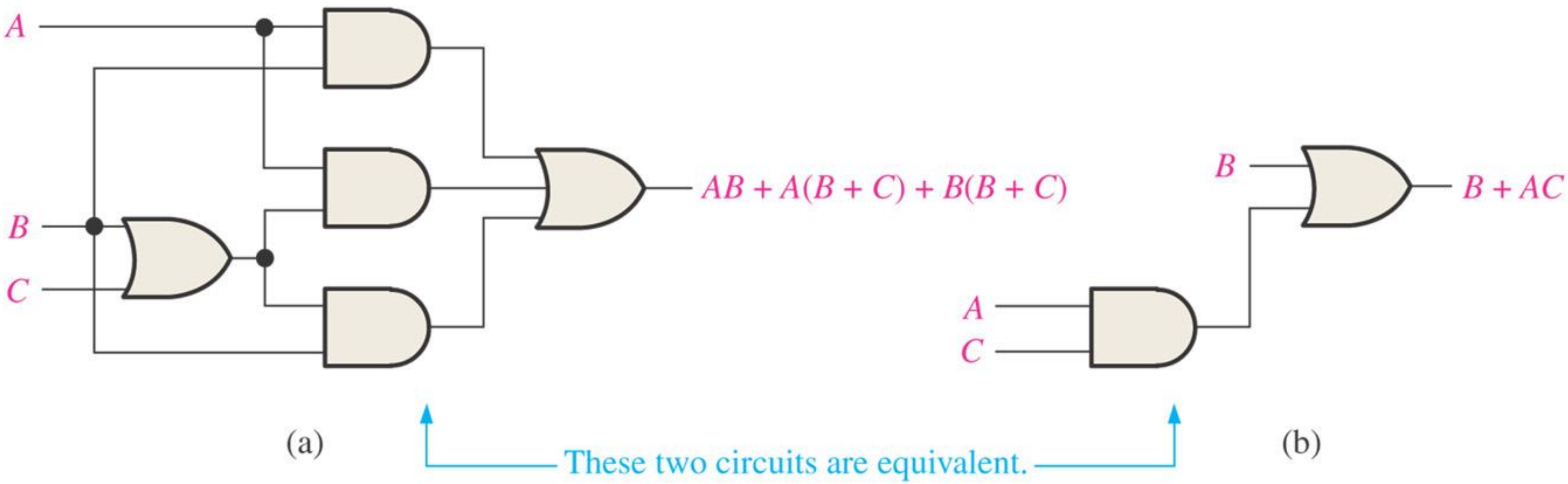
4-5 Simplification using Boolean Algebra

INPUTS				OUTPUT
A	B	C	D	A(B+CD)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
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	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

4-5 Simplification using Boolean Algebra

- In logic circuits applications, we need to *reduce* an expression to its *simplest form* or *change its form* to a more *convenient* one to implement the expression *more efficiently*
- How to do this??
  - We need to understand the laws, rules, DeMorgan Theorems and of course, basics in logic circuits like AND, OR, NAND, NOR, XOR, XNOR, NOT, etc.

Example of simplified logic circuits



- The details of this are presented in Example 4-8
- Notice that the output is the same from both circuits
- (b) is less complex compared to (a)

4-6 Standard Forms of Boolean Expressions

- Will be continued in the next slides